



Blender wiki PDF Manual conversion by Marco Ardito

Details, info, download: <http://amrc.altervista.org>

Updated: 10/11/2014 from:

<http://wiki.blender.org/index.php/Doc:JA/2.6/Manual>

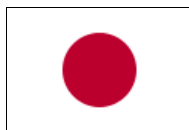


Table of Content

Table of Content	2
Blenderって何？	42
Video: From Blender 1.60 to 2.50	43
Version/Revision Milestones	43
IRC チャットチャンネル	46
どんな人が Blender を利用するのか？	46
読者となる方々へ	47
CG と Blender を学ぶ	47
バイナリディストリビューションのダウンロードとインストール	50
ハードウェアサポート	50
開発者プラットフォーム	50
ソースのコンパイル	51
プラグインのコンパイル	51
ダウンロード	53
バージョン	53
ディストリビューションのリリース	53
インストール	53
/opt または /usr/local へのインストール	53
Configuration	53
Alt+Mouse Conflict	53
アプリケーションへ Blender を追加	54
Dock に Blender を追加	55
ダウンロード	56
バージョン	56
インストール	56
ようこそ画面	56
ライセンス契約	56
インストールオプション	56
プログラムのオプション	56
インストール場所	57
インストール中	57
インストール完了	57
ポータブルのインストール	58
FreeBSD	59
他のサポートされていないオペレーティングシステム	59
MorphOS	59
入力 (Input)	60
ファイルパス (File Paths)	60
システム (System)	60
OSX	61
Windows	61
Unix (Linux/BSD/Solaris)	61
パス一覧	61
注意事項	62
ユーザースクリプトパス	62
環境変数	62
スクリプトパス & Missing Buttons	62
インタフェースのコンセプト	63
三原則	63
強力なインタフェース	64
概観	64
本マニュアルにおける慣例	66
一般的な使用方法	66
マウスボタンエミュレーション	66
テンキーエミュレーション	66
The default scene	67
The Info Window (main menu)	67
3D Window View	68
3D Window Header	68
Buttons (Properties) Window Header	68
Outliner Window	69
Timeline Window	69

ウィンドウ・フレームを最大にする。	70
ウィンドウ・フレームを分ける。	70
2つのフレームを結合する。	70
フレーム・サイズを変えること。	70
中身を入れ替える。	70
新しいウィンドウを開く。	70
ヘッダを隠す	72
ヘッダを表示する	72
ヘッダの位置	72
ウィンドウタイプボタン	72
メニューとボタン類	72
Windows 2000/Xp/Vista/7	73
Linux	73
コンソールウィンドウの状態及びエラーメッセージ	74
参考	75
スクリーン設定	76
新規スクリーンを追加	76
スクリーンの削除	76
スクリーンの再構成	76
デフォルトの上書き	77
追加レイアウト	77
Scene configuration	78
Adding a new Scene	78
A brief example	79
Deleting a Scene	79
Collapsing Menus	83
使用方法	84
インタラクション	84
リセンタリング	84
アイテムの位置	84
キーインタラクション	85
オプション	85
Blender へのパイメニューの追加	85
Operation Buttons	88
Toggle Buttons	88
Radio Buttons	88
Number Buttons	88
Expressions	88
Units	88
Menu Buttons	89
Color Selector Controls	90
Eye Dropper	92
Cascade Buttons	92
What is rendering?	95
Rendering an image using Full Render - Blender Internal	95
Rendering an image using OpenGL Render	95
Adjusting the resolution	96
Output format and output file	96
Saving your image	96
Rendering an animation using Full Render - Blender Internal	96
Rendering an animation using OpenGL Render	97
Showing Only Rendered Objects	97
The purposes of OpenGL Rendering	98
アンドゥ	100
リドゥ	100
アンドゥの履歴	100
Save and Auto Save	100
Recovering Auto Saves	101
そのほかのオプション	101
工場出荷時設定へのデフォルトのシーンの復元	102
OSの画面キャプチャ	103
Windowsの画面キャプチャ	103
Macintoshのスクリーンキャプチャ	103
GNU/Linux Screen Capture	103
Blender Window Dimension	105
Addon: 3D View: Screencast Keys	105
Web Access	107
Help Menu	107
Web-based Help options	107
Internal Help options	107

Configure	108
Save the new preferences	108
工場出荷時設定をロードする	108
Properties Window	109
View manipulation	109
2D Viewports	109
Manipulator	109
Menus	109
新規オブジェクト	110
Undo	110
Grease Pencil	110
Playback	110
Keyframing	110
Transform	111
Duplicate Data	111
Adding and deleting presets	112
Selecting presets	112
Setting presets to default	112
Export/Import key configuration	112
Mouse	112
Numpad emulation	112
View manipulation	112
Keymap editor	113
Customizing themes with Python	115
File Paths	122
Save & Load	122
Auto Save	122
General	124
Sound	124
Screencast	124
Open GL	124
Sequencer	125
Solid OpenGL lights	125
Miscellaneous	125
ウォーミングアップ	126
胴体の構築	127
ミラーモデリング	129
腕と脚	130
アンドゥリドゥ	130
頂点の一括移動	130
頭部	131
サブサーフェス (サブサーフ)	131
縮尺の拘束	132
Let's see what Gus looks like	133
カメラのセットアップ	134
地面	134
照明	135
レンダリング	136
作品の保存	137
マテリアルとテクスチャ	138
目とディテール	141
Flipping a duplicate around the cursor	142
口	142
目のマテリアル	143
レンダリング	143
保存	144
リギング	145
スキニング	147
頂点グループ	149
ポーゼング	150
Original position	150
Inverse Kinematics	150
Forward Kinematics	150
Gusが歩いている!	152
説明	154
保存	154

保存オプション	154
ファイル圧縮	154
相対リマップ	154
コピー保存	154
インクリメント保存の Tips	154
説明	155
ファイルブラウザやフォルダナビゲーションの使用	155
サイドパネル	155
システム	155
ブックマーク	155
最近の (recent)	155
開くのオプション	155
ヘッダーパネル	156
その他のファイルを開くのオプション	156
概要	157
保存	157
保存オプション	157
ファイルを圧縮 (Compress File)	157
相対パスを更新 (Remap Relative)	157
コピーを保存 (Save Copy)	157
インクリメント保存のための小ネタ	157
Collada (.dae)	158
Motion Capture (.bvh)	158
Scalable Vector Graphics	158
Stanford (.ply)	158
Stl (.stl)	158
3d studio (.3ds)	158
Autodesk FBX (.fbx)	158
Wavefront (.obj)	158
X3D Extensible 3D (.x3d)	158
Collada のエクスポート	159
プリセットのオペレータ	159
エクスポートデータのオプション	159
モディファイアの適用	159
Selection Only (選択部分のみ)	159
Include Children (子を含む)	159
Include Armatures (アーマチュアを含む)	160
Include Shape keys (シェイプキーを含む)	160
テクスチャのオプション	160
Only Active UV layer (アクティブな UV レイヤーのみ)	160
Include Textures (テクスチャを含む)	160
Copy (コピー)	160
アーマチュアのオプション	160
Deform Bones Only (ボーンのみ変形)	160
Export for Second Life (Second Life用エクスポート)	160
Collada のオプション	160
Triangulate (三角面化)	160
Use Object Instances (オブジェクトのインスタンスを使用)	160
Transformation Type (トランスフォームの種類)	160
Sort by Object Name (オブジェクト名で並べ替え)	161
Collada のインポート	161
技術的詳細	161
メッシュ	161
インポート	161
エクスポート	161
光源	161
インポート	161
エクスポート	161
マテリアルとエフェクト	161
エクスポート	161
アニメーション	161
エクスポートとインポート	161
Nodes	162
重要事項	162
Collada (.dae)	163
Motion Capture (.bvh)	163
Stanford (.ply)	163
Stl (.stl)	163

Autodesk® 3DS Max® (.3ds)	163
Autodesk® FBX (.fbx)	163
Wavefront (.obj)	163
X3D Extensible 3D (.x3d)	163
画像形式	164
ハイダイナミックレンジ画像	164
他の形式	164
動画形式	164
カラーモード	164
色深度	164
リファレンス	164
Render Output	165
Multi-layer, Multi-pass, tile-based files	165
Radiance HDR	166
Outliner	167
Users (Sharing)	167
Fake User	167
Copying and Linking Objects Between Scenes	167
Appending or Linking Across Files	167
Proxy Objects	167
Pack and Unpack Data	167
Unpack Data	168
Copying and Linking Datablocks	169
Copying and Linking Object Datablocks	170
Copying and Linking other Datablocks	170
Unlinked Datablocks	170
Planning Your Timeline	171
Loading the UI with “File” → “Open”	172
Working with Scenes	172
Adding a Scene	172
Naming a Scene	173
Linking to a Scene	173
Removing a scene from the file	173
説明	174
アウトライナーの操作系	174
アウトライナー・ウィンドウを選択	174
アウトライナーの利用	175
選択とアクティブ化	175
オブジェクト・レベルの制限の切り替え	177
検索 (Searching)	178
表示のフィルタリング (Filtering)	178
例	179
General Procedure	180
Folder and File Organization	180
Appending library objects into your current project	180
Reusing Objects (Meshes, Curves, Cameras, Lights, ...)	181
Reusing Material/Texture Settings	181
Reusing Node Layouts	182
Proxy Objects	182
Armatures and Multiple instances	182
ビューを回転させる	184
説明	184
補足	184
トラックボールとターンテーブル	184
View のパン(移動)	185
説明	185
View のズーム(拡大縮小)	185
説明	185
透視投影 (Perspective)と平行投影 (Orthographic)	186
解説	186
オプション	186
技術的な詳細	187
透視投影の定義	187
平行投影の定義	187
ビューの回転	187
解説	187

オプション	187
トラックボール/ターンテーブル	188
ビューのパン	188
解説	188
ビューのズーム	188
解説	188
ポーターにズーム (Zoom Border)	188
ドリー (Dolly the View)	188
解説	188
視点を揃える	189
視点を揃える (Align View)	189
ローカルビューとグローバルビュー	189
四分割表示 (Quad View)	189
矩形でクリッピング (View Clipping Border)	190
解説	190
例	190
視点の操作	190
解説	190
Description	192
View Properties Panel	192
Description	192
View	192
Item	193
Display	193
Background Image	194
Settings	194
Shortcuts	195
Render Border	196
Working with Layers	197
Viewing layers	197
Locking to the scene	197
Multiple Layers	197
Moving objects between layers	197
Example of object layer arrangement	198
解説	199
例	199
Basic transformations	200
Advanced Transformations	200
Transform Control	200
3D View	201
Transform Widget	201
Shortcuts in the 3D View	201
Controlling Grab/Move Precision	202
Orientations	202
Other Editor Windows	203
Python Scripting	203
Hints	203
解説	204
Usage	204
Rotation using the keyboard shortcut	204
Constraining the rotation axis (axis locking)	204
Fine Tuning The Rotation	204
Rotation with the 3D Transform Manipulator	205
Rotation with the Properties Panel	205
解説	206
使い方	206
キーボードショートカットを使った拡大縮小	206
拡大縮小の軸を制限する (軸ロック)	206
拡大縮小の精密な調整	206
マニピュレータを使った拡大縮小	207
プロパティパネルを使った拡大縮小	207
解説	208
使い方	208
Description	209
Usage	209
Description	211
Usage	211
Usage	213

Cursor position and view	213
Example	214
解説	215
使い方	215
Options	216
トランスフォームの量	217
トランスフォームの方向	217
トランスフォームの基準点	217
トランスフォームのスナップ	217
Description	218
Usage	218
With hotkeys	218
With the Transform Manipulator	218
Holding CTRL	218
Grab/move transformations	218
Rotation transformations	218
Scale transformations	219
Holding SHIFT	219
Holding CTRL and SHIFT	219
Grab/move transformations	219
Rotation transformations	219
Scale transformations	219
Translation	220
Rotation	220
Scaling	220
Numeric input via the Properties shelf	220
Transform Properties Panel	222
Options in Object mode	222
Transform Properties Locking	223
Description	224
Clear Options	224
Apply Object transformations	224
Apply Options	224
マニピュレータの操作	226
マニピュレータの設定	226
座標系 (Transform Orientation) の選択	226
Our Demo Cube	228
Orientations	228
Global	229
Local	229
Normal	229
Gimbal	230
View	230
Custom Orientations	230
Description	232
Usage	232
Axis locking types	232
Axis locking	232
Plane locking	232
Axis locking modes	232
ピボットの選択	234
In Object mode	235
In Edit mode	235
Single selection	235
Multiple selection	236
In Object mode	237
In Edit mode	237
Positioning the 3D cursor	239
Direct placement with the mouse	239
Using the Snap menu	239
Numeric input	240
In Object Mode	241
In Edit Mode	241
In Object mode	242
In Edit mode	242
スナップ (Snap)	244
トランスフォームのスナップ	244
スナップモード	244
スナップ要素	244
スナップ対象	245

さらなるスナップ操作	245
Object mode	247
Edit mode	247
Influence	248
Options	248
Examples	249
グリース・ペンシル (Grease Pencil) とは?	250
線を描く	251
説明	251
ヒント	251
直線を描く	251
説明	251
描画の消去	251
説明	251
オプション	251
ヒント	251
Sketching Session について	251
説明	252
ヒント	252
Grease Pencil の設定の共有に関して	252
描画の設定	252
制限	252
描画する際の感度について	252
タブレットのユーザーのために	252
Drawing Planes	252
2.5x 系からの追加機能	252
スケッチのアニメーション	253
Onion Skinning	253
スケッチのタイミングの調整	253
スケッチのコピー	253
Usage	255
Shortcuts	256
消去	258
結合	258
Select Links	258
選択とアクティブな Object	260
ポイント選択	260
境界(矩形)選択	260
説明	260
例	260
ヒント	261
投げ縄選択	261
説明	261
使い方	261
メニューによる選択	261
グループ選択	261
説明	261
オプション	261
リンク選択	262
説明	262
オプション	262
タイプによる選択	262
説明	263
オプション	263
レイヤーによる選択	263
説明	263
オプション	263
その他の選択方法	263
Object Mode	265
Creation and deletion	265
Add	265
Duplicate	265
Join	266
Delete	266
Transformation tools	266

Translate	266
Rotate	266
Scale	266
親子関係の作成	267
頂点の親子関係	267
オプション	267
子オブジェクトの移動	267
親子関係の解消 / 親オブジェクトのクリア	267
親子関係の例	268
ヒント	268
オブジェクトの分割	268
オプション	268
オブジェクトのグループ化	268
オプション	268
グループ化されたオブジェクトの選択	269
オプション	269
概要	271
例	271
リンク複製	271
Description	271
Examples	271
Procedural Duplication	272
Linked Library Duplication	272
ヒント	272
配置ツールとしてのDupliVerts	273
Setup	273
Rearranging	273
Orientation	273
モデリングツールとしてのDupliVerts	274
関連項目	274
Basic usage	275
Scale	275
基本的な使用方法	276
DupliGroup and Dynamic Linking	276
例	276
Making a DupliGroup Object Real	276
Examples	277
外部リンク	278
エディット・モードについて	279
表示の仕様	279
ツール・シェルフ (Tool Shelf)	280
プロパティ・シェルフ (Properties Shelf)	280
頂点 (Vertex)	281
エッジ (Edge)	281
面 (Face)	281
ループ (Loop)	281
エッジ・ループ (Edge Loop)	281
面ループ (Face Loop)	282
Plane	283
Cube	283
Circle	283
UV Sphere	284
Icosphere	284
Cylinder	284
Cone	284
Torus	285
Grid	285
Monkey	285
Add-ons	285
Overhang	287
Thickness	287
Intersections	287
Distortion	287
Sharp Edges	288
Selection Modes	289
Select Mode Header Widgets	289
Select Mode Pop-up	289
Switching select mode	289

Selection Tools	290
Select Modes	292
Select Mode popup	292
Select Mode header widgets	292
Selected elements after switching select mode	292
Adding to a Selection	294
Selecting Elements in a Region	294
Rectangular region (Border select)	294
Circular region	295
Lasso region	295
Additional Selection Tools	295
Select Similar	297
Selecting Loops	298
Edge Loops and Vertex Loops	298
Face Loops	299
Edge Ring	299
Path Selection	300
Loop Inner-Region	300
Boundary Loop	301
Edge Loops	302
Edge Rings	302
Face Loops	304
Ngons in Face Select Mode	304
ツールの種類	305
メッシュツールへのアクセス	305
メッシュツールパレット	305
メニュー	306
Delete	309
Dissolve	309
Convert Triangles to Quads	310
Unsubdivide	310
Merging (結合)	310
Merging Vertices	310
AutoMerge Editing	310
Remove Doubles	311
Further Reading	313
X-Mirror	314
Topology Mirror	314
Mirror Modifier	314
Snap to Symmetry	314
Symmetrize Mesh	315
Mirroring Geometry	316
Merging	317
Merge Vertices	317
AutoMerge Editing	317
Remove Doubles	317
Separating	317
Rip	318
Examples	318
Limitations	319
Rip Fill	319
Split	320
Separate	320
Vertex Connect	320
Vertex Slide	321
Smooth	322
Make Vertex Parent	322
Add Hook	322
Blend From Shape/Propagate Shapes	322
Set Edge Attributes	323
Mark Seam and Clear Seam	323
Mark Sharp and Clear Sharp	323
Adjust Bevel Weight	323
Crease SubSurf	323
Mark Freestyle Edge/Clear Freestyle Edge	323
Edge Slide	324
Usage	324
Even mode	324

Limitations & Workarounds	325
Rotate Edge	325
Using Face Selection	326
Delete Edge Loop	326
Example	326
Collapse	327
Edge Split	327
Bridge Edge Loops	328
Creating Faces	332
Make Edge/Face	332
Fill	332
Beauty Fill	333
Grid Fill	333
Convert Quads to Triangles	334
Convert Triangles to Quads	334
Solidify	335
Rotate Edges	336
Normals	336
Flip Direction	337
Recalculate Normals	337
押す/引く	338
湾曲 (Warp)	338
例	338
せん断 (Shear)	339
球へ変形 (To Sphere)	340
例	340
対称軸	342
ピボット (Pivot point)	342
座標系 (Transform Orientation)	343
ラプラシアンスムーズ (Laplacian Smooth)	345
Extrude Region	349
Extrude Individual	349
Extrude Edges and Vertices Only	350
オプション	352
Example	353
Angle	354
Dupli	354
Merge Duplicates	355
Recalculate Normals	355
オプション	357
例	358
1辺	358
三角形の2辺	359
四角形の対抗する2辺	359
四角形の隣接2辺	360
3辺	361
三角形	361
四角形/4辺	362
多分割	362
使い方	364
オプション	364
Usage	366
Options	366
Confirming and selection	368
Limitations	368
Optimizations	368
Knife Project	368
Examples	368
Known Issues	369
ベベルモディファイア	373
使い方	373
オプション	373
例	374
要素のソート (sort)	375
メッシュへのスナップ	376
収縮包装モディファイア (Shrinkwrap)	376

Sculpt Mode	377
Sculpt Brushes	377
Sculpting with the Multires Modifier	378
Sculpt Properties Panel	378
Brush Menu	378
Stroke Menu	379
Curve Menu	379
Texture Menu	379
Symmetry Menu	380
Options Menu	380
Appearance Menu	380
Tool Menu	380
Hiding and Revealing Mesh	380
Keyboard Shortcuts	381
Typical usage scenarios for Vertex groups	384
Creating Vertex Groups	384
Group Name	385
Active Group	385
Deleting vertex Groups	385
Locking Vertex Groups	385
Working with Content of Vertex Groups	385
Assigning verts to a Group	386
Checking assignments	386
Removing assignments from a Group	386
Using groups for Selecting/Deselecting	386
Finding ungrouped verts	387
Keyboard Shortcuts	387
Vertex Group Management	387
Hints	388
Vertex Group Categories	389
The Deform Groups	389
The Other Groups	389
The Weight Table	389
Set the Active Group	389
Display Weights in Edit Mode	390
Edit Weights in Edit Mode	390
Change a weight	390
Paste a weight to other verts	390
Delete a weight from a Group	390
The Function bar	391
About locked Vertex Groups	391
Weight Paint Mode	392
The weighting Color Code	392
Brushes	393
The main brush properties	393
Normalize Options	393
The Brush stroke definition	394
The brush Falloff curve	394
The brush appearance	394
Brush presets	395
Customizing brush color space	395
Selection Masking	395
Details about selecting	395
Vertex Selection Masking	396
Face Selection Masking	396
Hide/Unhide Faces	396
Hide/Unhide Vertices	396
The Clipping Border	397
Weight Paint Options	397
Weight Paint Tools	398
Weight Painting for Bones	398
Weight Painting for Particles	398
The Subset Option	400
Normalize All	400
Operator parameters	400
Normalize	400
Operator parameters	400
Mirror	401

Operator parameters	401
Invert	401
Operator parameters	402
Clean	402
Operator parameters	402
Levels	402
Operator parameters	403
Blend	403
Operator parameters	404
Transfer Weights	404
Prepare the copy	404
Call the tool	405
Redo Panel Confusion	405
Workaround	405
Operator parameters	405
Limit total	406
Operator parameters	406
Weight Gradient (wip)	406
Operator parameters	406
Smooth shading	407
Smoothing parts of a mesh	407
Auto Smooth	407
Edge Split Modifier	407
Smoothing the mesh geometry	408
メッシュ編集ツール	408
モディファイア	408
[edit] Fill Holes	410
[edit] Split Non-Planar Faces	410
[edit] Delete Loose Geometry	410
[edit] Degenerate Dissolve	410
Tutorials	411
Curve Primitives	411
Bezier Curves	411
Editing Bezier Curves	411
Curve Properties	412
Shape	412
Geometry	413
Path Animation	414
Active Spline	415
Non-Uniform Rational B-Splines (NURBS)	415
Editing NURBS Curves	415
Active Spline	415
Path	416
選択 (Select) メニュー	417
N番目ごとに選択/チェッカー選択 (Every Nth)	417
最初/最後を選択/解除 (Select/Deselect First/Last)	417
次/前を選択 (Select Next/Previous)	417
より多く/より少なく選択 (More and Less)	417
Curve Display	419
Display Options	419
Hiding Elements	419
Basic Curve Editing (translation, rotation, scale)	419
Snapping	419
Deforming Tools	419
Smoothing	420
Mirror	420
Set Bézier Handle Type	420
Extending Curves	420
Subdivision	420
Duplication	420
Joining Curve Segments	421
Separating Curves	421
Deleting Elements	421
Opening and Closing a Curve	422
Switch Direction	422
Converting Tools	422
Converting Curve Type	422
Convert Curve to Mesh	423

Convert Mesh to Curve	423
Curve Parenting	423
Hooks	423
Set Goal Weight	423
Interface	424
Example	425
Curve Extrusion	426
Extrusion	426
Simple Extrusion	427
Advanced Extrusion	427
Examples	428
Finding Surface Tools	430
Visualization	430
Surface Structure	430
Control Points, Rows and Grid	431
Surface Resolution	431
Closed and Open Surfaces	431
Knots	431
Order	432
Weight	432
Preset Weights	433
Primitives	433
Select Menu	435
Every Nth	435
Control Point Row	435
More and Less	435
Surface Editing	436
Basic Surface Editing (translation, rotation, scale)	436
Advanced Transform Tools	436
NURBS Control Points Settings	436
Adding or Extruding	436
Examples	437
Opening or Closing a Surface	437
Duplication	437
Deleting Elements	438
Example	438
Joining or Merging Surfaces	438
Examples	439
Subdivision	439
Spin	439
Switch Direction	439
Other Specials Options	439
Conversion	440
Retopology	440
Misc Editing	440
Notes	441
Editing Text	441
Inserting Text	442
Special Characters	442
Convert text to text object	442
3D Mesh	443
Text Selection	443
Formatting Text	443
Fonts	443
Loading and Changing Fonts	444
Size and Shear	444
Objects as Fonts	444
Text on Curve	445
Underline	445
Character	445
Setting Case	446
Paragraph	446
Align	446
Spacing	447
Offset	447
Shape	447
Resolution	447
Fill	447
Textures	447

Geometry	448
Text Box	449
Frame size	449
Adding/Deleting a Frame	449
Example: Text Flow	449
Example: Multiple columns	450
Assigning Materials	450
プリミティブ(基本形状)	451
可視化	451
メタボールのオプション	451
分割数	452
しきい値(影響)	452
Update	452
メタの構造	453
技術詳細	453
Underlying Structure	453
メタの形状	455
Stiffness	455
Negative Influence	455
要素を隠す	456
要素の削除	456
Conversion	456
Object Families	456
例	457
ヒント	457
Settings	458
Usage and functions	458
Visualization	460
修飾(Modify)	461
生成(Generate)	461
変形(Deform)	461
シミュレート(Simulate)	462
Interface	463
Stack	463
オプション	465
使い道	465
解説	466
共通設定	466
頂点ウェイト編集(Vertex Weight Edit)モディファイア	466
オプション	467
頂点ウェイト合成(Vertex Weight Mix)モディファイア	467
オプション	468
頂点ウェイト近接(Vertex Weight Proximity)モディファイア	468
オプション	468
例	469
ターゲットオブジェクトからの距離を使う	469
ターゲットオブジェクトのジオメトリからの距離を使う	469
テクスチャとマッピングカーブを使う	470
関連項目	471
解説	472
オプション	472
解説	473
オプション	473
例	473
単純な平面	473
Decimateされた円柱	473
解説	475
オプション	475
例	475
解説	476
オプション	476
解説	477
オプション	477
ヒント	478
MirrorモディファイアとSubdivision Surfaceモディファイア	478
Mirrorするために整列させる	478
オプション	479
Description	480

Usage	480
Mode	480
Octree Depth and Scale	480
Disconnected Pieces	481
Sharpness	481
Demo Videos	481
解説	483
オプション	483
解説	484
オプション	484
解説	485
オプション	485
Weighted creases (重み付き折り目) for subdivision surfaces	485
解説	485
オプション	485
説明	486
オプション	486
Description	487
オプション	487
Description	488
オプション	488
Description	489
オプション	489
解説	490
オプション	490
以下も参照してください	491
オプション	492
Hints	492
Example/Tutorial(s)	492
パーティクルとラティス	492
概要	494
オプション	494
See Also	494
概要	495
オプション	495
概要	497
オプション	497
解説	498
オプション	498
Options	499
概要	500
オプション	500
Technical Details and Hints	501
解説	502
作業の流れ	502
クロスシミュレーションの生成	502
クロスパネル	502
素材	503
減衰	503
ピン止め	503
衝突	503
衝突の設定	503
クロス自身との衝突	504
共有されたレイヤー	504
メッシュオブジェクトの衝突	504
クロスとオブジェクトの衝突	504
メッシュオブジェクトのモディファイアスタック	504
クロス キャッシュ	505
衝突をバイクする	505
シミュレーションのキャッシュを編集する	505
トラブルシューティング	505
例	505
成形のためにシミュレーションを使う	505
クロスを滑らかにする	506
アーマチュアとクロス	506
クロスをソフトボディとして使う	506
クロスと風	506
オプション	507

ソフトボディおよびクロスの衝突	508
力場の相互作用	508
例	508
ヒント	508
モディファイアの追加	509
タイプ	509
その他役に立つページ	509
解説	510
作業の流れ	510
さらに詳しく	510
シミュレーション内部	511
Ocean モディファイア	511
解説	511
Geometry オプション	511
シミュレータのオプション	511
Wave オプション	512
Sim Data Generation オプション	512
ベイキング	512
データファイル	513
泡のベイキング	513
Baking オプション	513
来歴	513
解説	514
概要	514
オプション	514
作成	515
表示	515
パスを使う	515
関連項目	518
作業の流れ	519
パーティクルシステムの作成	519
パーティクルシステムの種類	520
共通オプション	520
リンク	520
開発ノート	521
内部処理	521
作業の流れ	521
典型的な使い方	522
柔体の作成	522
シミュレーション品質	523
キャッシュとバイク	523
リアルタイムなインタラクション	523
ヒント	524
リンク	524
Viewing Restrictions	525
Global Influences	525
Lighting Settings	525
Lighting in the Workflow	525
Overriding Materials to Reset Lighting	525
Object Data	528
Preview	528
Lamp	528
Description	529
Falloff types	529
Lin/Quad Weighted	529
Linear	529
Quadratic	529
Mixing "Linear" and "Quad"	530
Zeroing both "Linear" and "Quad"	530
Graphical Summary	530
Custom Curve	531
Inverse Square	532
Inverse Linear	532
Constant	533
Sphere	533
Examples	534
Distance	534
Inverse Square	534
Sphere	535
Hints	535
Options	536

Lighting Groups	538
Materials	538
Render Layers	538
See Also	539
Lamps: Ray-traced Shadows	540
Lamps: Buffered Shadows	541
See Also	542
What is Quasi-Monte Carlo?	544
Example	545
See also	545
Lamp Options	547
Shadows	547
Lamp options	549
Shadows	549
Spot Shape	550
Shadow Buffer Types	553
"Classical" and "Classic-Halfway"	553
"Irregular"	555
Deep generation method	555
Common options	555
Hints	556
See Also	557
Lamp options	558
Shadows	558
Area Shape	558
Shadow Samples	560
Sample Generator Types	560
Technical Details	560
Hints	561
Options	562
Lamp options	563
Sky & Atmosphere	563
Shadow	563
Options	565
Sky	565
Atmosphere	566
Examples	567
Sky	567
Atmosphere	569
Hints and limitations	571
Environment or Ambient Only	572
Single Rig	572
Two-Point Rig	573
Three-Point Rigs	573
Studio rig	573
Standard Rig	574
Four-point Rig	574
Troubleshooting	574
Options	577
Gather	577
Raytrace	577
Approximate	579
Common Settings	579
Technical Details	579
Hints	580
Options	582
Approximate	582
Description	584
Options	584
Examples	584
Hints	585
How Materials Work	586
Using Materials	587
別のオブジェクトからマテリアルを共有する	588
オブジェクトまたはオブジェクトデータにマテリアルをリンクする	589
マテリアルのオプション	589
頂点グループの管理	590
頂点グループに頂点を割り当てる	590
頂点にマテリアルを割り当てる	590

概要	592
オプション	592
Examples	592
オプション	593
技術詳細	593
ヒント	593
ランベルト	593
オプション	593
Oren-Nayar	594
オプション	594
Toon	594
オプション	595
ミナルト	595
オプション	595
フレネル	596
オプション	596
概要	597
オプション	597
技術詳細	597
CookTorr	597
概要	598
オプション	598
Phong	598
概要	598
オプション	598
惑星の大気	598
Blinn	598
概要	599
オプション	599
Toon	599
概要	599
オプション	599
ヒント	599
WardIso	599
概要	600
オプション	600
概要	601
オプション	601
Colorbands	602
オプション	602
概要	605
オプション	605
例	606
フレネル	606
レイトレースによる透明	608
オプション	608
例	609
屈折率	609
フレネル	609
Depth	610
ヒント	610
透過物の影	611
一般的な材料のIOR値	611
表面下散乱を有効にする	614
オプション	614
自作のSSSマテリアルの開発	615
例	615
肌	615
関連項目	615
Context	617
マテリアルスロット	617
Multiple materials	617
マテリアルの命名とリンク	617
Material type	618
Material Properties Overview	618
Surface and Wire materials	618

プレビュー	618
Diffuse	619
Specular	619
Shading	619
Transparency	619
Mirror	619
Subsurface Scattering	619
Strand	619
オプション	619
Shadow	620
Volume Material	621
Halo Material	621
ノードエディタへのアクセス	622
マテリアルボタンでノードマテリアルを有効にする	622
外部リンク	623
Accessing The Node Editor	624
Activating Nodes	624
Node Editor Window Actions	625
Node Editor Header	625
At a glance	625
Menus	625
View, Select and Add	626
Node	626
Buttons	626
Material/Composite/Texture Selector	626
Use Nodes Button	626
Use Pinned Button	626
Go to Parent Button	626
Snap Button	627
Copy Nodes Button	627
Paste Nodes Button	627
Layout Nodes	627
Free Unused Button	628
Backdrop	628
Auto Render	628
Collapsing toggles	629
Sizing the node	630
Sockets	630
Curves	630
RGB Curves	631
Selecting curve points	631
Editing curves	631
Editing the view	631
Special tools	631
Adding Nodes	632
Arranging Nodes	632
Connecting nodes	632
Disconnecting nodes	632
Duplicating a node	632
Grouping Nodes	633
Editing Node Groups	633
Ungrouping Nodes	633
Appending Node Groups	633
Settings	635
Brush	635
Tool	635
Stroke	635
Curves	636
Appearance	636
Options	636
Texturing Volumes	638
オプション	638
密度	638
シェーディング	638
散乱	638
非対称	639
Transmission	639
Emission	639
Reflection	640
ヒント	640
照明	640
Transparency	641
Integration	641
Options	641

Examples	641
Options	642
Options	643
Halo Panel	643
Effects	644
Flare Panel	644
Lens Flares	645
例	645
Dotmatrix display	645
Material Textures	647
World Textures	648
Brush Textures	648
Choosing the Texture context	649
Choosing the Texture data type	649
Textures Slots	649
Creating a new Texture Datablock in a new Texture Slot	649
Creating a new Texture Datablock in a non-empty slot	649
Sharing a Texture Datablock in a non-empty slot	650
Textures Stack	651
Texture Datablock	651
Texture Type	651
Preview	651
Colors	652
Mapping	652
Influence	652
About UVs	653
Getting Started	653
Workflow	653
Mapping Types	654
Basic Mapping	654
Cube	654
Cylinder and Sphere	654
Project From View	655
Resetting UVs	655
Advanced Mapping	655
Unwrapping Using Seams	655
Marking Seams	656
Unwrap	656
Smart UV Project	657
Lightmap	658
Follow Active Quads	658
共通のオプション	661
Noise Basis	661
Nabla	662
ヒント	662
Options	663
オプション	664
技術概要	664
オプション	665
技術詳細	665
オプション	666
オプション	667
オプション	668
オプション	669
技術詳細	669
オプション	670
オプション	671
技術詳細	671
オプション	672
技術詳細	672
About Image Based Texturing	673
Workflow	673
Using Images and Materials	673
Creating an Image Texture	674
Examples	674
Layering UV Textures	674
Mix and Match Materials	675

Using Alpha Transparency	676
UV Textures vs. Procedural Textures	676
Settings	677
Image	677
Image Sampling	678
Image Mapping	680
Options	681
Image	681
テクスチャノードの利用	683
出力を増やす	683
関連する情報	683
Accessing The Node Editor	684
Activating Nodes	684
Node Editor Window Actions	685
Node Editor Header	685
At a glance	685
Menus	685
View, Select and Add	686
Node	686
Buttons	686
Material/Composite/Texture Selector	686
Use Nodes Button	686
Use Pinned Button	686
Go to Parent Button	686
Snap Button	687
Copy Nodes Button	687
Paste Nodes Button	687
Layout Nodes	687
Free Unused Button	688
Backdrop	688
Auto Render	688
[edit] Collapsing toggles	689
[edit] Sizing the node	690
[edit] Sockets	690
[edit] Curves	690
[edit] RGB Curves	691
[edit] Selecting curve points	691
[edit] Editing curves	691
[edit] Editing the view	691
[edit] Special tools	691
Adding Nodes	692
Arranging Nodes	692
Connecting nodes	692
Disconnecting nodes	692
Duplicating a node	692
Grouping Nodes	693
Editing Node Groups	693
Ungrouping Nodes	693
Appending Node Groups	693
Settings	695
Options	697
Point Density	697
Turbulence	698
Options	699
Ocean panel	699
Getting Started	701
Texture Preview	702
Brushes Settings	702
Brush	702
Types of brushes	703
Texture	703
Stroke	704
Curve	705
Paint options	705
Overlay	706
Appearance	706
Saving	706
Using an External Image Editor	706
Getting Started	707
Hints	707
Coordinates	708
Projection	709
Inheriting coordinates from the parent object	709
Coordinate Offset, Scaling and Transformation	709

UVs Explained	711
Cartography Example	711
Half-Sphere Example	711
The UV Editor	712
Advantages of UVs	712
About UVs	713
Getting Started	713
Workflow	713
Mapping Types	714
Basic Mapping	714
Cube	714
Cylinder and Sphere	714
Project From View	715
Resetting UVs	715
Advanced Mapping	715
Unwrapping Using Seams	715
Marking Seams	716
Unwrap	716
Smart UV Project	717
Lightmap	718
Follow Active Quads	718
UV の転送	719
作業手順	719
複数の UVマップ	719
UVテクスチャの一覧	719
Selecting UVs	720
Selection Modes	720
Transforming UVs	721
Axis Locking	721
Pivot Points	721
Proportional Editing	721
Snapping	721
Weld and Align	722
Mirror	722
Stitch	722
Minimize Stretch	722
Reverse and Rotate UVs	722
Pinning	722
Optimizing the UV Layout	722
Combining UV Maps	723
Average Island Scale	724
Packing Islands	724
Constraining to Image Bounds	724
Iteration and Refinement	724
Refining the Layout	724
Reusing Textures	724
Exporting UV Layout Image	726
Applying Textures to UVs	726
Use UV Coordinates	727
Face Textures	727
Loading and Saving Images	727
Replacing the active Image	728
New Images	728
Using the Test Grid	728
Image Settings	729
Saving Images	729
Modifying your Image Texture	730
Packing Images inside the Blend file	730
Surface and Wire materials	731
Diffuse	731
Specular	731
Shading	731
Geometry	731
Other Controls	732
Volume materials	732
Halo materials	733

Description	734
Workflow	734
Description	735
Options	735
Hints	735
解説	737
オプション	737
テクスチャ	737
解説	739
オプション	739
透明度	739
例	739
解説	740
オプション	740
アーマチャ	741
編集	741
スキニング	741
ポージング	741
スケルトンパネル (全モード)	743
表示パネル (全モード)	743
ボーングループパネル (全モード)	743
ポーズライブラリパネル (ポーズモード)	743
ゴーストパネル (全モード)	744
iTaSC パラメータパネル (全モード)	744
モーションパスパネル (ポーズモード)	744
カスタムプロパティパネル (全モード)	744
ボーンパネルの概要	745
関係パネル (エディットモード)	745
表示パネル (オブジェクトモード)	745
変形パネル (全モード)	745
カスタムプロパティパネル (全モード)	745
変換パネル (エディットモードとポーズモード)	746
変換ロックパネル (ポーズモード)	746
インバースキネマティクスパネル (ポーズモード)	746
Bones Visualizatio	747
Bones properties	748
Bones Rigidity	748
Bones influence	748
ディスプレイパネル	750
ボーンの種類	750
八面体のボーン	750
棒状のボーン	751
B-Bone bone	751
エンベロープのボーン	751
属性	751
Shaped Bones	751
属性	752
アーマチュアレイヤー	752
ボーンのレイヤーの表示/非表示	753
保護されたレイヤー	753
ボーンレイヤー	753
レイヤー間のボーンの移動	753
ボーンを隠す	753
ボーンのカンファクトリー	755
ボーンの関係の編集	755
ボーンのエンドを選択する	756
選択範囲を反転	756
接続されたボーンのエンドを選択する	756
ボーンを選択	756
接続されているボーンを選択解除	757
ボーンを追加	759
追加メニュー	759
突き出し	759
Mouse Clicks	760
関節の間を埋める	761
複製	762
ボーンを削除	762
標準的な削除	762
マージ	763
ボーンの詳細化	763

ボーンのロック	764
X軸ミラー編集	764
新しいArmatureのボーンの分離	764
ボーンの変形	765
エンベロープの可視化における半径とスケーリング	765
Bとエンベロープのスケール	765
ボーンの方角	766
ボーン回転	766
プロパティ	767
ボーンの剛性の設定	767
チェーンの編集	768
ボーン命名	769
命名規則	769
ボーンの名前のフリッピング	770
ボーン自動ネーミング	770
チェーンを描く	771
ストレートセグメント	771
フリーセグメント	772
ストロークの選択	772
ストロークの消去	772
ストロークの修正	772
ジェスチャ	772
カット	773
消去	773
反転	773
ボーンに変換する	773
Fixed	773
Adaptative	774
Length	774
Template	774
Retarget	774
ポーズセクションの概要	776
関連項目	776
Colors	777
Coloring from Bone State	777
Coloring from Bone Group	777
Ghosts	778
Motion Paths	779
ボーン選択	782
ポージングの基礎	782
自動IK (Auto IK)	783
休止ポーズ (Rest Pose)	783
中間ポーズ (In-Betweens)	783
ポーズのコピーとペースト	783
ボーン関係性の影響	784
Intro	786
Pose Library Panel	786
Editing	786
Automatic IK	789
IK拘束	789
スプラインIK	789
基本設定	789
設定とコントロール	789
ロールの制御	789
オフセット制御	789
厚さのコントロール	789
良好にセットアップするためのヒント	789
基本的なセットアップ	791
設定とコントロール	791
Roll Control	791
Offset Controls	791
Thickness Controls	791
セッティングのヒント	791
概要	792
拘束の使用	792
ボーン拘束	792
使用可能な拘束	792

Transform Constraints	792
Tracking Constraints	792
Relationship Constraints	793
拘束のヘッダ	794
拘束の設定	794
ターゲット	794
拘束空間 (Space)	795
影響度	795
拘束の追加/削除	796
概要	797
オプション	797
オプション	798
概要	799
オプション	799
概要	800
オプション	800
概要	801
オプション	801
概要	802
オプション	802
概要	803
オプション	803
概要	804
オプション	804
概要	805
オプション	805
See also	805
概要	806
オプション	806
概要	808
オプション	808
概要	809
オプション	809
概要	810
オプション	810
概要	811
オプション	811
概要	812
オプション	812
See also	812
Options	813
概要	814
オプション	814
Description	815
オプション	815
Notes	815
Description	817
オプション	817
Tips	817
例	817
概要	819
オプション	819
概要	820
オプション	820
概要	822
オプション	822
Example	822
See also	822
Description	823
オプション	823
概要	825
オプション	825
Chapters	826
キーフレームの作成	827
3Dビューでの作成	827
プロパティパネルでの作成	827
アニメーション用エディタでの作成	827
キーフレームの編集	828

Actions	829
F-Curve Interpolation	829
Basic Animation	829
Insert Keyframes	830
Auto Keyframe	830
Keying Sets	830
Properties	831
Editing	831
Working with Actions	832
マッピング	833
ドライバの作成	833
ドライバの編集	833
ドライバプロパティ	833
ドライバの種類	833
変数	833
単一プロパティ	834
座標変換チャンネル	834
例	834
回転の差	835
距離	835
式ドライバ	835
Valid Expressions	835
Builtin resources and aliases	836
The pydrivers.py Blender text	836
Example	836
Links	837
ポーズマーカー	838
視覚化	838
標準	838
シーケンサ	838
3Dビュー	838
マーカーの作成と編集	838
マーカーの作成	838
マーカーの選択	839
マーカーの命名	839
マーカーの移動	839
マーカーの複製	839
マーカーの削除	839
Keyframe Visualization	840
Motion Paths	840
Options	840
タイムラインの要素	843
カレントフレーム	843
キーフレーム	843
マーカー	843
ビューの調整	843
Viewメニュー	844
編集	844
Frameメニュー	844
再生	844
Playbackメニュー	844
ヘッダ	845
Curve Editor Area	847
Navigation	847
2D Cursor	847
View Axes	848
Markers	848
Header	848
Menus	848
Header Controls	848
Channels Region	849
Channel Editing	850
Properties Region	850
View Properties Panel	850
Active F-Curve Panel	851
Active Keyframe Panel	851
Drivers Panel	852
Modifiers Panel	852

参考	852
補間のコンセプト	853
補間の種類	853
時間の進む方向	853
カーブの選択	853
選択ツール	854
カーブの編集	854
座標変換	854
コピー	854
削除	854
スナップ	854
反転	855
クリーニングとスムージング	855
補間と補外	855
補間	855
補外	856
ハンドルの種類	856
モディファイアの追加	857
モディファイアの種類	857
Generator	857
Built-in Function	857
Envelope	857
Cycles	857
Noise	858
Python	858
Limits	858
Stepped	858
Dope Sheet Modes	859
Interface	859
View Menu	860
Select Menu	860
Marker Menu	860
トラック	861
ストリップ	861
アクションストリップの作成	861
トランジションストリップの作成	861
メタストリップの作成	861
ストリップの編集	861
ストリップの再実体化	862
ストリップのプロパティ	862
ストリップの名前変更	862
Active Track	862
Active Strip	862
Active Action	863
Evaluation	863
Strip Modifiers	863
コンストレイントによるアニメーションの制御	864
例	864
コンストレイントの影響をアニメートする	864
Parenting Method	865
The Follow Pathコンストレイント	865
The Clamp Toコンストレイント	865
Gravity	866
フォースフィールドの作成	867
共通の場設定	867
減衰 (Falloff)	867
場のタイプ	868
力 (Force)	868
風 (Wind)	868
渦 (Vortex Field)	868
磁石 (Magnetic)	868
調和 (Harmonic)	868
チャージ (Charge)	869
レナードジョーンズ (Lennard-Jones)	869
テクスチャ (Texture field)	869
例	869
カーブガイド (Curve Guide)	869
ボイド (Boid)	871

乱流 (Turbulence)	871
空気抵抗 (Drag)	871
リンク	871
オプション	872
ソフトボディおよびクロスの衝突	873
力場の相互作用	873
例	873
ヒント	873
作業の流れ	874
パーティクルシステムの作成	874
パーティクルシステムの種類	875
共通オプション	875
リンク	875
典型的な使い方	876
柔体の作成	876
シミュレーション品質	877
キャッシュとバイク	877
リアルタイムなインタラクション	877
ヒント	878
リンク	878
bouncing cube	879
result	879
解説	881
作業の流れ	881
クロスシミュレーションの生成	881
クロスパネル	881
素材	882
減衰	882
ピン止め	882
衝突	882
衝突の設定	882
クロス自身との衝突	883
共有されたレイヤー	883
メッシュオブジェクトの衝突	883
クロスとオブジェクトの衝突	883
メッシュオブジェクトのモディファイアスタック	883
クロス キャッシュ	884
衝突をバイクする	884
シミュレーションのキャッシュを編集する=	884
トラブルシューティング	884
例	884
成形のためにシミュレーションを使う	884
クросを滑らかにする	885
アマチュアとクロス	885
クrossをソフトボディとして使う	885
クロスと風	885
解説	886
作業の流れ	886
さらに詳しく	886
ドメイン (Domain) オブジェクト	887
オプション	887
Domain World	888
Domain Boundary	889
Domain Particles	889
バイク	890
バイクボタン (BAKE)	890
バイクディレクトリ	890
注意	890
発生源 (Inflow)	894
吸収源 (Outflow)	894
解説	897
例	897
オプション	897
関連項目	897
物理的正確さ	900
謝辞	901

制限と解決策	902
関連項目	902
外部リンク	902
開発ノート	903
内部処理	903
作業の流れ	903
ドメインの作成	904
共通オプション	904
煙グループオプション	905
高解像度オプション	905
フィールドウエイトオプション	905
フローオブジェクトの作成	906
設定	906
力 (Forces)	907
マテリアルの作成	908
テクスチャの追加	909
煙シミュレーターの拡張: 炎	910
トラブルシューティング	913
外部リンク	913
モディファイアの追加	914
タイプ	914
その他役に立つページ	914
メインパネル	915
Source パネル	915
Mesh Volume	915
Proximity	915
Mesh Volume + Proximity	916
Object Center	916
Particle System	916
Velocity パネル	916
Waves パネル	917
メインパネル	918
Advanced パネル	918
共通するオプション	918
Paint	918
Displace	919
Waves	919
Weight	920
Output パネル	920
Effects パネル	920
Cache パネル	920
はじめに	923
利用方法	923
手動2Dトラッキング	923
グリースペンシルやグリッドを使用した手動レンズキャリブレーション	923
カメラモーションの解析	923
シーン配置と安定化用基本ツール	923
実写映像にシーンの合成を行う基本的なノード	923
未実装のツール	923
マニュアル	923
Movie Clip エディタ	923
Tracking モードで利用可能なツール	925
Markers パネル	925
Track パネル	925
Solve パネル	925
Cleanup パネル	925
Clip パネル	926
Tracking モードで使用可能なプロパティ	926
Grease Pencil パネル	926
Objects パネル	926
Track パネル	926
Camera Data パネル	927
Display パネル	927
Tracking Settings パネル	928
共通オプション	928
KLT tracker オプション	928
SAD tracker オプション	928
Marker パネル	928
Proxy/ Timecode パネル	928
Reconstruction モードで利用可能なツール	929
レンダリング	930
概観	930
分散型レンダーファーム	930
レンダリング作業台の統合	930
Render 設定パネル	931
Render	931
Layers	931
Dimensions	931

Anti-Aliasing	931
Motion Blur	931
Shading	932
Output	932
Performance	932
Post Processing	932
Stamp	932
Bake	932
Freestyle	932
Add a new camera	933
Change active camera	933
Camera Presets	933
Depth of Field	934
Display	934
Composition Guides	935
Camera Navigation	935
Move active camera to view	935
Camera View Positioning	935
Roll, Pan, Dolly, and Track	935
Aiming the camera in Flymode	936
Three Point Rendering	937
Two Point Rendering	937
To achieve 2-point rendering:	938
One Point Rendering	938
To get 1-point (1pt) renders	938
Orthographic Rendering	939
Isometric Rendering	939
Field of View and Lens Size	941
Zooming in Blender	941
Depth of Field in Computer Graphics	941
Displaying Renders	943
Saving	943
Animation Playback	943
Display Options	943
Color Management and Exposure	945
Anti-Aliasing	945
Exposure (Lighting)	945
Depth of Field	945
Motion Blur	945
Options	946
Filtering	946
Filter Size	947
Examples	947
作業手順	949
フレーム分割法 (Frame Sequence) の作業手順	949
オプション	950
ヒント	950
Options	951
Bake Mode	951
Full Render	951
Ambient Occlusion	951
Shadow	951
Normals	952
Textures	952
Displacement	952
Emission	953
Alpha	953
Mirror Colors and Intensity	953
Specular Color and Intensity	953
Additional Options	953
Workflow	953
構文	955
レンダリングオプション:	955
ファイル形式のオプション:	955
アニメーション再生のオプション:	955

ウインドウのオプション:	956
Game Engine 専用のオプション:	956
Python のオプション:	956
デバッグのオプション:	956
その他のオプション:	956
その他のオプション:	957
実行例	957
静止画のレンダリング	957
動画のレンダリング	957
指定エンジンで Blender を起動する	957
プラットフォーム	958
Windows	958
Mac OS X	958
Linux	958
Dimensions	959
Presets	959
Output Panel	959
File Locations	959
File Type	960
Compression	960
Channels	960
Image Formats	960
VSE Rendering	961
Rendering to an Image Sequence	961
Rendering to Video	961
Dimensions Presets	962
Pixel Aspect Ratio	962
Colour Saturation	962
Rendering to fields	963
Options	963
Video Files	964
Advanced Encoding	964
Video Settings	965
Video Formats	965
Video Codecs	966
Options	966
Audio Settings	966
Tips	967
Home-made Render Farm	967
Options	969
Examples	969
Dithering	970
シーンリニア(Linear)色空間	972
表示	972
レンダー	972
画像ファイル	973
シーケンサー(Sequencer)	973
OpenColorIO 設定	973
互換性	974
Developer Documentation	974
The Human Eye	976
In Film	976
In CG	976
Motion Blur in Blender	976
Sampled Motion Blur	976
Vector Blur	976
Examples	976
Hints	977
Hardware Improvements	978
Operating System Configuration	978
Blender Settings	978
Scene and Specific Objects	978
Render Settings	979
Multi-Pass Compositing	979
Multi-Threading	981
Frame Ranges	981
Collaborative Rendering	981
Remote Renderfarms	981

目的	982
使い方	982
GUI	982
マスター/Master	982
マスターのウェブインターフェイス	982
スレーブ/Slave	982
クライアント/Client	982
コマンドライン	982
その他	983
設定	983
マスター/Master	983
スレーブ/Slave	983
クライアント/Client	984
物理演算のバイク処理	985
バージョン管理ジョブ	985
覚書および既知の不具合	985
負荷の分散/Load Balancing	985
例外	986
最優先(基準)	986
To do	986
技術的詳細	986
機能一覧	986
API 機能のウィッシュリスト	986
使い方	988
チュートリアル(英文)	988
Reference	988
ワークフロー	990
既知の制限と問題	990
使い方	991
チュートリアル(英文)	991
Reference	991
パースペクティブ(Perspective)	995
正射投影(Orthographic)	995
被写界深度(Depth of Field)	995
Clipping	995
サーフィスシェーダ	996
ボリュームシェーダ	996
ディスプレイメント	996
エネルギー保存	996
種類	997
再分割	997
用語	998
BSDFのパラメータ	998
3Dビューの描画タイプ	1000
テクスチャのプロパティ	1000
ペインティングとUV編集	1000
サーフィスシェーダ	1001
ボリュームシェーダ	1001
ヒント	1001
Point Lamp	1002
Area Lamp	1002
Sun Lamp	1002
シェーダ	1003
テクスチャ	1003
その他	1003
BSDF	1004
Diffuse(拡散)	1004
Translucent(半透明性)	1004
Glossy(光沢)	1004
Transparent(透明性)	1005
Glass(ガラス)	1006
Velvet(ベルベット)	1006
Emission(放射)	1007
Background(背景)	1007
Holdout(抵抗)	1007
Mix and Add(合成と加算)	1007
Image(イメージ)テクスチャ	1009
Environment(環境)テクスチャ	1009
Sky(空)テクスチャ	1009
Noise(ノイズ)テクスチャ	1009
Wave Texture	1009
Voronoi(ボロノイ)テクスチャ	1010

Musgrave(マズグレイブ)テクスチャ	1010
Gradient(勾配)テクスチャ	1010
Magicテクスチャ	1010
Checker Texture	1010
Value(数値)	1012
RGB	1012
Geometry(ジオメトリ)	1012
Texture Coordinates(テクスチャ座標)	1012
Attribute(属性)	1012
Mapping(マッピング)	1012
Layer Weight(レイヤーウエイト)	1013
Fresnel(フレネル)	1013
Light Path(ライトパス)	1013
レイタイプ	1014
バウンスコントロール	1014
透過	1014
Ray Visibility	1014
シーンの設定	1015
Samples	1015
Bounces	1015
Transparency	1015
Caustics	1015
マテリアルの設定	1015
ワールドの設定	1015
パス・トレーシング Path tracing	1016
ノイズはどこから来るのか	1016
跳ね返り Bounces	1016
集光と艶フィルター Caustics and Filter Glossy	1017
ライトの減衰 Light Falloff	1017
ランプとして採取 Sample as Lamp	1017
ガラスと半透明の影 Glass and Transparent Shadows	1018
窓型ライト Window Lights	1018
蛍ノイズの固定 Clamp Fireflies	1018
はじめに	1021
Userプリファレンスによる設定	1021
CUDA	1021
古いカード	1021
OpenCL	1021
よくある質問と答え	1021
レンダリング中、Blender が無反応になるのはなぜ？	1021
なぜCPU上でレンダリングされたシーンがGPU上ではレンダリングされないのでしょうか？	1021
複数の GPU をレンダリングに使用できますか？	1021
複数の GPU では利用可能なメモリ量は増えますか？	1021
NVidia と AMD、CUDA と OpenCL ではどれがレンダリングが高速ですか？	1022
エラーメッセージについて	1022
Unsupported GNU version! gcc 4.5 and up are not supported!	1022
CUDA Error: Invalid kernel image	1022
CUDA Error: Out of memory	1022
The NVIDIA OpenGL driver lost connection with the display driver	1022
ワークフロー	1024
既知の制限と問題	1024
可視性による選択	1027
エッジタイプによる選択	1028
辺マーク(Edge Marks)	1029
面マーク(Face Marks)による選択	1030
グループによる選択	1031
画像境界(Image Border)による選択	1031
チェーン化(Chaining)	1033
分割(Splitting)	1033
選択(Selection)	1033
先端(Caps)	1034
破線(Dashed Line)	1034
モディファイアー(Modifiers)	1035
ストローク追従(Along Stroke)	1035
カメラからの距離(Distance from Camera)	1035
オブジェクトからの距離(Distance from Object)	1035
マテリアル(Material)	1036
モディファイアー	1037
ストローク追従(Along Stroke)	1037
カメラからの距離(Distance from Camera)	1037
オブジェクトからの距離(Distance from Object)	1038
マテリアル(Material)	1038
モディファイアー	1039

ストローク追従(Along Stroke)	1039
カリグラフィー(Calligraphy)	1039
カメラからの距離(Distance from Camera)	1040
オブジェクトからの距離(Distance from Object)	1040
マテリアル(Material)	1041
モディファイアー	1042
2Dオフセット(2D Offset)	1042
2Dトランスフォーム(2D Transform)	1042
バックボーン拡張(Backbone Stretcher)	1043
ベジエカーブ(Bezier Curve)	1043
設計図(Blueprint)	1044
ガイドライン(Guiding Lines)	1044
パーリンノイズ1D(Perlin Noise 1D)	1044
パーリンノイズ2D(Perlin Noise 2D)	1045
ポリゴン化(Polygonization)	1045
サンプリング(Sampling)	1045
Sinディスプレイスメント(Sinus Displacement)	1045
空間ノイズ(Spatial Noise)	1046
先端削除(Tip Remover)	1046
Nodes Concepts	1047
Nodes	1047
Noodles	1047
Node Groups	1047
Accessing and Activating Nodes	1047
Examples	1048
Accessing The Node Editor	1049
Activating Nodes	1049
Node Editor Window Actions	1050
Node Editor Header	1050
At a glance	1050
Menus	1050
View, Select and Add	1051
Node	1051
Buttons	1051
Material/Composite/Texture Selector	1051
Use Nodes Button	1051
Use Pinned Button	1051
Go to Parent Button	1051
Snap Button	1052
Copy Nodes Button	1052
Paste Nodes Button	1052
Layout Nodes	1052
Free Unused Button	1053
Backdrop	1053
Auto Render	1053
Collapsing toggles	1054
Sizing the node	1055
Sockets	1055
Curves	1055
RGB Curves	1056
Selecting curve points	1056
Editing curves	1056
Editing the view	1056
Special tools	1056
Adding Nodes	1057
Arranging Nodes	1057
Connecting nodes	1057
Disconnecting nodes	1057
Duplicating a node	1057
Grouping Nodes	1058
Editing Node Groups	1058
Ungrouping Nodes	1058
Appending Node Groups	1058
Using the Sequence Editor	1061
Adding Strips	1061
Add Movies or Images	1062
Add Scene	1062
Add Audio	1063
Adding Effects	1063
Adding Plugin Effects	1063
Strip Properties	1064
Edit Strip Panel	1064
Effect Strip	1064
Strip Input	1064
Filter	1064
Proxy Strip Properties Panel	1065
Sound	1065

Scene	1065
Adjusting the View	1065
Scrubbing	1066
View Modes	1066
Scene Preview	1066
View Settings	1066
Refresh View	1066
Selecting Strips	1066
Moving and Modifying Strips	1067
Snap to Frame	1067
Separate Images to Strips	1067
Editing Strips	1067
Copy and Paste	1068
Meta Strips	1068
Image Preview	1069
Luma Waveform	1069
Chroma Vectorscope	1070
Histogram	1071
[edit] Add	1073
[edit] Subtract Effect	1073
[edit] Cross and Gamma Cross	1073
[edit] Fade to Black	1074
[edit] Multiply	1074
[edit] Alpha Over, Under, and Over Drop	1074
[edit] Wipe	1075
[edit] Glow	1076
[edit] Transform	1076
[edit] Color	1077
[edit] Speed Control	1077
[edit] Creating a Slow-Motion Effect	1077
[edit] Keyframing the Speed Control	1077
[edit] Changing Video Frame Rates	1077
[edit] Multicam Selector	1078
[edit] Adjustment Layer	1078
[edit] Options	1079
[edit] Audio Mixing in the VSE	1079
[edit] Working with Audio Tracks	1079
[edit] Animating Audio Track Properties	1079
[edit] Output	1080
アドオン/Addons	1081
スクリプト/Scripts	1081
独自のスクリプトの保存	1081
ファイルの場所	1081
インストール	1081
前置き	1082
一般情報	1082
はじめに - Wiki チュートリアル	1082
はじめに - 外部リンク	1082
アドオンのインストール	1083
2.5 におけるファイルの場所	1083
有効化と無効化	1083
開発ガイドライン	1083
プログラミングって何？	1084
Python って何？	1084
Python インタプリタ	1084
Hello World	1084
外部リンク	1085
ウェブページ	1085
動画チュートリアル	1085
内蔵された Python コンソールにアクセスする	1086
初めてのコンソール環境	1086
オート・コンプリート機能	1086
モジュールで試行錯誤する前に	1087
例	1087
bpy.context	1087
やってみよう!	1087
bpy.data	1088
やってみよう!	1088

エクササイズ	1088
bpy.ops	1088
やってみよう!	1089
テキストウィンドウの他の用途	1090
デモンストレーション	1090
エクササイズ	1090
Blend ファイル内のスクリプト	1092
自動実行	1092
手動実行	1092
スクリプト実行のコントロール	1092
デフォルト設定	1092
コマンドライン	1093
スクリプト	1094
Game Engine を利用する	1095
Logic Bricks	1098
Properties	1098
States	1098
1 Properties	1099
2 Associated Object(s)	1099
3 Links	1099
4 Sensors	1099
5 Controllers	1099
6 Actuators	1100
Column Heading	1102
Object Heading	1102
コントローラの種類	1120
Column Heading	1121
Object Heading	1121
例	1123
オプション	1123
例	1124
オプション	1124
例	1125
オプション	1125
例	1126
オプション	1126
例	1127
オプション	1127
例	1128
オプション	1128
変数	1129
演算	1129
算術演算	1129
論理演算	1129
条件文 (if)	1129
例	1129
Script Mode	1131
Module Mode	1131
Column Heading	1133
Object Heading	1133
Motion Blur	1136
Built-In 2D Filters	1137
Custom Filters	1137
オプション	1140
関連項目	1140
種類	1143
使用する際の注意	1143
使用する際の注意	1144
Simple Motion	1145
Servo Control	1145
例	1148
処理	1154
Usage Notes	1154
使用する際の注意	1155
プロパティパネル	1156
プロパティの種類	1156
プロパティを使う	1156

Property Sensor	1156
Property Actuator	1157
How States Operate	1159
Editing States	1159
デフォルトのカメラ	1161
オブジェクトを親にする	1161
頂点を親にする	1161
オブジェクトをカメラにする	1161
レンズシフト	1161
関連項目	1161
Game	1162
Standalone Player	1162
Stereo (立体視)	1163
Shading	1163
Performance	1163
Display	1163
Stereo Settings	1165
Stereo Modes	1165
Quad Buffer	1165
Above-Below	1165
Interlaced	1165
Anaglyph	1165
Side by Side	1165
Vinterlaced	1165
Dome Camera Settings	1166
Fisheye Mode	1166
Front-Truncated Dome Mode	1167
Rear-Truncated Dome Mode	1167
Cube Map Mode	1168
Spherical Panoramic	1168
Warp Data Mesh	1168
サンプルファイル	1169
World	1170
Mist	1170
Game Physics	1170
Obstacle Simulation	1171
オプション	1172
Actor	1173
Ghost	1173
Invisible	1173
Radius	1173
Anisotropic (異方性)	1173
設定	1175
Attributes	1175
Velocity (速度)	1175
Damping (減衰)	1175
Lock Translation and Rotation	1175
Collision Bounds	1175
Settings	1177
Attributes	1177
Cluster Collision	1177
Collision Bounds	1177
動作の仕組み	1178
使い方	1178
ゲーム内でのコントロール	1178
既知の制限	1179
設定	1180
Collision Bounds	1180
Options	1182
Options	1183
動作の仕組み	1186
ゲームの準備	1186
例	1186
マテリアルの取得	1186
テクスチャの作成	1186
テクスチャを永続的にする	1187
ソースを作成する	1187
ソースのセットアップ	1187

Play the video	1188
ビデオのステータスをチェックする	1188
高度なワークフロー	1188
デモのダウンロード	1189
高度なデモ	1189
Standalone Player License	1191
Distributing Games	1191

はじめに



Blender 2.5 with a Big Buck Bunny scene open

ようこそ、Blenderへ！ このBlenderの解説書は、ユーザマニュアル、リファレンスガイド、チュートリアル、フォーラム、そして多くのウェブリソースというように、沢山のパートから成り立っています。このマニュアルでは、まず始めに、ブレンダーのダウンロードとインストールの方法や、ソースコードから実行ファイルを生成する方法から説明していきます。

Blenderのインターフェースは、3Dグラフィックス制作のために高度に最適化されていて、ちょっと普通ではありません。なので、初めて触れる人にとってはとっつきにくいかも知れませんが、じきにその威力が分かるでしょう。このインターフェースとこの解説書で使われている慣例に慣れるために、[インターフェース](#)を丁寧に読むことを強くお勧めします。

Blenderって何？

Blenderは、2Dと3Dの幅広いコンテンツ制作のための統合アプリケーションとして、1993年12月に着想され、1994年8月に使用可能なソフトウェアプロダクトとして誕生しました。Blenderは、モデリング、テクスチャリング、ライティング、アニメーション、そしてビデオポストプロセッシングの機能を一つのパッケージとして提供しています。オープンなアーキテクチャを用いているので、クロスプラットフォームであって、広範囲にわたる機能性があり、とても小さなファイルサイズでありながら、上手く統合されたワークフローを提供しています。Blenderは、世界で最も人気のあるオープンソースの3Dグラフィックスアプリケーションの一つとなっています。

Blenderは、世界中のメディア制作のプロやアーティストが使用することを目標としているので、3Dビジュアライゼーションだけでなく放送や映画クオリティの映像生成に用いることができます。一方で、リアルタイム3Dエンジンも備えているため単独で3Dインタラクティブコンテンツを制作することもできます。

Blenderは、もともとはNot a Number (NaN)という会社によって開発されていましたが、今は、GNU GPLライセンスの下、ソースコードを手に入れられる「フリーソフトウェア」となっており、オランダにあるBlender Foundationによって開発が続けられています。

Blenderは、2008年から2010年にかけて、機能やワークフロー、インターフェースを向上させるために完全に0から作り直されています。この作業の結果、Blender 2.5(現在はベータテストバージョン)が生み出されました。

主な特徴:



Rendering image and post-working on it

- 3Dコンテンツ制作のための高機能な統合ソフトウェア(モデリング、UVマッピング、テクスチャリング、リギング、スキニング、アニメーション、パーティクル等のシミュレーション、スクリプティング、レンダリング、コンポジティング、ポストプロダクション、ゲーム制作機能など)
- OpenGLによる統一されたGUIを用いたクロスプラットフォームソフトウェア(Pythonスクリプトによってカスタマイズ可能) Windows (98, NT, 2000, XP, Vista, 7), Linux, OS X, FreeBSD, Irix, Sun等のOSに対応
- 高速かつ高効率な制作ワークフローを実現する高品質な3Dアーキテクチャ
- 全世界で200,000件以上のダウンロード(各リリース毎)
- ユーザコミュニティによるサポート(質問や批評に関するフォーラム <http://BlenderArtists.org> ニュースサービス <http://BlenderNation.com>)
- 小さなファイルサイズ、簡単に頒布可能

最新バージョンのBlenderは[ここ](#)からダウンロードできます。

Blenderの歴史

1988年に Ton Roosendaal 氏はアニメーション・スタジオ *NeoGeo* を共同で設立しました。NeoGeoは瞬間に成長し、オランダ最大の3Dアニメーション・スタジオ、欧州をリードするアニメーション会社の1つになりました。NeoGeo は国際的な電器会社である Philips のような大企業に対して、数々の賞(例として1993年及び1995年の European Corporate Video Awards)を受賞した作品を提供しました。NeoGeo で Ton 氏はアートディレクションと内製ソフトウェア開発に携わっていました。Ton氏は、当時の NeoGeo 社が利用していた古く、維持と更新に手間のかかる自社製の3Dツール・セットについて注意深く検討し、ゼロから作り直す必要があると結論付けました。1995年にこの作業は開始され、現在私たちが知る 3D 総合製作ソフトウェア *Blender* の歴史が始まったのです。NeoGeo 社による Blender の更新作業が進む中で、Ton氏は Blender が NeoGeo 社外のアーティストに利用されるツールとなる可能性を感じるようになりました。

1998年、Ton氏は市場開拓と Blender の開発を促進するために *Not a Number* (以下 *NaN*) という新会社を NeoGeo 社から分業独立させて設立しました。NaN の目標は、コンパクトでクロス・プラットフォーム対応の3D総合開発環境をフリーで提供するという事でした。殆どの商用モデリングソフトウェアが数千(アメリカ)ドルだった当時、これは革命的なコンセプトでした。NaN はプロフェッショナル水準の 3Dモデリング・アニメーションツールを広く一般の人々が利用可能なコンピュータで動作させることを目指しました。NaN のビジネスモデルは Blender 向けの商品やサービスを提供に携わるという物でした。

1999年に NaN は Blender をより広く認知してもらうために Siggraph に初めて参加しました。Blender の初めての Siggraph への参加は大きな成功を収め、プレス・参加者双方から並々ならぬ注目を集めました。成功を収めたことで Blender の大きな可能性が明らかになったのです！

2000年に開催された Siggraph でも成功を収めた事を受け、NaN はベンチャーキャピタルから450万ユーロの投資を確約されました。この莫大な資金が NaN の急速な事業の拡大を可能にしました。ほどなくして NaN は世界中で50名もの従業員を雇用して Blender の改良と利用促進に乗り出しました。2000年の夏には Blender v.2.0 がリリースされました。このバージョンの Blender ではゲームエンジンが組み込まれました。2000年の終盤には、NaN に登録しているユーザーの数が25万人を超えました。

しかし不運にも、NaNの野心及び恵まれた機運と、当時の NaN の現実的な市場価値や企業の限界との間には差が生じていました。行き過ぎた事業の拡大は、NaN が新たに募った投資家からの投資によって、事業を縮小して2001年の春に再出発するという結果を招きました。その6ヶ月後、NaN は初の商業ソフトウェアである *Blender Publisher* が世に送り出されました。この商品は、インタラクティブなWebベースの3Dメディアの市場を刺激する事を目的としていました。ですが売り上げが期待された程ではなかった事と、進行していた景気の悪化を受け、新たに募った投資家らは NaN の業務停止を決定しました。この業務停止の内容には Blender の開発中止も含まれていました。しかしながらその決定は当時の、複雑な内部構造を持ち、未完成の機能と非凡なGUIを提供し、ユーザーのコミュニティと過去に Blender Publisher を購入した利用者による熱狂的な支持を得ていた Blender にとって明らかに不利益となるものであり、Ton氏は Blender をただ記憶から消え行く存在としておく事ができませんでした。潤沢な人数を備えた開発者チームを持った企業として再出発する事が可能ではなかったため、Ton Roosendaal は非営利団体 *Blender Foundation* を2002年3月に設立しました。

Blender Foundation の第一の目標は コミュニティによる [オープンソース](#) のプロジェクトとして Blender の開発の継続と利用の促進の方法を模索する事でした。2002年7月、Ton氏は NaN の投資家らから Blender Foundation だけが Blender をオープンソース化して提供する事に対する合意を取り付けました。熱心なボランティアの人々と NaN の一部の元従業員達の努力によって、“Free Blender” という資金調達のためのキャンペーンが行われました。この“Free Blender”のキャンペーンは Blender Foundation が Blender のソースコードと知的財産権を NaN の投資家らから買い上げ、オープンソース・コミュニティに提供する権利を得るために10万ユーロを集めました。皆が驚き、感激したことにそのキャンペーンは10万ユーロという目標をわずか7週間弱という期間で達成してしまいました。そして、2002年12月13日の日曜日に Blender は [GNU General Public License \(GPL\)](#) に基づいて全世界に公開されました。Blender の開発は今日まで、作者である Ton Roosendaal 氏が引導する、世界中からの献身的なボランティアの方々によって続けられています。

Video: From Blender 1.60 to 2.50

[\[video link\]](#)

Version/Revision Milestones

Blender's history and road-map:

- 1.00 – 1995年1月: アニメーション・スタジオ NeoGeo で Blender が開発される。
- 1.23 – 1998年1月: SGI バージョンがWebで公開、IrisGLを利用。
- 1.30 – 1998年4月: Linux 及び FreeBSD バージョンをリリース、利用環境を OpenGL と Xウィンドウシステムに変更。
- 1.3x – 1998年6月: Not a Number が設立される。
- 1.4x – 1998年9月: Sun 及び Linux Alpha バージョンをリリース。
- 1.50 – 1998年11月: マニュアル(第1版)が刊行される。
- 1.60 – 1999年4月: C-Key (ロックされた新機能、95ドルで販売)を搭載、Windows バージョンをリリース。
- 1.6x – 1999年6月: BeOS 及び PPC バージョンをリリース。
- 1.80 – 2000年6月: C-key終了、Blender は再び完全なフリーウェアに。
- 2.00 – 2000年8月: インタラクティブ3D及びリアルタイム・エンジンを搭載。
- 2.10 – 2000年12月: 新型エンジン、物理エンジンを搭載、利用環境を Python に変更。
- 2.20 – 2001年8月: キャラクター・アニメーション・システムを搭載。
- 2.21 – 2001年10月: Blender Publisher 販売開始。
- 2.2x – 2001年12月: Mac OSX バージョンリリース。
- **2002年10月13日: Blender がオープンソース化。Blender Conference 初開催。**
- 2.25 – 2002年10月: [Blender Publisher](#) 無償での入手可能に。
- Tuhopuu1 – 2002年10月: コーダーによって試験的に Blender の tree が製作される。
- 2.26 – 2003年2月: オープンソースとしての初の Blender のリリース。
- 2.27 – 2003年5月: オープンソースとしての2回目の Blender のリリース。
- 2.28x – 2003年7月: バージョン2.28台の初のリリース
- 2.30 – 2003年10月: [Preview release](#)によって刷新されたバージョン2.3台のUIが第2回目の Blender Conference で発表される。
- 2.31 – 2003年12月: [Upgrade](#)によってバージョン2.3台のUIの安定版がリリース。

- 2.32 – 2004年1月: [Major overhaul](#)によって内部レンダリングが可能に。
- 2.33 – 2004年4月: [Game Engine returns](#)リリース、環境自然光(ambient occlusion)、procedural textures対応。
- 2.34 – 2004年8月: [Big improvements](#): particle interactions、最小二乗法による等角マップ(LSCM)による UV マッピング、YafRay の内蔵、subdivision surfaces での weight 付き crease 、ランプシェーダ、full OSA他多数の機能搭載。
- 2.35 – 2004年11月: [Another version full of improvements](#): object hooks, curve deforms and curve tapers, particle duplicators and much more.
- 2.36 – 2004年12月: [A stabilization version](#), much work behind the scene, normal and displacement mapping improvements.
- 2.37 – 2005年6月: [A big leap](#): transformation tools and widgets, softbodies, force fields, deflections, incremental subdivision surfaces, transparent shadows, and multithreaded rendering.
- 2.40 – 2005年12月: [An even bigger leap](#): full rework of armature system, shape keys, fur with particles, fluids and rigid bodies.
- 2.41 – 2006年1月: [Lots of fixes](#), and some game engine features.
- 2.42 – 2006年7月: [The Node release](#). Over [50 developers](#) contributed nodes, array modifier, vector blur, new physics engine, rendering, lipsync and, many other features. This was the release following [Project Orange](#).
- 2.43 – 2007年2月: [The Multi release](#): multi-resolution meshes, multi-layer UV textures, multi-layer images and multi-pass rendering and baking, sculpting, retopology, multiple additional matte, distort and filter nodes, modeling and animation improvements, better painting with multiple brushes, fluid particles, proxy objects, sequencer rewrite, and post-production UV texturing. whew! Oh, and a website rewrite. And yes, it still has multi-threaded rendering for multi-core CPUs. With Verse it is multi-user, allowing multiple artists to work on the same scene collaboratively. Lastly, render farms still provide multi-workstation distributed rendering.
- 2.44 – 2007年5月: [The SSS release](#): the big news, in addition to two new modifiers and re-awakening the 64-bit OS support, was the addition of subsurface scattering, which simulates light scattering beneath the surface of organic and soft objects.
- 2.45 – 2007年9月: [Another bugfix release](#): serious bugfixes, with some performance issues addressed.
- 2.46 – 2008年5月: [The Peach release](#) was the result of a huge effort of over 70 developers providing enhancements to the core and patches to provide hair and fur, a new particle system, enhanced image browsing, cloth, a seamless and non-intrusive physics cache, rendering improvements in reflections, AO, and render baking; a mesh deform modifier for muscles and such, better animation support via armature tools and drawing, skinning, constraints and a colorful Action Editor, and much more. It was the release following [Project Peach](#).
- 2.47 – 2008年8月: [Bugfix release](#).
- 2.48 – 2008年10月: [The Apricot release](#): cool GLSL shaders, lights and GE improvements, snap, sky simulator, shrinkwrap modifier, python editing improvements.
- 2.49 – 2009年6月: [The Pre-Re-Factor release](#) added significant enhancements to the core and GE. Core enhancements include node-based textures, armature sketching (called Etch-a-Ton), boolean mesh operation improvements, JPEG2000 support, projection painting for direct transfer of images to models, and a significant Python script catalog. GE enhancements included video textures, where you can play movies in-game (!), upgrades to the Bullet physics engine, dome (fish-eye) rendering, and more API GE calls made available.
- 2.5 – From 2008 to 2010年9月. This series [release 4](#) pre-version (from Alpha0 - 2009年11月 - to Beta 2010年7月. It is one of the most important development [project](#) of blender with a total re-coding of the software with new functions, redesign of internal window manager and event/tool/data handling system, new python API... The final version of this project will be Blender 2.6 (planned to 2010年7月).

フリーソフトウェアとGPLについて



「フリーソフトウェア」について真っ先に思い浮かぶ特徴は「費用がかからない」事ではないでしょうか。それは多くの場合で正しいのですが、GNU プロジェクトの創設者らと GNU General Public License の起草者らからなる、フリーソフトウェア財団 (Free Software Foundation, FSF) によって用いられる「フリーソフトウェア」という言葉は「費用がかからない」という意味よりもむしろ「自由である」という (例えば "free beer" と表現された時に「無料のビールを飲め」ではなく「自由にビールを飲め」ととらえる) 意味に重きを置いています。この意味の元で言うフリーソフトウェアとは一般の人々が自由に利用、複製、改変及び再配布を制限なく行えるという物です。対照的に、多くの商業ソフトウェアパッケージのライセンスでは、1つのパッケージに対して1つのコンピュータのみでの起動が許可され、ソフトウェアの複製は許可されず、ソースコードを見ることもできません。フリーソフトウェアはエンドユーザーに対し、想像を超えた自由を提供します。加えて、ソースコードが全世界に公開されているため、バグを発見して修正する多くの機会に恵まれています。

GNU General Public License (以下GPL)に基づくライセンスが適用されたプログラムでは:

- あなたはそのプログラムを目的を問わず利用できる権利を得られます。
- あなたはそのプログラムを修正し、ソースコードを取得する権利を得られます。
- あなたはそのプログラムを複製して再配布する権利を得られます。
- あなたはそのプログラムを改善し、あなた自身のバージョンとして公開する権利を得られます。

これらの権利を得る代わりに、あなたは GPL に基づくライセンスが適用されたプログラムを配布する場合に、あなた自身と他者の自由を守るために以下に示すいくつかの責任を負うことになります:

- あなたは配布するプログラムに、その利用者がライセンスに基づく権利を認識できるように、GPL のコピーを付加しなければなりません。
- あなたは配布するプログラムにソースコードを付加するか、そのソースコードを自由に得られるようにしなければなりません。
- あなたが、そのコードを修正して修正版を配布する場合はその修正部分を GPL に基づくライセンスで提供し、修正したソースコードを自由に得られるようにしなければなりません。(あなたはプロプライエタリなプログラムに GPL が適用されたコードを利用する事はできません。)
- あなたは GPL によって定められた条件を弱めるライセンスを、配布するプログラムに対して適用できません。(あなたは GPL が適用されたプログラムを、プロプライエタリな商品として提供できません。)

GPL に関する詳細は [GNU Project Web site](http://www.gnu.org) を閲覧して下さい。GNU General Public License のコピーは Volume II に含まれています。

サポート: Blender のコミュニティ

Blender の無償で入手可能であったという当初からの特徴は、ソースコードが公開されていなくとも、Blender の利用者の増加に大きく貢献しました。Blender の利用者のための、規模が大きく、安定した、活発なコミュニティが1998年から形成されてきました。

2002年の夏の終わりごろに Blender が GNU GPL に基づいたオープンソースのフリーソフトウェアとして公開された時には、コミュニティが最盛期を迎えました。

このコミュニティは現在では以下の2つの、広範にわたって共通しているサイトに分かれています:



[Blender Foundation site](http://www.blender.org) を中心として形成された開発者コミュニティです。ここは開発プロジェクト、機能とドキュメントについての掲示板、Blender のソースコードを保持している CVS レポジトリ、全てのドキュメントのソース、そしてそれらに関連した議論を行うフォーラムを持つホームとなっています。Blender のコーディングを行う開発者や、Python プログラム、ドキュメントの執筆者、さらに Blender 開発のために努力している全ての人々がここに集います。

[BlenderArtists](http://www.BlenderArtists.org) という独立したサイトを中心として形成された利用者コミュニティです。ここでは Blender を利用したアーティスト、ゲーム開発者や Blender のファンらが自身の創作を公開し、フィードバックを得て Blender の機能について理解を深めるための助言を得ています。加えて Blender のチュートリアルや役立つ情報をまとめたページ等があります。

参照 <http://www.BlenderArtists.org>

参照 <http://www.blender.org>

これら2つのウェブサイトのみが Blender の情報源ではありません。世界中のコミュニティが、地域毎の言語や一部の話題に特化した多数の独立したサイトを作成しています。日常的に更新される Blender の情報源となるサイトのリストは上記のサイトで見つける事ができます。

IRC チャットチャンネル

オンラインでの迅速なフィードバックを得るために、3つの IRC チャットチャンネルが恒久的に irc.freenode.net 上に公開されています。あなたは任意の IRC クライアントでこれらのチャンネルに参加できます。:

- [#blenderchat](#) Blender に関する一般的な議論を行うためのチャンネル。
- [#blenderqa](#) Blender を利用する上での質問のためのチャンネル。
- [#gameblender](#) ゲームエンジンを含む Blender でのゲーム開発に関する議論を行うためのチャンネル。

開発者向け IRC チャットチャンネル:

- [#blendercoders](#) 開発者による質問と開発上の問題について議論するためのチャンネルであり、オランダ時間(CEST)の毎週日曜午後4時からミーティングを行っている。
- [#blenderpython](#) Python API 及びスクリプト開発に関する議論を行うためのチャンネル。
- [#blenderwiki](#) wiki を編集する上での質問のためのチャンネル

どんな人が Blender を利用するのか?

Blender の新しいバージョンがリリースされると、リリースされてからほんの10日以内に世界中で100万人以上の人々にダウンロードされます。これは全てのプラットフォーム (Windows, Linux そして MacOS) を計算に含み、完全にオーソライズされて制限が無い再配布版は含まれていません。私たちは200万人以上の利用者が存在すると推定しています。このマニュアルは Blender を利用する多彩な才能に恵まれた人々のために書かれています。:

- コンピュータグラフィックス (CG) および3Dアニメーションの世界を探求したいと思っている、趣味としての利用者や学生
- 各々のアートやポスターを作成する、あるいは画像処理を施して作品の品質を向上させる2-Dアーティスト
- テレビジョン向けコマーシャルや短時間の映像のための漫画や風刺画("The Magic of Amelia" のような作品)を製作する2-Dアーティストあるいはチーム
- 時にライブ・アクションも利用する短編CGアニメーション("Suburban Plight" のような作品)を製作する3-Dアーティスト個人および団体。
- 完全にCGで製作された長編アニメーション作品("Elephant's Dream" 、 "Plumiferos" 、 "Big Buck Bunny" 等の作品)を製作する3-Dクリエイターの団体。
- 一部にCGを含んだライブ・アクションの映像作品を共同で製作している3Dクリエイターの団体。

10代からお年を召された方々まで幅広い年齢層の人々が Blender を使い、ユーザーコミュニティは初心者から、時に企業での利用を行うプロフェッショナルの人々向けまで公正かつ均等に分類されています。私達は2-Dあるいは3-Dの映像作品やアニメーションを行う団体を、さらに細分化した業務別に分けて考える事ができます。Blender の利用者には以下に示す人々が含まれます:

- ディレクター - 利用するシーン、シーン中で必要なアクション(アニメーション)の選択を行います。シーン中のカメラの割り振りも決定します。
- モデラー - キャラクター、演出用のパーツ、風景や舞台に特化したヴァーチャル・リアリティの製作を担います。
- カメラマン、映像監督 (Director of Photography, DP) - カメラの設定と動作、ライブ・アクションの撮影、映像出力を担います。
- マテリアル・ペインター - セット、出演者等動く物全てに描画を行います。動作しないものであっても描画します。
- アニメーション及びリギング担当 - Armature を利用して物に動きを与えます。
- 光源及び色彩担当 - 舞台やセットを照らし、照明に応じた適切な色彩調整を行い、マテリアルやシーン、テクスチャに汚しの効果を付加します。
- 特殊効果担当 - 流体シミュレーション、モーション・キャプチャー、クロス・シミュレーション、汚し効果、炎や爆発のシミュレーション等を担当します。
- エディター - DPからの指示や所定の工程を映像作品に編集し、不必要な部分を除去します。

このマニュアルについて

このマニュアルは MediaWiki を利用し、世界中のボランティアによる共同作業で執筆されています [authors](#)。このマニュアルは日々更新され、これは英語版のマニュアルになります (NdT: 訳者によるメモ: "It is updated daily, and this is the English version")。英語以外の言語のマニュアルは一般に、世界中の閲覧者の利便性のため、英語版から翻訳されます。世界中で活躍する50名を超えるボランティアの開発者らによるたゆまぬ努力によって、マニュアルの内容と最新の情報との間に差が生じる事がしばしばあります。しかしながら、これは想像を超える Blender のパッケージにおいて、可能な限りのプロフェッショナルによる文章に触れる事ができる建設的な目標になります。

あなたを可能な限り効果的な方法で補助するため、このマニュアルは3Dアーティストが一般的に踏む、この特殊な領域における斬新であり複雑なソフトウェアを、あなたが好きなように活用できるように、適切に定めた手順にならって編集されています。もしもあなたがこのマニュアルを順番どりに読めば、アーティストが踏む手順を追うことで Blender について学習できかつフル・アニメーション作品の開発も行えます:

1. Blender について知る = はじめに、3Dインタラクション、シーン編集
2. モデル = モデリング、モディファイアと変形
3. ライティング
4. シューディング = マテリアル、テクスチャ、ペイント、ワールドと背景
5. アニメーション = 基本アニメーション、キャラクター、高度なアニメーション、エフェクトと物理シミュレーション
6. レンダリング = レンダリング、コンポジティング、シーケンスの編集
7. Blender をより良くするために = Blender の拡張

Warning: 2.5 update

This manual is the update for the future Blender 2.5 . The last stable version is [Blender 2.49b](#) and its documentation is available [here](#).

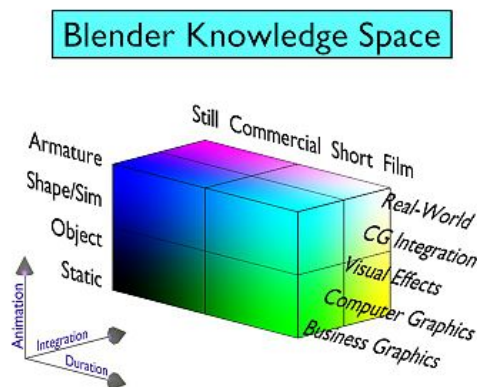
読者となる方々へ

このマニュアルは幅広い読者を対象として、「何かをやりたいのですが、Blender をどう活用すればいいですか?」つまり「メッシュを編集する最も新しい方法はありますか?」という問いに対して答えられるように書かれています。

このマニュアルは上記の目標達成のために時間を使っていただけの方々による世界規模の共同作業の成果物です。搭載された主要機能とそのドキュメントとの間にいくばくかの差が生じる場合もありますが、我々は可能な限り最新の情報を掲載し続けるように努力します。私達はあなたやエンドユーザーが知りたい事について、生きるこの意味について考えるかのように、本来の話題から大きく離れて思案することが無いよう注意深く検討します。

[その他の wiki books](#) により深く現在の Blender を広い視点から理解するためのチュートリアルや、リファレンス、ソフトウェアにスクリプトなどが存在します。もしこのマニュアルで疑問が解消されない場合は [他の Blender wiki books](#) もご参照ください。

CG と Blender を学ぶ



Blender について知る事とコンピュータ・グラフィックス (CG) について学ぶ事は2つの異なる話題になります。また他方で、コンピュータ・モデルが何であるかについて学ぶ事とそれを Blender でどのように開発するかについて学ぶ事も、2つの異なる学ぶべき物事です。一般の優れたライティング手法を学ぶ事と、Blender における種類の異なるランプについて学ぶ事は2つの異なる話題になります。初等的な、あるいは抽象的な理解は、大学でのアートやメディア論を学ぶ事で副次的に、あるいは図書館や書店で入手可能なアートやコンピュータ・グラフィックスについての本を読むことによって、あるいは試行錯誤を経て身に付ける事ができます。Blender 以外のパッケージ(例として Max や Maya)に関する書籍や記事であってもコンセプトが似ているために有用です。

コンセプトに対する理解をすると、あなたはより簡単に Blender (あるいは他のCGパッケージ) について理解を深めることができます。これらを同時に学習する事は困難であるため、あなたは2つの問題を別個の物として扱うのです。このように書く理由はあなた自身がこのジレンマと、どのようにしてこのマニュアルが1つの wiki book で両方の問題について言及するのか、について知っていただきたいからです。コンセプトの理解は普通、各章の冒頭の1つか2つの短い節で記述され、その章が扱うトピックとそれを習得するための全体の流れや工程について説明します。各章のそれ以降の部分では Blender の機能や特徴に特化した説明がなされます。このユーザー・マニュアルでは完全なコンセプトの理解を提供できません - これらは複数の書籍や雑誌、チュートリアルを参照あるいは経験し、長期間努力をすることで身に付けられる物です。あなたは Blender を長編の映像作品を作るために使用できますが、このマニュアルを読んで Blender を使用してもあなたを第二の Steven Spielberg にはできません!

非常に高い水準では、Blender を利用する事が、3次元空間に対する理解を深める方法を知る事と同様であるかの様に感じることができます:

1. インテグレーション - コンピュータ・グラフィックスをレンダリングし、現実世界の映像と統合あるいは組み合わせる事で用いる。(例として CGI や VFX)
2. アニメーション - 手作業やシミュレーションを利用して物の形を変えたり、移動させたりする。
3. 持続時間 - 製作する物が静止画、短編映像、1分間のコマーシャル映像、10分間のインディーズ映像作品、長編映像作品のいずれであるのか。

3D空間での移動、モデリング、ライティング、シェーディング、合成そしてその他多くのスキルが、所定のスペース内で生産性を維持するために必要とされます。スキルに熟達する事はあなたを生産的にするのです。Blenderの提供する応用性の高いツールについて熟達する事も同様です。例えば、ビデオ・シーケンス・エディター (video sequence editor, VSE) はアニメーションのスキルとはあまり関係がありませんが、持続時間とインテグレーションのスキルとして応用できます。インテグレーションの観点からは、VSEでEffect stripをアニメーションに適用するために利用される、IPO曲線と呼ばれるアニメーションの曲線について注目することが興味深い物となります。

このマニュアル内のコーナーやインターセクションには多くの人々の興味をひく、見る人によってはそれ自体が目的となりうる事柄がいつも存在しています。例えば、多くの才能あるアーティストによるCGの静止画像等です。Tony Mullen氏の著作*Introducing Character Animation With Blender*にはアーマチャとシェイプによって変形させられたCGモデルを利用した1分間のアニメーションを製作する手法についての記述があります。Blenderの流体シミュレーションをTV向けの作品やコマーシャルとして利用する場合はShape/Sim-Integrated-Minuteというインターセクションが利用できます。Elephants DreamやBig Buck BunnyはArmature-CG-Indieのスペースで非常に頻りに利用されます。結論として、あなたが何を必要としているのかによりますが、Blender内の多様なツール類やトピックスはあなたを多かれ少なかれ興奮させるでしょう。

そしてゲーム・デザインは四次元空間のようなものです、それはここまで述べてきた全ての知識でゲームを包含するように利用するためです。ゲームは単なる1分間程度の動画を含むだけの物ではなく、実際にゲームとして遊べる内容、ストーリーラインのプログラミングなど多くの要素から成ります。ゲームを製作することがいかに難しいかについての説明にもなり得ますが、あなたが実際にゲームを製作する前にはここまで述べられる全ての物事について理解しておく必要があります。そのため、このマニュアルではGame Engineの利用法について詳しくは記述されていません。そのためにもう一つwiki bookが必要になります。

Downloadable versions of this manual

(このマニュアルのダウンロード版)

公式ドキュメントはこの wiki で作成維持されており、常に最新の寄稿を含んでいますが、一方で wiki マニュアルの内容をダウンロード可能な形(たとえば PDF か他の形式で)にすることが誰かの役に立つかもしれません。

この需要に応じるため有志が wiki マニュアルの内容の **非公式版** を作成し始めました:

- wiki マニュアルの PDF 形式への非公式変換版
 - 次のページから利用できます: <https://archive.org/details/BlenderWikiPDFManual>
 - 詳細はダウンロードページをご覧ください。
 - 毎月更新予定
 - 最終更新: (訳注: この記事の English 版をチェックしてみてください)
 - PDF サイズ: ~44MB, ~1500 ページ (english wiki version)
 - 他の言語: JA, ET, RO, DE, ES, CZ

バイナリのインストール

Blender 2.59 のバイナリ実行ファイルまたはソースコードの双方とも財団のサイト(<http://www.blender.org/>)にあります。現在、2.59 をダウンロードするには、[homepage](#) の上の右側のナビゲーションメニューから "Blender 2.59" を選択してください。

ウィキでホストされているオンラインマニュアルでは、一般的には Blender Foundation のウェブサイト(最新のリリースバージョンからの機能のすべてが完全に更新されていないものもあります)に位置するの Blender の最新バージョンを使用することができます。このマニュアルの出版されたバージョンを使用する場合、ガイド CD-ROM に収録されているバージョンの Blender を使用することをお勧めします。次のテキストでは、"ダウンロード" が言及されるたびに、本を使っている人は、代わりに CD-ROM から Blender を取得する必要があります。

バイナリディストリビューションのダウンロードとインストール

バイナリ配布は、以下のオペレーティングシステムファミリー用に提供されています。

- [Windows](#)
- [Linux](#)
- [MacOSX](#)
- [FreeBSD, Irix, Solaris](#)

いくつかの非公式の配布が他のオペレーティングシステムのために存在するかもしれませんが、それらが [Blender Foundation](#) でサポートされていない場合、あなたはそれらのメンテナに直接問題を報告する必要があります。

Macintosh オペレーティングシステム用のバイナリは、2つの異なるハードウェアアーキテクチャ (Intel および AMD プロセッサ、および PowerPC 用の x86)、および静的にリンクまたは動的にロードされるライブラリとの間の選択のために用意されています。

インストーラは、コンピュータ上の2つの場所にファイルといくつかのフォルダを作成します。フォルダのセットを使えば Blender のプログラムのためであり、そしてもう一つは、ユーザーデータ用フォルダのセットです。あなたはこれらを作成するには、管理者権限を持っている必要があります。フォルダは次のとおりです。

- blender - コンフィギュレーション情報 (主に母国語でプロンプト)
- blendcache_B - 物理シミュレーション情報 (ソフトボディ、クロス、流体) のための一時スペース
- plugins - テクスチャとシーケンスのための追加機能
- scripts - Blender の機能を拡張する Python スクリプト
- tmp - 一時的な出力、中間のレンダリング

ハードウェアサポート

Blender は 2GB アドレス可能なメモリ制限を除去する、64bit の UNIX オペレーティングシステムを実行している 64bit のハードウェアプラットフォームをサポートしています。

Blender はまた、Intel Core-Duo プロセッサと AMD X2 チップなどのマルチコア CPU の性能のチップをサポートしています。スレッドの設定は、レンダリング時に並列で使用するコア数を示すために、レンダリングオプションのパフォーマンスセクションに記載されています。固定設定は、ユーザーが手動で表示する場合に使用されるコアの数を指定できると同時に、自動検出の設定は、システムで利用可能なすべてのコアを活用していきます。

Blender は特定の OS X、Windows XP、および Linux OSes で、すべての主要オペレーティングシステム上でペンベースのタブレットをサポートしています。

レンダリング時間を短くする方法に関する情報はで見つけることができます [Render](#) のセクション。

開発者プラットフォーム

これは、使用中のシステムのリストであり、アクティブな Blender の開発者によってサポート:

This is the list of systems in use and supported by active Blender developers:

Name	OS	CPU	Graphics card
Andrea Weikert	Windows XP 32	AMD Athlon 64 X2	Nvidia Quadro FX1500
Andrea Weikert	Windows XP 32	Intel P4	ATI Radeon 9000
Andrea Weikert	Linux 32	AMD Athlon 64 X2	Nvidia Quadro FX1500
Antony Riakiotakis	Ubuntu 14.04	Intel Core i5	Nvidia Geforce GT 540M
Antony Riakiotakis	Windows 7 64	Intel Core i5	Nvidia Geforce GT 540M
Bastien Montagne	Debian Testing 64	Intel Core i7 Q4700m	Nvidia Geforce GTX 850M
Benoit Bolsee	Windows XP 32	AMD Athlon XP	ATI Radeon 9200
Brecht van Lommel	Linux 64	Intel Core 2 Duo	Nvidia GeForce 460 GTX
Brecht van Lommel	OS X 10.6	Intel Core 2 Duo	Nvidia GeForce 9600M GT
Brecht van Lommel	Windows 7 64	Intel Core 2 Duo	Nvidia GeForce 460 GTX
Campbell Barton	Linux 64	AMD-FX 6-Core	Nvidia GeForce GTS 450
Daniel Genrich	Windows Vista 64	Intel Core 2 Duo	Nvidia GeForce 8500 GT
Diego Borghetti	Linux 64	Intel Core i5	Nvidia GeForce GTX 480

Diego Borghetti	Linux 64	Intel Core i7	Nvidia GeForce GTX 460M
Dustin Martin	OSX 10.5	Dual Quad Intel	Nvidia Geforce 8800 GT
Howard Trickey	Ubuntu 12.04 64	Intel Xeon E5-1650	NVidia Quadro 600
Howard Trickey	Windows 7 64	Intel Core i7	NVidia GeForce GTX 460
Howard Trickey	OSX 10.8.2	Intel Core Duo	NVidia GeForce 9400M
Jens Verwiebe	OSX 10.6/7/8/9	Intel Xeon 6-core@ 3.33	ATI 5870/7970
Jeroen Bakker	Latest Ubuntu 64bit	Dell m4300 Intel Core 2 Duo 2.0Ghz	Nvidia Quadro FX360M
Joshua Leung	Windows Vista 32	Intel Core2 Duo	Nvidia GeForce Go 7600
Julian Eisel	Linux 64	AMD Phenom II X4	Nvidia GeForce GTX 570
Julian Eisel	Linux 64	Intel Core i7	Nvidia GeForce GT 645M
Ken Hughes	Linux 32	Intel Core Duo	Nvidia GeForce GO 7500
Ken Hughes	Linux 64	AMD Athlon 64 X2	Nvidia GeForce 6600
Kent Mein	Linux 64	Intel Core Duo	Nvidia Quadro FX 1400
Kent Mein	SunOS 5.8	Sun Blade 150	ATI PGX
Matt Ebb	OSX 10.5	Dual Core Intel MBP	nVidia 8600M
Michael Fox	Linux 32	Celeron	Nvidia GeForce 6200
Nathan Letwory	Windows 7 Ultimate 64	AMD Turion X2 Mobile RM-74	ATI HD 4650
Nathan Letwory	Windows 7 Ultimate 64	AMD Athlon II X4 620	2x HIS ATI HD 5550 /w four monitors
Nicholas Bishop	Fedora 18 64bit	Intel Core i7 @ 2.93GHz	AMD Radeon HD 6950 (Gallium drivers, currently at OpenGL 2.1)
Nicholas Bishop	Ubuntu 12.10 64bit	Intel Core i5	ATI Mobility Radeon 5650 (Gallium drivers)
Raul Fernandez Hernandez	Linux 32	Pentium D 945	ATI X1550
Robin Allen	Linux 32	Intel Centrino duo	NVidia GeForce go 7600
Robin Allen	Windows XP 32	Intel Centrino duo	NVidia GeForce go 7600
Sergej Reich	Arch Linux 64bit	Intel Core2 Quad @ 2.83GHz	Nvidia GeForce GTX 285
Sergej Reich	Arch Linux 64bit	Intel Core i3 @ 2.10GHz	Intel Sandybridge Mobile
Sergey Sharybin	Debian Wheezy 64bit	Intel Core i7 920 2.6Ghz	Nvidia GeForce GTX 560Ti + GeForce GT 620
Sergey Sharybin	Debian Wheezy 64bit	Intel Core i5 2.4GHz	Intel Sandy Bridge + Nvidia GT 520M
Thomas Dinges	Windows 7 x64	Intel Core i5	Intel HD 2500
Thomas Dinges	Windows 7 x64	Intel Core i7	NVidia GeForce 540M + Intel HD 3000
Timothy Baldrige	SGL Irix 6.5 (mipspro)	8 x R16000	(headless)
Timothy Baldrige	SGL Irix 6.5 (mipspro)	2 x R10000	
Tamito Kajiyama	Windows Vista 64bit	Intel Core2 Duo	Nvidia Quadro FX 770M
Ton Roosendaal	OSX 10.7	iMac Intel Core i7	AMD Radeon HD 6970M
Ton Roosendaal	OSX 10.8	MacBook Pro i7 "Retina"	NVidia GT 650M + Intel HD 4000

ソースのコンパイル

現在、異なるサポートしているオペレーティングシステムのためにバイナリを作る4台のビルドシステムがあります。あなたのマシンのためにカスタムインストールバイナリをコンパイルすることに関する詳しい情報は [Building Blender](#)を参照してください。

プラグインのコンパイル

プラグインは、動的テキスト生成またはシーケンス(画像のmanipulation)のいずれかで機能を強化するルーチンをロードされます。詳細については、[このスレッド](#)を参照。

Python、スクリプト言語

[Python](#) は、汎用スクリプト言語で、[Blender download](#) ページから入手可能なバイナリのバージョン変更することなく、Blenderの機能を拡張するために使用できます。Blenderの現在のバイナリ版には適切なPythonライブラリが同梱されているので、多くの場合、Pythonスクリプトの実行にPythonのフルインストールを要求されることはありません。

ユーザ独自の[スクリプトを記述](#)や、[Blenderの独自のバージョンをコンパイル](#)したり、あまり一般的でない機能を利用する場合に、Pythonのフルインストールが必要になることがあります。完全なPythonの機能を希望するユーザーは、インストール手順について [Python](#) のウェブサイトを参照する必要があります。

Linuxへのインストール

ダウンロード

Linux用のBlenderの最新安定版は、[Blender download page](#) から入手できます。あるいは、お使いのディストリビューションのソフトウェアリポジトリがBlenderのパッケージを提供していれば、そちらからも。

バージョン

Linux用のブレンダーは現在、32ビットおよび64ビットバージョンで利用可能です。32ビットバージョンのLinuxのユーザーは、32ビットバージョンのBlenderをダウンロードする必要があります。Linuxの64ビットバージョンのユーザーは、Blenderの32ビットまたは64ビットバージョンのいずれを使用することも選択できます。とはいえ、Blenderの64ビットバージョンを使用すれば、パフォーマンスの向上を体感できるでしょう。特に大量のRAMを持つシステムでは尚いっそう。

32ビットまたは64ビットバージョンのどちらのLinuxを使用しているかを判断するには、ディストリビューションのマニュアルを参照するか、`uname` コマンドに `-m` オプションをつけて使用しましょう。`uname` はシステム情報を表示するもので、`-m` オプションをつけるとマシンのハードウェア名を表示します。

- 端末コンソールを開きます。
- `uname -m` コマンドを入力します。

あなたが32ビットシステムを使用している場合、`uname -m` は `i686` の値を返します。64ビットシステムなら `x86_64` の値を返します。

ディストリビューションのリリース

Ubuntu, Debian, Open SUSE, Fedora など多数の主要なディストリビューションのほとんどは、それらのソフトウェアリポジトリでBlenderのビルドを提供しており、そのディストリビューションのパッケージマネージャを介してアクセスすることができます。お使いのディストリビューションでこれが行なわれていない、またはリポジトリが更新されておらず最新のBlenderのリリースが含まれていない場合は、以下の手順でそれを自分でインストールすることができます。

ディストリビューションのリリースは、概して非常に古い、旧式なバージョンであることに気をつけましょう。

インストール

まず、お使いのディストリビューションが最新のBlenderのバージョンを提供しているかどうかを、パッケージマネージャを介して確認しておきましょう。それがない場合は、[Blender download page](#) から該当バージョンのLinux用のBlenderをダウンロードし、アーカイブを任意の場所に解凍します。

こうすると `blender-VERSION-linux-glibcVERSION-ARCH` という名前のディレクトリが作成されます。`VERSION` のところはBlenderのリリースバージョンで、`glibcVERSION` は必要になるglibcのバージョンであり、`ARCH` はあなたのコンピュータのアーキテクチャ (`i686` or `x86_64`) です。このディレクトリで、`blender` のバイナリが見つかるはずですよ。

Blenderを実行するには、


- [X.Org server](#) を起動 (まだ実行されていない場合)
- ファイルマネージャを使用して、Blenderのディレクトリに移動し、Blenderの実行ファイルをダブルクリック、もしくは
- 端末コンソールを開き、Blenderのディレクトリに移動し、`./blender` コマンドを実行する

/opt または /usr/localへのインストール

また、Blenderのディレクトリを移動させることによって、`/opt` または `/usr/local` のいずれかの場所にBlenderをインストールすることもできます。任意のディレクトリからBlenderを実行できるようにしたい場合は、さらにPATH変数を更新する必要があります。PATHを設定するための推奨方法は、ご使用のオペレーティング・システムのマニュアルを参照してください。

Configuration

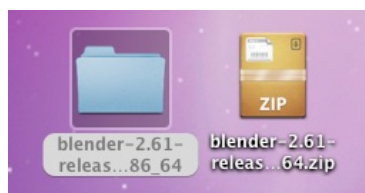
Alt+Mouse Conflict

多くのLinuxディストリビューションでは、Alt LMB  をウィンドウ移動に使うのがデフォルトになっています。そのため、Blenderでは Alt+Click の機能が無効になりやすいため、Superキー (Windows Key) に変更しましょう

- Ubuntu 11.04: Settings > Window Manger Tweak > Accessibility > Change Window Key to Super
- todo - others

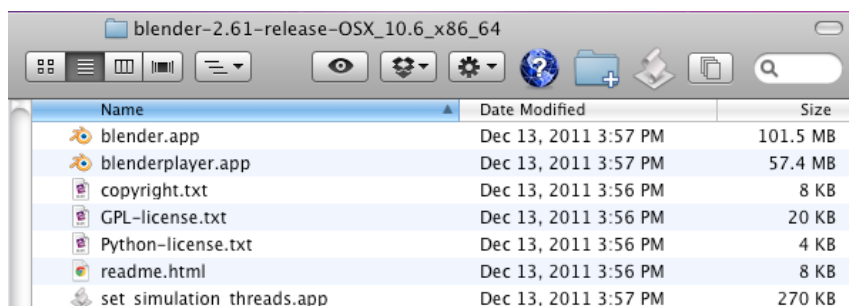
Macへのインストール

MacへのBlenderのインストールは非常に簡単です。PowerPCプロセッサ、およびIntelプロセッサを使用した32ビットおよび64ビットのMacのためのプリコンパイルされたバージョンが用意されています。ご使用のプラットフォームとプロセッサが分かったら：



Blender Unzipped (left) and Zipped (right)

- あなたのシステムに適したBlenderを [blender.org download page](http://blender.org/download/page)からダウンロードしましょう
- ダウンロードしたファイルでクリック/右クリック RMBして、フォルダに解凍を選択します。
- ダウンロードしたBlenderがある場所に、ダウンロードしたファイルと同じ名前の、拡張子なしのフォルダが作成されます。
- フォルダをクリックすると、Finderで開くことができます (下の、Opening Blender From Finder- の画像を参照してください)
- *blender.app* ファイルで2回クリック LMB しましょう。準備完了です！



Opening Blender from Finder - Example using Blender 2.61 for 64 bits Intel Mac, Mac OS X 10.6

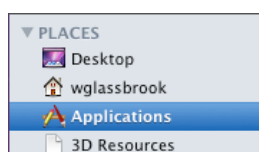


Blender 2.6X以上のサポート対象プラットフォーム

Blender 2.6Xバージョンとそれ以上は、PowerPC G5プロセッサをそなえたコンピュータ、またはIntelプロセッサを使用したMacでないと実行できず、Mac OS X 10.4以降の搭載が必要になります。

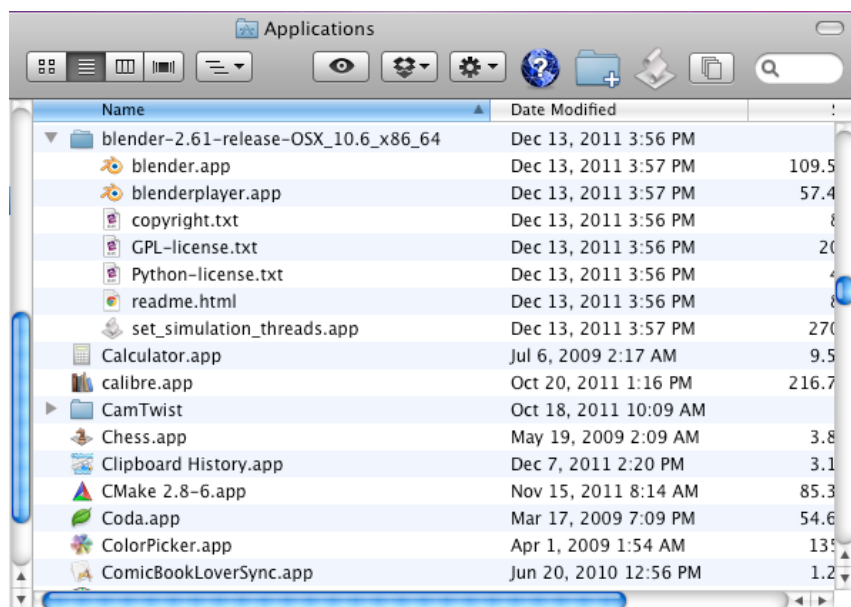
重要: コンソールウィンドウを表示するためにコマンドラインからBlenderを実行する必要がある場合は、[Blenderのコンソール・ウィンドウ](#) についてのページをご覧ください

アプリケーションへBlenderを追加



Places and Applications on Mac


Mac OS XのアプリケーションフォルダにBlenderを移動することもでき、Macの他のアプリケーションと同じように動作します。これは、解凍したフォルダをMacのアプリケーションフォルダにコピーするか移動することで行えます。Finderのファイルブラウザを使用したりPlacesを探して、アプリケーションフォルダを見つけましょう。左側にある、Places タブを開くと、アプリケーションフォルダが見つかるはずですが、解凍されたBlenderのフォルダをアプリケーションフォルダに移動しましょう。



Blender, when moved to Applications folder

DockにBlenderを追加

DockへのBlenderの追加は、次の操作で実行できます:

- Blenderが *アプリケーションフォルダ*に配置されている状態で、*blender.app* をクリックして、dockにドラッグします。
- Blenderが実行されているときに、MacのDock内にあるアイコンを右クリック RMB  して、*Dockに追加*を選択します。

Windowsへのインストール

ダウンロード

次のサイトからWindows用の Blender の最新安定版を入手できます。[Blender ダウンロード ページ](#)

バージョン

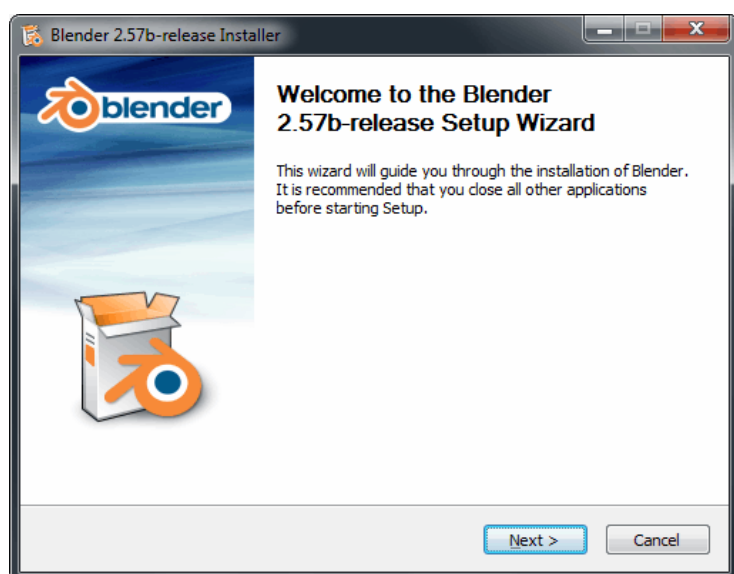
Windows用の Blender は、32bit および 64bit バージョンが利用可能です。Windows の 32bit バージョンを持つユーザーは、Blender の 32bit版をダウンロードする必要があります。Windows の64bit バージョンを持つユーザーは、Blenderの32bitまたは64bitバージョンを使用することもできますが、大きな RAM が搭載されたシステム上で、Blenderの64ビット・バージョンを使うとパフォーマンスの向上に気づくでしょう。

あなたがWindowsの 32bit または 64bit バージョンをどちらを持っているか判断するには、C:パーティションで確認することができます。Program Files (x86) フォルダがある場合は、Windowsの64ビットバージョンを持っている。32ビットプログラムが格納されている場所はProgram Files (x86) フォルダにあります。(

インストール

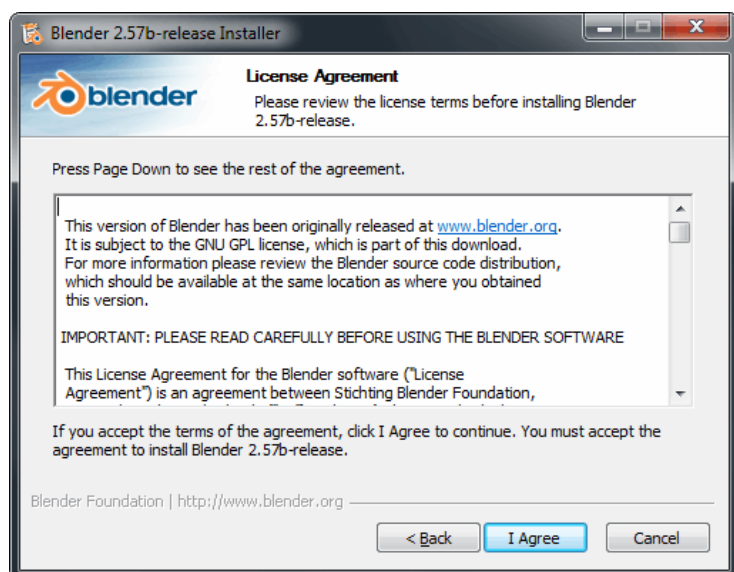
ダウンロードが完了したら、ダウンロードフォルダに移動し、Blender の実行ファイルをダブルクリックし、インストールプロセスを開始しましょう。Blender をインストールするために管理者権限を必要とすることに注意してください。

ようこそ画面



ようこそは、インストールプロセスの最初の画面です。[Next]をクリックして続けます。

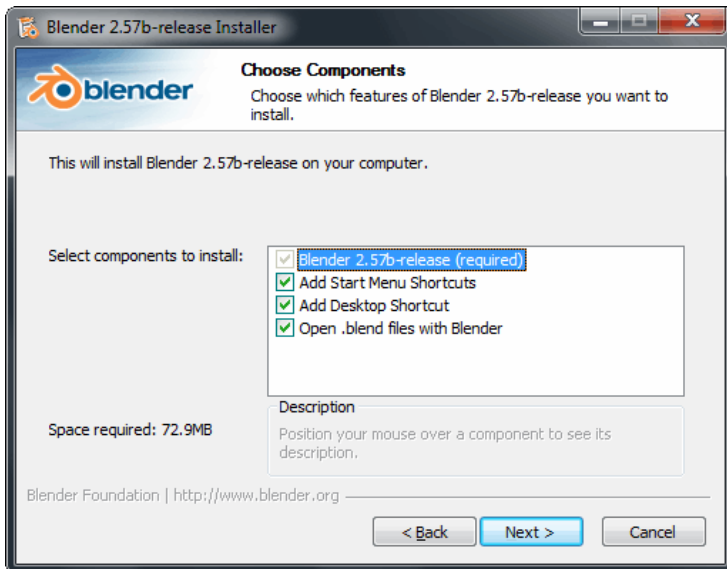
ライセンス契約



インストールを続行する前にライセンス契約を受け入れる必要があります。

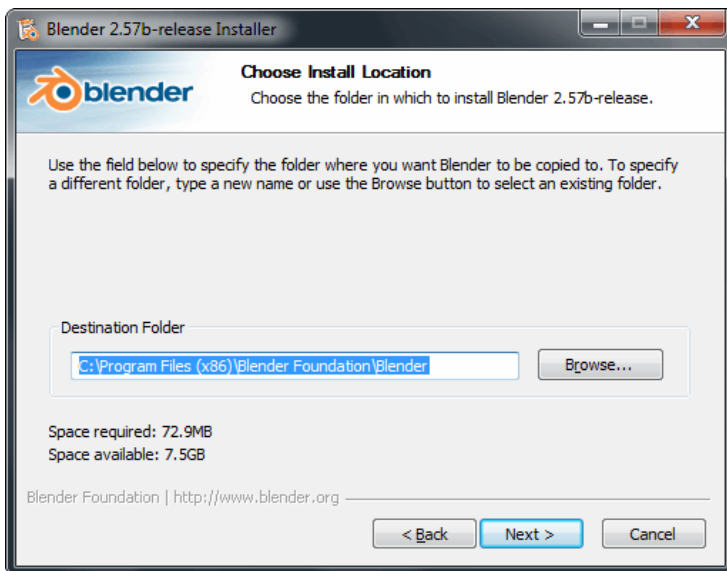
インストールオプション

プログラムのオプション



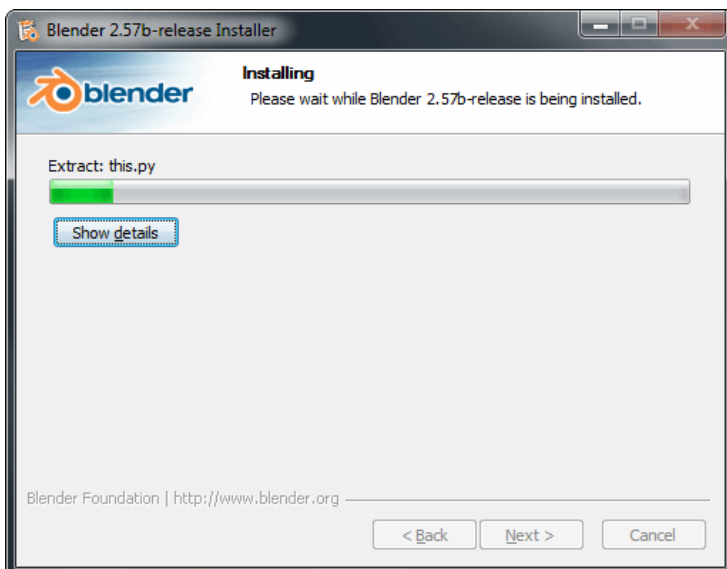
任意のオプションを選択します。

インストール場所

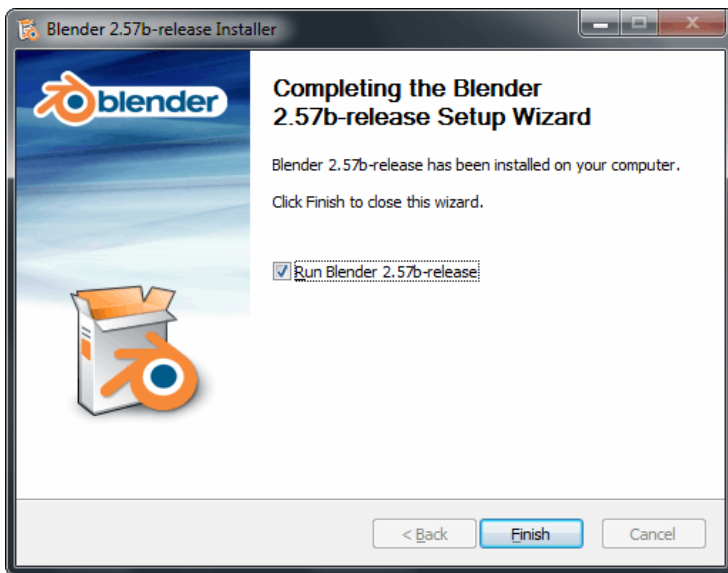


ここではBlenderがインストールされるディレクトリを選択することができます。

インストール中



インストール完了



インストール完了！ ブレンダー2.5の新規インストール。インストールを閉じた後、Blenderを実行するかしないか設定した後は、[Finish]をクリックします。

ポータブルのインストール

BlenderをUSBキーに持ちだして、どこへ行っても使用できるようにしたい場合は、zipバージョンをダウンロードし、USBキーに展開しましょう。あなたはアニメーション出力を保存することを避けたいかもしれませんが、あるいは、ドライブとして頻繁に記述される一時ファイルはUSBキーの寿命を短くするかもしれません。

USBキーにすべての構成ファイルとインストールされているアドオンを保つために、解凍したBlenderのフォルダにconfigという名前のフォルダを作成します。今すぐすべての設定ファイルの読み込みや、USBキーではなく、あなたがBlenderを実行しているコンピュータに書き込まれます。

サポートされている他のオペレーティングシステム

FreeBSD

blender-2.##-FreeBSD-####.tbz ファイルを [Blender download page](#) からダウンロードしましょう。2.## の部分はBlenderのバージョンであり、#### の部分はマシンのアーキテクチャ (i386 または amd64)です。

Blenderを起動するには

- アーカイブを解凍し
- シェルを開き、解凍したアーカイブのディレクトリに移動し
- [X.Org server](#)が動作している間に、./blender コマンドを実行しましょう。

他のサポートされていないオペレーティングシステム

MorphOS

このプラットフォームへの移植は、Guillaume Roguezによって継続されています。この移植の詳細情報は、彼のウィキ [his wiki](#)で見つかるはずです (フランス語版[french version](#))。

この移植版のインストールでは、[RAM](#) よりむしろ、ハードドライブ上にアーカイブを解凍することを推奨します。PythonがRAMから、その dynamic .pymモジュールを使用することができないからです。

インストール後の設定

Blenderは設定変更無しでも普通動きますが、あなたは自分のハードウェアに依存した変更がしたくなるでしょう。

このセクションではシステム固有のユーザー設定だけをカバーします。



Info

キー組み合わせ CtrlU は現在開いているBlenderファイルの全ての設定を標準Blenderファイルに保存します。標準Blenderファイルにある設定は、Blenderを最初に起動したときや、CtrlN を押して新しいファイルを始めた時に読み込まれます。もし間違って設定を標準設定ファイルに保存してしまった場合(例えば作業中のものを標準として保存してしまった場合)は、ファイルメニューから初期設定(Factory Settings)を復帰させることができ、CtrlUで新しく読み込んだ初期設定を保存できます。

以下のセクションタイトルはユーザー設定のカテゴリと一緒にです。

入力 (Input)

- テンキーを持っていないのであれば、'Numパッドを再現' ('Emulate Numpad')を有効にしたいくなるでしょう。
- マウスの中ボタンを持っていないのであれば、'3ボタン マウスを再現' ('Emulate 3 Button Mouse')を有効にすることができます。

ファイルパス (File Paths)

これは必須ではありませんが、あなたはより使いやすい標準ロケーションをパスにセットしたいくなるでしょう。

- テンポラリディレクトリ: 一時ファイル記憶のための高速なディスクを持っている場合、これを変えたいくなるでしょう。
- 画像エディター: 画像を(画像スペースから)外部で編集できるようにするために有益です。例えばGIMPのパスをここに設定できます。



Info

Blenderではパスの最初にある"/"が現在開いてるblendファイルのディレクトリを意味し、相対パスを参照するのに使われています。

システム (System)

特別な場合に設定したくなるであろうたくさん設定がここにあるので、このドキュメントでは一般的な特徴にフォーカスしています。

- VBOs (Vertex Buffer Objects): ビューポートのパフォーマンスに良いスピード加速をもたらす可能性があります。これが問題を引き起こすまでは有効に保ってください(いくつかの古いハードウェアはVBOsをサポートしません)

シーケンサー / クリップエディタ:

- メモリキャッシュの上限 (Memory Cache Limit): ムービークリップのシーケンサーを使う場合、あなたはキャッシュリミットを上げたいくなるでしょう。これはscrubbingをかなり速くします。しかしリミットを全システムメモリ以下に留めることに気を付けてください、

設定およびデータへのパス

Blenderはシステム全体にインストールされるか、同梱されている必須ファイルを抽出することで動作します。

Blenderが使用するディレクトリは3つに分かれます。正確な場所は、システムによります。

- **LOCAL:** (内蔵されている)設定・ランタイムデータの場所。
- **USER:** (通常ユーザーのホームディレクトリにある)設定ファイルの場所。
- **SYSTEM:** システム全体のインストールに関わるランタイムの場所(おそらく読み取り専用になっている)。

システムにインストールするためには、**SYSTEM** および **USER** ディレクトリ双方が必要となります。

配布されているBlenderをローカルに引き出すために、ユーザー設定およびデータのランタイムデータが同じサブディレクトリに置かれます。これにより、複数のバージョンのBlenderが競合することなく動作することができ、**USER** および **SYSTEM** ファイルを無視します。

各システムにおけるデフォルトの設置場所をご覧のとおり:

OSX

LOCAL: `./2.57/`

USER: `/Users/{user}/Library/Application Support/Blender/2.57/`

SYSTEM: `/Library/Application Support/Blender/2.57/`

注意: OSXはBlenderのバイナリを `./blender.app/Contents/MacOS/blender` に保存します。そのため、データおよび設定へのローカルパスは `./blender.app/Contents/MacOS/2.57` となります。

Windows

LOCAL: `.\2.57\`

USER: `C:\Documents and Settings\{username}\Application Data\Blender Foundation\Blender\2.57\`

SYSTEM: `C:\Documents and Settings\All Users\Application Data\Blender Foundation\Blender\2.57\`

Unix (Linux/BSD/Solaris)

LOCAL: `./2.57/`

USER: `$HOME/.blender/2.57/`

SYSTEM: `/usr/share/blender/2.57/`

注意: `./2.57/` はBlenderの実行ファイルで、`blender.org`公式ビルドに内蔵されているファイルと関係があります。

パス一覧

This is the path layout which is used within the directories described above.

Where `./config/startup.blend` could be `~/blender/2.57/config/startup.blend` for example.

- `./autosave/ ...`
autosave blend file location. *Windows only, temp directory used for other systems.*
search order: **LOCAL, USER**
- `./config/ ...`
defaults & session info
search order: **LOCAL, USER**
- `./config/startup.blend`
default file to load on startup.
- `./config/bookmarks.txt`
file selector bookmarks.
- `./config/recent-files.txt`
recent file menu list.
- `./datafiles/ ...`
runtime files
search order: **LOCAL, USER, SYSTEM**
- `./datafiles/locale/{language}/`
language translations. *not currently in use!*
- `./datafiles/icons/*.png`
icon themes for blenders user interface. *not currently selectable in the theme preferences.*
- `./datafiles/brushicons/*.png`
images for each brush.
- `./scripts/ ...`
python scripts for the user interface and tools
search order: **LOCAL, USER, SYSTEM**
- `./scripts/addons/*.py`
python addons which may be enabled in the user preferences, includes import/export format support, render engine integration and many handy utilities.

- `./scripts/addons/modules/*.py`
modules for addons to use (added to python's sys.path)
- `./scripts/addons_contrib/*.py`
another addons directory which is used for community maintained addons (must be manually created).
- `./scripts/addons_contrib/modules/*.py`
modules for `addons_contrib` to use (added to python's sys.path)
- `./scripts/modules/*.py`
python modules containing our core API and utility functions for other scripts to import (added to python's sys.path)
- `./scripts/startup/*.py`
scripts which are automatically imported on startup.
- `./scripts/presets/{preset}/*.py`
presets used for storing user defined settings for cloth, render formats etc.
- `./scripts/templates/*.py`
example scripts which can be accessed from: Text Space's Header -> Text -> Script Templates
- `./python/ ...`
bundled python distribution only necessary when the system's python is absent or incompatible
search order: **LOCAL, SYSTEM**

注意事項

ユーザースクリプトパス

The user preferences script path provides a way to set your own directory which is used for scripts as well as the user scripts path. Be sure to create subfolders within this directory which match the structure of blender's scripts directory, `startup/`, `addons/`, `modules/` etc. because copying scripts directly into this folder will not load them on startup or as addons.

環境変数

Environment variables can be used to override default path locations, eg: `$BLENDER_USER_CONFIG`, `$BLENDER_SYSTEM_PYTHON`.

This is not normally something which needs setting but can be useful for custom configurations.

For details see the 'Environment Variables' section in 'blender --help'

スクリプトパス & Missing Buttons

If blender starts with no interface this is probably because the scripts are not loading correctly and can be caused by...

- script path not found.
- an error in one of the scripts.
- a version mis-match between blender and the scripts.

It's best to load blender from a terminal to see any error messages to see what's wrong.

はじめの一歩

Blenderのインストールができました！ さあはじめましょう。もしBlender 2.4xや他の3Dソフトウェア(Maya, 3ds Max, XSIなど)を良く知っているなら、Blender 2.5はちょっと違うと思うことでしょう。でも、以前に使っていたソフトウェアと似ているところ(3Dビューポート、アウトライナー、タイムライン...)にも気付くはず。もし3Dソフトウェアを使うのが初めてならば、これがまさに全く新しい体験になるでしょう。ただし、Blenderで3Dを学ぶ時のルールが一つあります——怖がらないで。噛み付かないから！

Blenderを開いたら、スプラッシュスクリーンを見てみましょう。

一番上にはBlenderのバージョンが書かれています。



左側には、[リリースログ](#)(このバージョンでの変更点)や、wikiマニュアル(あなたが今読んでいるページです)、[Blender公式ページ](#)などへのリンクが張られています。(これらのリンクは、[Help](#)メニューからたどることもできます。)

右側には、最近保存した6つのblenderファイル(.blend)が並んでいます。Blenderを初めて起動したときは、まだ編集したり保存したファイルはないのでこの部分はからっぽです。このリストは、File → Open Recent からアクセスできます。また、Interactionメニューから、キー

マップ設定を選択することができます(デフォルトでは、Blenderと Maya があります)。

Blenderを使い始めるには、以下の2つの方法があります。

- (もし可能なら)最近保存した6つのファイルのどれかを選ぶ。
- スクリーンのどこか(スプラッシュスクリーンの暗い部分以外)をクリックするか、Esc キーを押して新しいプロジェクトを開始する。

インタフェースのコンセプト



まずはじめに、Blenderはマルチプラットフォームのソフトウェアとなっています。つまり、Linux や Mac OS X、Windowsなどで動作させることができます。なので、*Blenderはこれらの異なるシステム上でも同じものなのか？*と疑問に思うかもしれません。答えは、そのとおりです。なぜなら、Blenderのインタフェースは [OpenGL](#) を用いて独自のデザインで設計されているからです。

三原則

Blenderユーザインタフェースは以下の三原則に基づいて設計されています。

- **重ねない**: ウィンドウを押し出したりドラッグしたりしなくとも、全ての関連するオプションやツールを一目で見渡せるようにすること⁽¹⁾。
- **ふさがない**: ツールやオプションがBlenderの他の部分をふさがないこと。Blenderは、何かを実行する前にデータ入力を要求するポップアップウィンドウを出すことはありません。
- **じゃまをしない**: 作業中にマウスやキーボードをあれこれ持ち替えさせるようなことはせず、ユーザの入力を可能な限り一貫させ、次に起こることが予測できるようにすること。

⁽¹⁾ただし、マルチスクリーン用にマルチウィンドウにすることは認めています。これは、**重ねられない原則**の例外です。

強力なインタフェース



[OpenGL](#)によって描画されているので、それぞれの作業に特化した、思うがままの画面構成にすることができます。この画面構成は名前を付けて保存することができます。

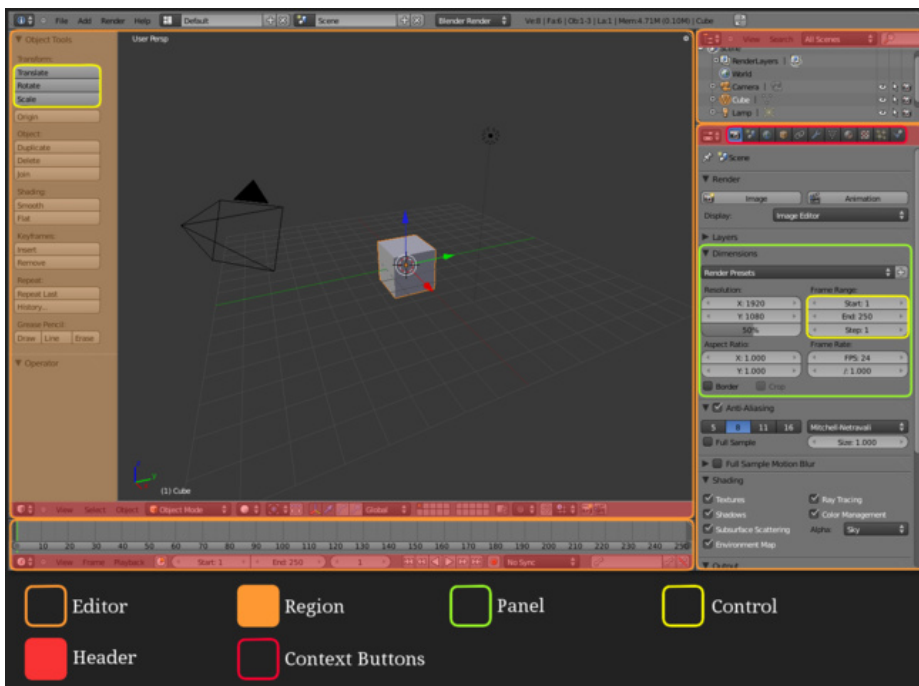
Blenderでは効率よく作業するためにキーボードショートカットを多用しますが、このキーマップも簡単に編集して覚えやすいホットキーを設定することができます。

概観

では、デフォルトのインタフェースを見てみましょう。エディタ (*Editor*)、ヘッダ (*Header*)、コンテキストボタン (*Context Buttons*)、パネル (*Panel*)、コントロール (*Control*) があるのがわかります。

- Blenderでは、特定の機能(3Dビュー、プロパティエディタ、ビデオシーケンスエディタ、ノードエディタなど)に反応する部分をエディタと呼びます。エディタの一番上か一番下にはヘッダがあります。
- コンテキストボタンはタブのようなもので、大抵ヘッダ上にあります(例えば、プロパティエディタみたいなものです)。これを用いてオプションを操作します。
- 各エディタのオプションは、論理的に整理されたパネルにまとめられています(Shadow/パネル、Color/パネル、Dimensions/パネルなど)。
- エディタにはサイドバーが付いているものもあります。その場合、パネルとコントロールはそこにまとめられています。サイドバーは、ワークスペースを最適化するために一時的に隠すことができます。
- コントロールはパネル中にあり、これを用いて機能やオプション、数値を変更します。Blenderには以下のようなコントロールがあります。
 - **ボタン**: ツール(並進、回転、キーフレームの挿入)にアクセスします。これらのツールには、大抵、作業のスピードアップのためのキーボードショートカットが設定されていますが、ボタン上にマウスカーソルを乗せることによってツールチップ上に表示させることができます。
 - **チェックボックス**: オプションの有効/無効を設定します。このコントロールはブール値(True/False, 1/0)のみを持つことができます。
 - **スライダ**: 数値を入力します。数値は有限(0.0 → 100.0)や無限(-∞ → +∞)といった値を持つことができます。スライダには2つの種類があることに注意してください。
 - **選択メニュー**: リストから値を選択します。チェックボックスとの違いは、値には名前がついていて、2つ以上の値があることです。

[ボタンとコントロールについてもっと詳しく読む](#) »



入力機器

Blenderのインターフェースは以下の推奨入力機器を用いることを想定してデザインされています。






- ホイール付き3ボタンマウス
- テンキー (NumPad) 付きフルキーボード

もし、推奨入力機器を持っていない場合 (例えばノートパソコンを使っている場合) は、Blenderのユーザ設定を変更してエミュレートすることができます。

[Blenderの設定について詳しく読む](#) »

本マニュアルにおける慣例

このマニュアルでは、以下のような慣例を用いてユーザの入力を表現します。

- マウスボタンは、LMB  (左マウスボタン)、MMB  (中マウスボタン)、RMB  (右マウスボタン) と示します。
- マウスにホイールが付いている場合は、ホイールのクリック操作によって MMB  となり、ホイールを回転させると Wheel  になります。
- ホットキーはキーに実際に描かれているような形で表示します。例えば、Gは小文字のgを表しています。⇧ Shift、Ctrl、Altといったモディファイアキーを用いる場合はCtrlWや⇧ ShiftAltAのように示します。
- 0 NumPadから9 NumPad、+ NumPadなどはテンキーを表します。NumLockを常にオンにしておきましょう。
- その他のキーは、Esc、⇐ Tab、F1からF12のように、その名前前で示されます。
- また、矢印キーは←や→のようになります。

一般的な使用方法

Blenderではマウスとキーボードの両方を多用するので、Blenderユーザ達は次のような黄金則を導きだしました。片手をマウスに乗せ、もう片手をキーボードに乗せたままにすること。もし、英語キーボードと全く異なる配列のキーボードを常用しているのなら、Blenderでの作業のために英語あるいは米語キーボードに変更するとよいでしょう。頻繁に使われるキーは、英語キーボード配列の左手のスタンダードポジション (人差し指がFの上) から届きやすい位置にまとめられています。これは、マウスを右手で使うことが想定されているためです。










Blenderのデフォルトキーマップとデフォルトホットキーを変更することはできますが、このマニュアルではデフォルトキーマップに基づいて説明していきます。



[Blenderの設定について詳しく読む](#) »

マウスボタンエミュレーション

もし、3ボタンマウスを持っていないなら、[User Preferences](#)のオプションをチェックして、3ボタンマウスをエミュレートするようにしましょう (デフォルトでチェックされています)。

以下の表は、キーとのコンビネーション (コンボ) を表しています。

2-button Mouse	Apple Mouse
LMB  LMB 	LMB  (マウスボタン)
MMB  Alt LMB 	⇧ Opt LMB  (Option/Altキー + マウスボタン)
RMB  RMB 	⌘ Cmd LMB  (Command/Appleキー + マウスボタン)

このマニュアルに書かれている全てのマウス/キーボードコンビネーションは、この表に書かれているコンボで表せます。例えば、シングルボタンマウスの環境では、⇧ ShiftAlt RMB  は⇧ ShiftAlt⌘ Cmd LMB  となります。

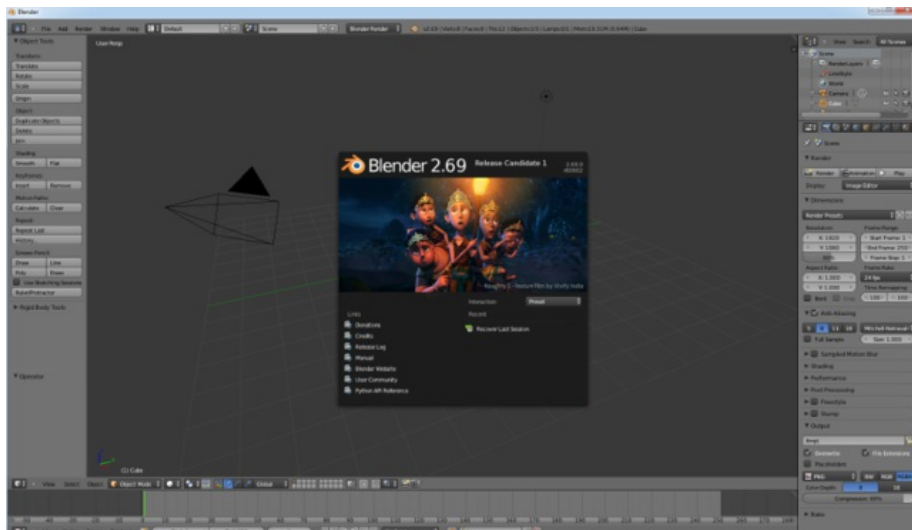
テンキーエミュレーション

[ユーザ設定のテンキーエミュレーションについて詳しく読む](#) »

The Window System

(ウィンドウシステム)

Blenderを起動すると、このような画面が現れます(中央のsplashスクリーンはバージョンによって異なっているかもしれません)。



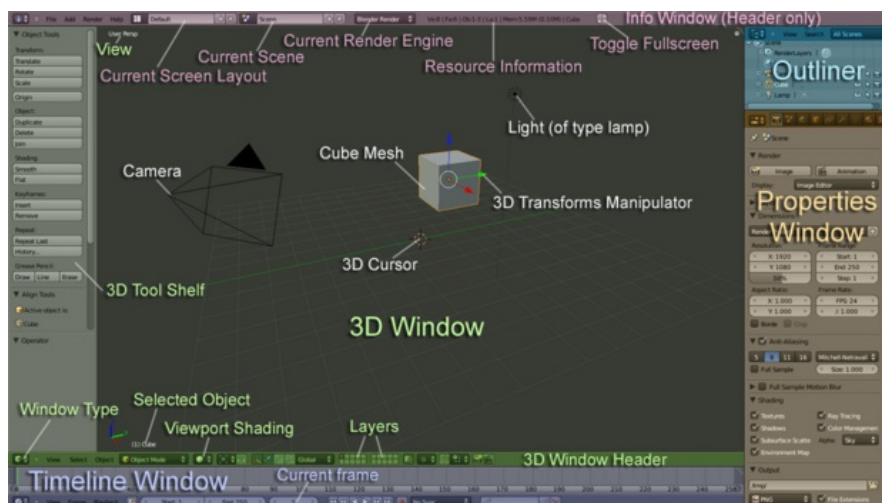
ウィンドウの中央にあるのがsplashスクリーンです。ここから最近使ったBlenderファイルに簡単にアクセスすることができます。新しいファイルで作業を始めるならsplashスクリーンの外側の画面をクリックしましょう。splashスクリーンが消えて、デフォルトレイアウトと立方体が姿を現します。

全てのウィンドウはいくつかの部分に分割することができます([このページ](#)を参照)。以下では、デフォルトシーンについて見て行きます。

The default scene

(デフォルトシーン)

デフォルトシーンはBlenderや新規ファイルを開くたびにロードされます。



The default blender scene

デフォルトシーンは5つのウィンドウに分かれています。

- 情報ウィンドウ(メインメニュー、図の赤色部分): 最上部にありヘッダのみが表示されています。
- 大きな3Dウィンドウ(3Dビュー、図の緑色部分)
- タイムラインウィンドウ(図の紫色部分): 最下部に表示されています。
- アウトライナーウィンドウ(図の黄色部分): 右上に表示されています。
- プロパティウィンドウ(ボタンウィンドウ、図の青色部分): 右下に表示されています。

このページではいくつかの基本要素のみを紹介していきます。

The Info Window (main menu)

(情報ウィンドウ(メインメニュー))

上の画像を参考にしながら読んでください。

- **Current Screen** (デフォルトでは**Default**): Blenderには、予め設定されたスクリーンが用意されています。独自のスクリーンが必要ならば、新規に作って名前を付けることができます。
- **Current Scene**: 複数のシーンを用いることによって、制作する作品を整理されたパターンに分けることができます。他の3Dソフトによってはシーンとファイルが同じになっているものもありますが、Blenderでは一つのファイルに複数のシーンを含めることができます。

- **Current Rendering Engine**: 使用可能なレンダリングエンジンのリストです。
- **Resource Information**: Blenderとシステムリソースの情報を表示します。選択されたシーンに含まれる頂点、面、オブジェクトの数を基にメモリの使用量を表示します。コンピュータの限界に近づいているかどうかをチェックできる便利な機能です。

3D Window View

(3Dウィンドウビュー)

上の画像を参考にしながら読んでください。



- **3D Transform Manipulator**: オブジェクトの3D変換を視覚化したものです。オブジェクトの3D変換(並進: grab、回転: rotation、拡大縮小: scale)はキーボードショートカット(G/R/S)を用いて行うこともできます。



- **3D Cursor**: 3Dカーソルは複数の機能を持っています。例えば、新しいオブジェクトを作成したときにそれが配置される場所や、回転中心位置などを表します。下にあるのが3Dカーソルの画像です。
- **Cube Mesh**: デフォルトでは、Blenderはいつも立方体メッシュをグローバル3D空間の中央に配置した状態で(上の絵ではちょっと動かしてありますが)始まります。しばらく使っていると、おそらくデフォルト設定を変更したくなるでしょう。その場合は、好みの状態に編集してから、その状態をスタートアップ時のデフォルトとして保存(キーボードショートカットCtrlU)してください。



- **Light (of type Lamp)**: デフォルトでは、Blenderはいつもグローバル3D空間の中央付近に光源を配置した状態で始まります。



- **Camera**: デフォルトでは、Blenderはいつもグローバル3D空間の中央付近にカメラを配置した状態で始まります。
- **Selected object**: この場所には選択中のオブジェクト名が表示されます。

3D Window Header

(3Dウィンドウヘッダ)


これは3Dウィンドウのヘッダです。Blenderの全てのウィンドウにはヘッダがあります(この例では実際にはウィンドウの下部にあるフッタですが、Blenderではとにかくヘッダと呼びます)。ヘッダについての詳細は、[このページ](#)を見てください。

上の画像を参考にしながら読んでください。



- **Window Type / Editor Type Selector**: ここをクリックすることによって表示されるリストから、ウィンドウの種類を変更することができます(これは全てのウィンドウヘッダにあります)。ウィンドウの種類については、[Window Typeリスト](#)を見てください。



- **3D Transform manipulator options**: ツールバー上の座標系アイコン(2.5以前では手アイコンでした)をクリックするか、CtrlSpaceで [マニピュレータ](#) の表示/非表示を切り替えられます。並進/回転/拡大縮小マニピュレータは、座標系アイコンの右隣の3つのアイコンのどれかをクリックすると表示されます。このアイコンを Shift LMB  すると、それぞれのマニピュレータの表示を加えたり消したりできます。



- **Viewport shading**: Blenderでは、[OpenGL](#)を用いて3Dウィンドウを描画します。このボタンをクリックすることによって表示される陰影スタイルから、リアルタイム陰影生成の種類(Draw Typeと呼ばれるBlenderの陰影リスト)を選ぶことができます。単純なbox表示から複雑なテクスチャ表示まで好みの表示を選べますが、テクスチャ表示を用いるならば、高性能なグラフィックスカードを使用することをお勧めします。



- **Layers**: レイヤーを使うとモデリングやアニメーションがより簡単になります。Blenderのレイヤーを使えば、オブジェクトを特定の目的毎のグループに分散させることができます。例えば、あるレイヤーに水のオブジェクトを、別のレイヤーに木々を配置したり、カメラとライトをまた別のレイヤーに配置したりできます。レイヤー表示をオン/オフしてビューを整理することもできます。

Buttons (Properties) Window Header

(ボタン(プロパティ)ウィンドウヘッダ)

Blenderの全てのウィンドウにはヘッダがあります。ヘッダに関する詳細は[このページ](#)を見てください。プロパティウィンドウのヘッダは上の画像では少し暗く表示されています。

上の画像を参考にしながら読んでください。



Properties Window Header

ボタン(プロパティ)ウィンドウにはグループ化されたパネルがあります。このウィンドウのヘッダには、ボタンのリスト(コンテキストボタンと呼ばれます)があり、表示するパネルのグループを選ぶことができます。

ボタン(プロパティ)ウィンドウのパネルは、関連するボタンやコントロールをグループ化したり整理したりするのに役立ちます。パネルは、パネルタイトルの左(例えば、*Render*の隣)にある小さな矢印をクリックして折り畳んだり、ドラッグして並び替えたりできます。

Outliner Window

(アウトライナーウィンドウ)




Outliner Window Header

このウィンドウにはシーン内の全てのオブジェクトのリストを表示します。これは、大きなシーン(沢山のオブジェクト)で作業するときにとっても役立ちます。

このウィンドウのヘッダからウィンドウに表示される要素の種類を選ぶことができます。

Timeline Window

(タイムラインウィンドウ)

このウィンドウでは、LMB によって移動できるタイムラインを表示します。

フレームをアレンジする。

Blenderは、斬新なスクリーン分割法を使用し、ウィンドウ・フレームをアレンジします。

デスクトップ上にアプリケーション・ウィンドウは長方形で表示されます。

Blenderはそれを多くのリサイズ可能なウィンドウ・フレームに分割します。

ウィンドウ・フレームは、3D Viewウィンドウまたはアウトライナー・ウィンドウのような特定のタイプのウィンドウ用の、作業領域を含んでいます。

それは、大きいアプリケーション・ウィンドウを幾つもの小さな(長方形の)重ならないウィンドウ・フレームに分けるという考え方です。

そのように、それぞれのウィンドウはいつも完全に表示されており、あるウィンドウで作業したり、別のウィンドウで作業したり、ウィンドウをまたがって作業することが非常に簡単です。

ウィンドウ・フレームを最大にする。

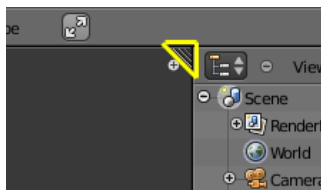
View → Toggle Full Screenでウィンドウ・フレームを最大にし、アプリケーション・ウィンドウ全体にウィンドウ・フレームを広げることができます。

元のサイズに戻すためには、もう一度View → Toggle Full Screenを使用します。

これは⇧ ShiftSpace, Ctrl↓またはCtrl↑を使うことで、「ウィンドウを最大化」と「ウィンドウを元に戻す」の切り替えをより素早く行うことができます。

注意:キーボード・ショートカットを使用することで最大にされるウィンドウは、あなたのマウスが現在乗っているウィンドウです。

ウィンドウ・フレームを分ける。



ウィンドウの右上側の角にある三角(右図の黄色の線で囲まれた部分)はスプリッター・ウィジェットです。

それは、ウィンドウを分割したり、結合したりします。

splitter widget(スプリッター・ウィジェット)をマウスオーバーすると、あなたのカーソルは十字に変化します。

それを左に LMB でドラッグすると垂直にウィンドウが分割され、下向きにドラッグすると水平にウィンドウが分割されます。

2つのフレームを結合する。

2つのウィンドウ・フレームを結合するためには、結合したい方向に対してそれらが同じ大きさでなければなりません。

例えば、横に並列している2つのウィンドウ・フレームを結合したいなら、それらは同じ高さでなければなりません。

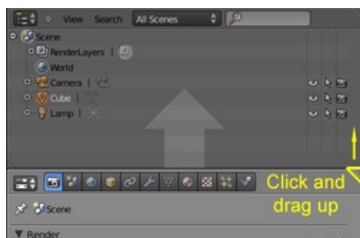
左の高さが右の高さと同じでない、水平にそれらを結合できません。

これは、結合した後のウィンドウ・スペースが長方形でなければならぬためです。

同じルールは、上下に分割された2つのウィンドウ・フレームを接合する場合でも成立します。

それらは両方、同じ横幅でなければなりません。

もし上のものを垂直に分割すれば、まず最初に、分割したそれら2つを結合してから、次にその結合した上部のものと下部のものを結合しなければなりません。



現在のウィンドウをその上のウィンドウに結合するためには、ウィンドウ・スプリッターをマウスオーバーし、ポインタが十字に変わる時、LMB でドラッグして、結合プロセスを開始します。(右図ではプロパティウィンドウが、“上の”Outliner Window(アウトライナー・ウィンドウ)と結合しようとしています)。

上側のウィンドウが少し暗くなり、その上の上を向いた矢印が表示されます。

これは、下側(現在の)のフレームがその暗くなったフレーム・スペースを「併合する」ことを示します。

結合するために LMB を放してください。

逆のことがしたいなら、オリジナル(下側の)のフレームの方にマウス・カーソルをドラッグしながら戻してください。

下向きの矢印が下側のフレーム上に表示されるでしょう。

同様に、ウィンドウは左から右に結合されます。その逆もまた同様です。

ドラッグを放す前に Esc を押すと、その作業は中止されます。

フレーム・サイズを変えること。

ウィンドウ・フレームの境界を LMB でドラッグすることによって、ウィンドウ・フレームをリサイズすることができます。


マウス・カーソルが二重頭の矢に変化するまで、2個のフレームの間の境界の上でマウス・カーソルを動かし、変わったら、ドラッグします。

中身を入れ替える。

ふたつのフレームの中身を入れ替えることができます。はじめのフレームのスプリッターのひとつを Ctrl LMB し、目的のフレームに向けてドラッグして、マウスをそこで離します。ふたつのフレームは隣り合わせになっている必要はありませんが、同じウィンドウの中になければなりません。

新しいウィンドウを開く。

Blender フレームを含んだ、新しい完全なウィンドウを作りたいと思うかもしれません。これは、たとえばモニタを複数台持っていて、Blender の同じインスタンスの、別の情報をそれぞれに表示させたいときに役立ちます。

必要な操作は、フレーム・スプリッタの上で **⇧ Shift LMB**  を押し、少しドラッグするだけです。最大化、最小化、クローズとその他のボタン(お使いのプラットフォームに依存します)を持った、新しいウィンドウがポップアップします。このウィンドウの中身は、あなたが操作を実行したはじめのウィンドウの複製を備えた、単体のフレームです。

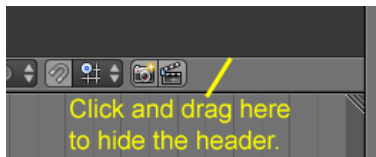
ウィンドウヘッダ

全てのウィンドウは1つのヘッダ(アイコンボタンを持つ薄い灰色の背景色の細長い部分)を持っています。ヘッダはツールバーとも呼ばれます。ヘッダはウィンドウの領域の上部(プロパティウィンドウ等)もしくは下部(3Dウィンドウ等)に有ります。下図は3Dウィンドウにおけるヘッダの例です。



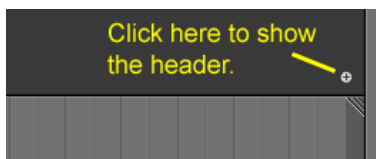
ウィンドウ上にマウスカーソルを移動させると、そのウィンドウのヘッダの色がわずかに明るい灰色になります。これはウィンドウが "フォーカスされた" 事を意味します。そのウィンドウに対してのみホットキーを押した影響が出ます。

ヘッダを隠す



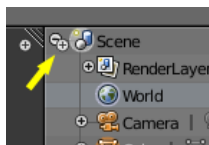
ヘッダを隠すためには、まずカーソルの位置をウィンドウとヘッダの間の細い線に合わせ、ポインタの形が上下の矢印の形になるようにします。続いて LMB を押しながらウィンドウの外にヘッダが消えるようにカーソルを移動させます。

ヘッダを表示する



ヘッダが隠れている場合は小さなプラスの印(右図参照)が表示されています。この印を LMB する事で、ヘッダは再び表示されます。

Note 1: 3Dウィンドウではこの小さなプラスの印が2つ以上(ウィンドウ上部及び左側と右側)表示されています。これらは第2のヘッダではなく、いくつかのツールを持ったパネルを表示させます。



Note 2: いくつかのウィンドウでは、アイコンの一部のように見えてしまうために、ここで述べた小さなプラスの印を見つけにくい場合があります。小さいプラスの印のような物を利用しているアウトライナーウィンドウはその一例であり、ヘッダを戻す妨げになる場合があります。

ヘッダの位置

ヘッダの位置を上部から下部、あるいは他の部分に移動するためには、そのヘッダ上で RMB を押して出現するポップアップメニューから任意の位置を選択するだけです。

💡 テーマの色について

Blender はユーザーの好みに合わせて殆どのインターフェースの色を設定によって変更できます。もしあなたが自身のスクリーン上で見た色と、マニュアルで見た色とが異なっている場合はあなたのデフォルト・テーマが変更されている可能性が有ります。新たなテーマの作成および現在のテーマからの変更と選択は [User Preferences](#) ウィンドウを選択して Themes セクションをクリックする事で行うことができます。

ウィンドウタイプボタン

ヘッダの左端にあるアイコンのボタンを LMB でクリックすることによって、16種類のウィンドウタイプを選択できます。Blender 内のウィンドウフレームは全て、どんなウィンドウタイプでも表示できます。もしあなたが全てのウィンドウフレームに3Dビューを表示したいならば、全てのウィンドウフレームを変更して下さい。

メニューとボタン類

ほとんどのウィンドウヘッダでは、ウィンドウタイプボタンのすぐ隣に表示されているメニュー類を-表示されている小さなマイナスの印によって-隠すことができます。もしあなたがマニュアルで言及された何らかのメニューを見つける事ができない場合、今一度ウィンドウタイプボタンの隣に有る小さなプラスの印を探してみてください。その印を LMB でクリックする事でメニューが表示されるようになります。

メニューを利用するとそこに有る項目を選択するだけで、多くの機能やコマンドに直接アクセスできます。

全てのメニューの項目は、存在すれば、項目に関連するホットキーのショートカットも表示します。

メニューとボタン類は利用しているウィンドウタイプや選択されたオブジェクトとモードによって変わります。利用可能なときのみ動作を表示します。

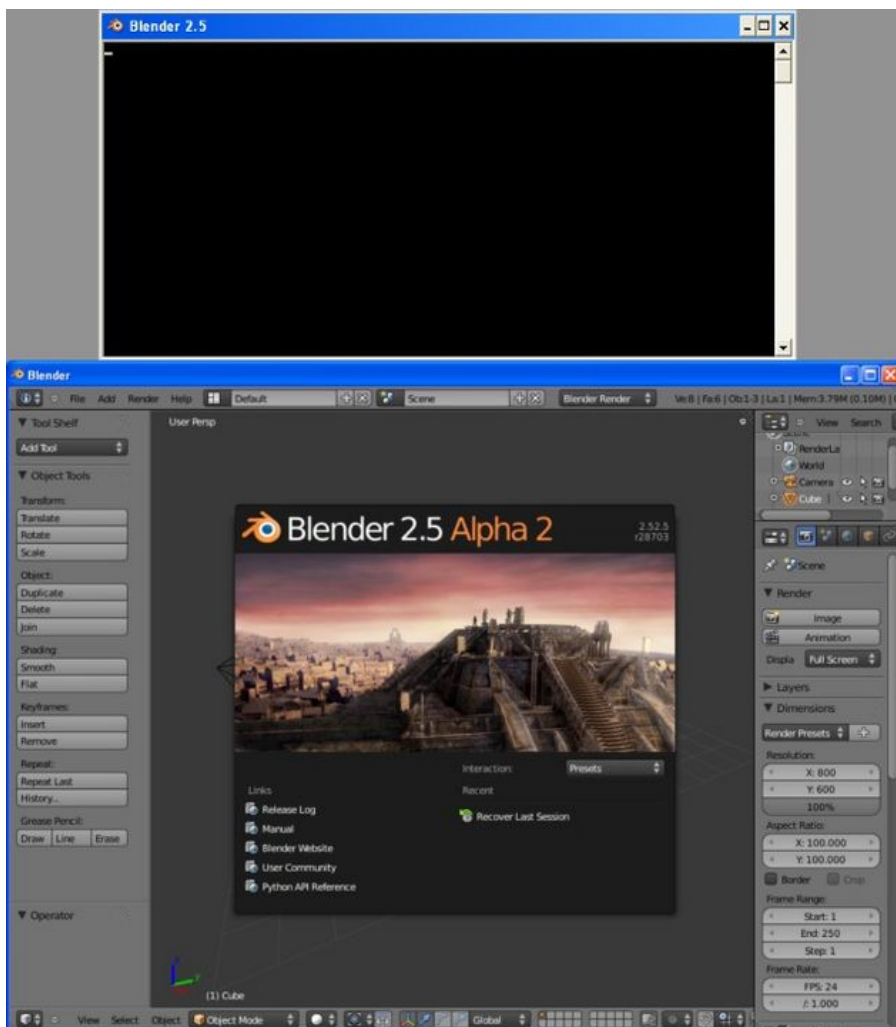
コンソールウィンドウ

コンソールウィンドウは Blender の運用状況、状態、内部エラーについてのメッセージを表示するオペレーティング・システムのテキストウィンドウです。Blender がクラッシュする場合に、解決の糸口を求めてコンソールウィンドウを参照するのは良い考えです。

Windows 2000/Xp/Vista/7

Blender が Microsoft Windows 系列の OS で起動された場合、最初にデスクトップ上に別のウィンドウとしてコンソールウィンドウが生成されます。状態が正常であることが確認されると、メインの Blender Application のウィンドウが同様に表示されます。

以下のスクリーンショットは Windows XP 上で2つのウィンドウを表示したものです:



Blender のコンソールウィンドウと Blender Application.

Blender のコンソールウィンドウは以下に示す理由で見えないことがあります:

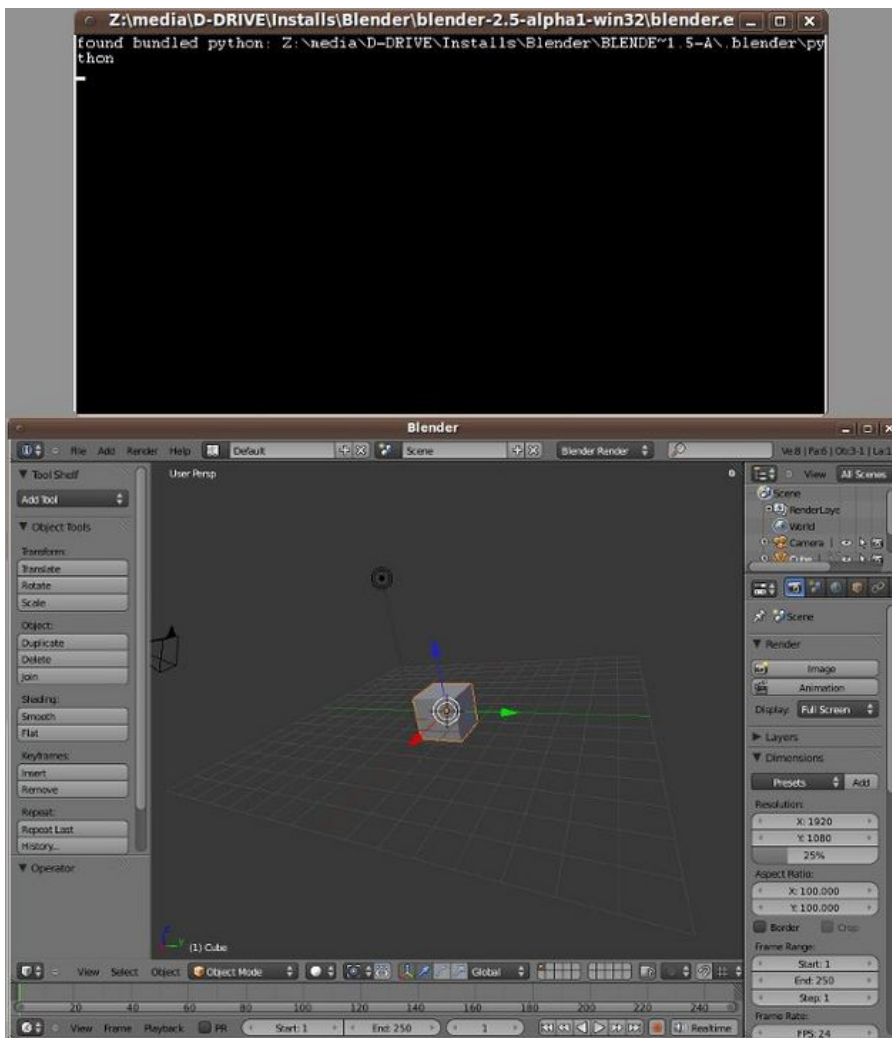
- Blender Application のウィンドウがコンソールウィンドウを覆ってしまっている場合があります。この場合は Windows のタスクバー上の Blender のコンソールウィンドウのアイコンをクリックする事で Blender のコンソールウィンドウが見えるようになります。
- Blender のコンソールウィンドウが Blender を起動した際に最小化されている場合があります。この場合でも、Windows のタスクバー上の Blender のコンソールウィンドウのアイコンをクリックするだけで、Blender のコンソールウィンドウが見えるようになります。

Linux

Linux環境におけるコンソールウィンドウは通常 Blender が Linux のターミナルかコンソールウィンドウから起動されたときのみ表示され、Blender はコンソール出力を表示するためそのコンソールウィンドウを利用します。

ほとんどの Linux のディストリビューションは Blender をそのパッケージからインストール可能なアプリケーションの一つとして備えています。Blender がその方法でインストールされた場合はそのシステムのメニューにも、Linux のコンソールやターミナルウィンドウを別に開かず Blender を起動できるようにアイコンが表示されるようになります。ターミナルからではなくアイコンを利用して Blender が起動された場合は Blender のコンソールウィンドウが [XWindows](#) が起動したターミナルによって隠される事がよくあります。

以下のスクリーンショットは Blender を起動した Linux のコンソールウィンドウです。Blender のコンソールの文字列はここに出力されます:



ターミナルから起動された Blender



Blender のコンソールウィンドウを閉じる

Blender のコンソールウィンドウは Blender が起動している間は開かれている必要が有ります。もしコンソールウィンドウが閉じられると、Blender も同様に終了し、全ての保存されていない Blender 内のデータが失われてしまいます！ Windows の MS-DOS コマンドウィンドウと Blender のコンソールウィンドウは見た目が似通っているため、正しいウィンドウを閉じるようにいつも心がける (あるいは Blender で作業している際はショートカットキーの Ctrl+W を利用して頻繁に保存する) 必要が有ります。

コンソールウィンドウの状態及びエラーメッセージ

コンソールウィンドウはとても多様な種類の状態とエラーメッセージを表示します。そのメッセージは重要でない物 (Blender の動作に実際に影響を与える物ではありませんが、Blender が何を行っているかの通知等) から致命的な物 (Blender の適切なタスクを阻害し、場合によっては反応が失われて完全に終了してしまうような重大なエラー等) まで多岐に渡ります。コンソールウィンドウのメッセージは複数の場所 (いくつか例示すると、Blender の内部コード、外部の [Python スクリプト](#)、多様な TODOLINK-Plugins 等) から発せられた物です。

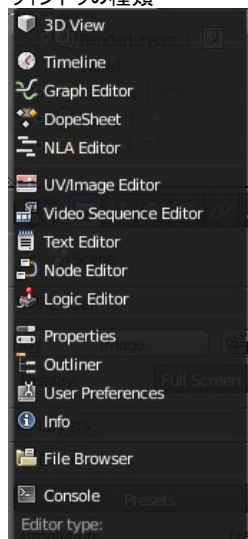
- found bundled python: (FOLDER)

このメッセージは Blender が、内蔵された [Python](#) インタプリタのライブラリを検出できた際に表示され、エラーが発生した場合はこのメッセージが表示されません。

- malloc returns nil()

Blender は新たにメモリ (RAM) を必要とするタスクを起動した際、malloc (memory allocate の略) という関数を利用して Blender が要求された容量のメモリを確保しようとします。しかし、Blender に要求されたメモリの容量を満足できない場合、malloc 関数は nil、null もしくは 0 を返し、要求に応じられないことを明示します。もしこのエラーが発生した場合、Blender はユーザーの要求に応じるためのタスクを実行できなくなります。このエラーは Blender の終了あるいは非常に緩慢な動作や反応の喪失という結果を招きます。メモリを使い切る事を防ぐためには、メモリの増設、Blender のモデルの品質を落とす、Blender が利用可能なメモリを得るために他のプログラムやサービスを終了する等の対策を講じることができます。

ウィンドウの種類



ウィンドウタイプ選択メニュー

Blenderインターフェースでは、OSが提供する四角いウィンドウをいくつかの四角いウィンドウフレームに分割して表示します。それぞれのウィンドウフレームには、ウィンドウの種類によって異なるタイプの情報が含まれています。

ウィンドウフレームは他のウィンドウフレームとは独立して動作し、いくつかの同じタイプのウィンドウを表示することが可能です。例えば、異なる視点から見たシーンを複数の3Dウィンドウに開いたり、作業中の作品に都合のよいように、ウィンドウフレームを分割したり結合したりリサイズしたりすることができます。また、スクリーンスペースを節約するために、いくつかのウィンドウフレームのヘッダを表示したり非表示にすることができます。

ウィンドウフレームの設定については[ここ](#)をご覧ください。

ウィンドウの種類はその機能によって下記のように分けられます。

- [3D View](#) - シーンを表示します。
- [Node Editor](#) - テクスチャ、マテリアル、コンポジットのノードを編集します。
- [UV/Image Editor](#) - 高度なUVマネージメントツールによる画像エディタです。
- [Properties Editor](#) - 選択中のオブジェクトの属性を表示します。
- [File Browser](#) - ファイルを管理したり、ロード／セーブを行います（多くの場合は必要なときに自動的に表示されます。）。
- [Outliner](#) - オブジェクトの管理に役立ちます。
- [Timeline](#) - アニメーション再生の制御を行います。
- [Graph Editor](#) - アニメーションキーやそれらの内挿／外挿を編集します。
- [NLA Editor](#) - 非線形アニメーションシーケンスの編集します。
- [Dope Sheet](#) - 各々のアクションをアクションシーケンスに結合します。
- [Video Sequence Editor](#) - ビデオシーケンスをフィルムストリップに組み上げます。
- [Logic Editor](#) - ゲームロジックを編集します。
- [Text Editor](#) - プロジェクトに関するメモや文書を作ったり、Pythonスクリプトを作成します。
- [Console](#) - Blenderで直接Pythonを実行します。
- [User Preferences](#) - Blenderでの作業スタイルやコンピュータ設定をカスタマイズします。

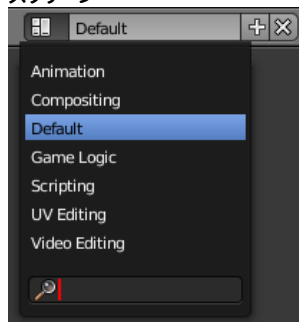
ウィンドウヘッダの一番左にあるボタンを用いてウィンドウの種類を選択することができます。ポップアップメニューに選択可能なウィンドウタイプが表示されます（ウィンドウタイプ選択メニュー画像を参照）。

それぞれのウィンドウタイプの詳細は、上記のリンクをクリックするか、リファレンスのセクション [III ウィンドウ](#) をご覧ください。

参考

- [ボタン](#)
- [ウィンドウシステム](#)

スクリーン



レイアウトリスト

Blenderの自由度の高いウィンドウシステムによって、モデリング、アニメーション、スクリプティングなどの様々なタスクに対してカスタマイズした作業環境を設定することができます。これはファイルを編集中に異なる環境へ素早く切り替えるのに役立ちます。シーン毎に、モデリングしたりマテリアルを使って飾り付けたりすることなどによってステージを設定する必要があります。[ウィンドウシステム](#)の画像ではモデリングステージで作業しています。

Blenderには、主要な制作ステップ毎での作業のために、予め設定されたスクリーンやウィンドウレイアウトが用意されています。これらのスクリーンには効率の良い作業に必要な様々なウィンドウが表示されています。もし特定のスクリーンを見つけられなければ、右図のリストの一番下にある検索機能を用いて探し出すことができます。

Animation

アクターやその他のオブジェクトを動かしたり形や色を変化させます。

Compositing

シーンの様々なパートを結合し(背景やアクター、特殊効果など)、フィルタ(色補正など)をかけます。

Default

Blenderの開始時や新規ファイル時に使われるレイアウトで、新しいオブジェクトをモデリングするときに有用です。

Game Logic

Blender内でのゲームのプランニングやプログラミングをします。

Scripting

作品の文書化や、Blenderを自動化するスクリプトを書きます。

UV Editing

オブジェクトメッシュを2D平面に投影し、その表面上のテクスチャマップをコントロールします。

Video Editing

アニメーションシーケンスを編集します。

これらのスクリーンレイアウトはアルファベットと数字でソートされてに並んでいます。このリストは情報ウィンドウのヘッダにあります(プリセットスクリーンでは常にレイアウトの最上部にあるので、Blenderの初心者にはよくメニューバーと間違えられますが、実際にはこれはヘッダのみを表示したウィンドウです。)

(アルファベット順で) 次のスクリーンに切り替えるにはCtrl→を、前のスクリーンに切り替えるにはCtrl←を押します。




スクリーンとシーンセレクタ


デフォルトでは、各スクリーンレイアウトは最後に使ったシーンを記憶しています。異なるレイアウトを選択すると、そのレイアウトに切り替わって、さらに、そのシーンに移動します。

[ウィンドウシステム](#)や[ウィンドウの種類](#)にて説明したような、ウィンドウに対する全ての変更は一つのスクリーン内に保存されます。あるスクリーン内のウィンドウを変更しても、他のスクリーンに影響を与えることはありません。


スクリーン設定

新規スクリーンを追加

追加ボタン(+)をクリックします。現在のレイアウトを基にした新しいフレームレイアウトが生成されます。


矢印キーを使った切り替えの際に予測がつくように、新しいスクリーンには名前だけでなく番号も付けるとうまいでしょう。レイアウトの名前は、フィールドをLMB()をクリックし新しい名前を入力するか、編集したい箇所をもう一度クリックすることによって変更できます。例えば、"6-MyScreen"という名前を使うことができます。(スクリーンとシーンセレクタを参照のこと。)

スクリーンの削除

Delete datablock ボタン()をクリックするとスクリーンを削除できます。(スクリーンとシーンセレクタを参照のこと。)

スクリーンの再構成

フレームの境界を動かしたり、ウィンドウの分割や結合を行うには、ウィンドウコントロールを使います。気に入ったレイアウトを設定できたら、CtrlUIによってユーザデフォルトを上書きしましょう(注意:現在のシーンがデフォルトの一部となるので、単純なシーンを開いて設定したレイアウトにするべきです)。

プロパティウィンドウには特別なオプションがあります。背景上でRMB()をクリックすると、パネルを水平方向(左右)や垂直方向(上下)に並べ替えることができます。

デフォルトの上書き

.blendファイルを保存すると、スクリーンレイアウトもそのファイルに保存されます。ファイルを開く時には、ファイルブラウザヘッダにあるLoad UI チェックボックスによって、ファイル中のスクリーンレイアウトか現在のレイアウトかのどちらを使うかを選ぶことができます。Load UI がチェックされていれば、ファイル中のスクリーンレイアウトが使用され、作業中のデフォルト設定はそのファイルによって上書きされます。

追加レイアウト

Blenderを上手く扱えるようになってきたら、完璧なワークフローのために何か別のスクリーンレイアウトを加えてみてはどうでしょうか。これによって劇的な機能の向上もたらされるでしょう。

1-Model

4つの3Dウィンドウ(上面、正面、側面と透視図)と編集のためのプロパティウィンドウ

2-Lighting

光源を動かすための3Dウィンドウと、レンダリング結果を表示するUV/イメージ編集ウィンドウ、レンダリングや照明プロパティをコントロールするためのプロパティウィンドウ

3-Material

マテリアル設定のためのプロパティウィンドウ、オブジェクト選択のための3Dウィンドウ、アウトライナー、ライブラリスクリプト(もし使っているなら)、ノードエディタ

4-UV Layout

UV/イメージ編集ウィンドウ、シーミングとメッシュ展開のための3Dウィンドウ

5-Painting

テクスチャペインティングのためのUV/イメージ編集ウィンドウ、オブジェクトに直接ペインティングするための3Dウィンドウ、背景に参照用画像を配置した3つのミニ3Dウィンドウをずらっと端側に縦に並べて、プロパティウィンドウ

6-Animation

グラフエディタ、アーマチャのポージングのための3Dウィンドウ、NLAウィンドウ

7-Node

入り組んだノードのための大きなノードエディタウィンドウ、レンダリング結果とリンクしたUV/イメージ編集ウィンドウ

8-Sequence

グラフエディタ、イメージプレビューモードのビデオシーケンスエディタ、タイムラインモードのビデオシーケンスエディタ、タイムラインウィンドウ、そして古き良きプロパティウィンドウ

9-Notes/Scripting

アウトライナー、テキスト(スクリプト)エディタウィンドウ

レイアウトの再利用

新しいウィンドウレイアウトを作って、それを今後の.blendファイルで使いたいと思ったら、CtrlUと押してユーザデフォルトとして保存してください(注意: 全てのスクリーンとシーンもデフォルトとして保存されます)。

Scenes

(シーン)

シーンはプロジェクト管理にとっても役立つツールです。Blender を最初に開いたときに見る、立方体モデルの置かれた空っぽの空間がデフォルトのシーンです。シーンはウェブブラウザのタブに似たものと考えることができます。たとえば、ウェブブラウザは同時にいくつもタブを持てます。タブは空にすることも、同じウェブページのまったく同じ視界 (views) を表示することも、同じページの別の視界、あるいは完全に別のページを取り混ぜて表示することもできます。Blender のシーンはだいたい同じやり方で動作します。空のシーン、シーンの完全に独立したコピー、もしくは元のシーンに複数の方法でリンクする新しいコピーを持つことができます。

情報ウィンドウヘッダ (Blender レイアウトの最上部にあるバーです。スクリーンとシーンセクタを参照のこと) 中の シーンセクタ (Scene selector) より、シーンを選択したり生成したりできます。



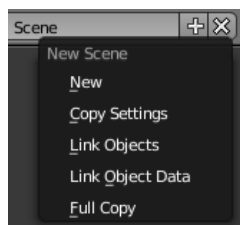
スクリーンとシーンセクタ

Scene configuration


(シーンの設定)

Adding a new Scene

(新しいシーンの追加)



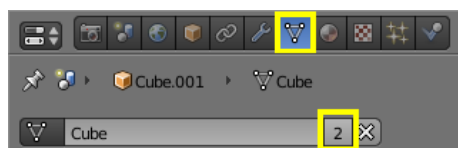
シーン追加メニュー

シーンセクタにある  をクリックすると新しいシーンを追加することができます。新しいシーンを作成する時に、コンテンツのコピーについて5つのオプションから選ぶことができます (シーン追加メニュー)。

これらのオプションを選択するには、オブジェクト (Object) とオブジェクトデータ (Object Data) の違いを正しく理解しておく必要があります。Blender のグラフィック要素 (メッシュ、光源、曲線、など) は、オブジェクト (Object) とオブジェクトデータ (Object data, ObDataともいう) の2つから成り立っています。オブジェクトは、位置、回転や個々の要素のサイズといった情報を保持しています。オブジェクトデータは、個々の種類の要素のインスタンス毎に共通の情報、例えばメッシュやマテリアルリストなど、を保持しています。各オブジェクトは、その関連するオブジェクトデータに対するリンクを持っていて、1つのオブジェクトデータは多数のオブジェクトによって共有されています。

従って、この5つのオプションから、現在選択中のシーンから新しいシーンにどのような情報がコピーされ、どのような情報が共有 (リンク) されるかを決定することができます。

- New (新規): 空 (から) のシーンを生成します。新規シーンではレンダリング設定がデフォルトに戻ります。
- Copy Settings (設定をコピー): 空のシーンを作成し、直前のオプションをコピーします。レンダリング設定もコピーされます。
- Link Objects (オブジェクトをリンク): 選択可能な中で最も浅いコピー形式です。このオプションでは、現在選択中のシーンと同じコンテンツを持つ新しいシーンが生成されます。何もコピーされませんが、その代わりに、新しいシーンには古いシーンのオブジェクトへのリンクが含まれます。つまり、新しいシーンでの変更はコピー元のシーンにも反映されます。なぜならば、新しいシーン中のオブジェクトは現在選択中のシーン中のオブジェクトと文字通り全く同じものだからです。逆に元のシーンの変更は新しいシーンにも反映されます。
- Link Object Data (オブジェクトデータをリンク): 現在選択中のシーン中の全てのオブジェクトが複製されますが、これらの複製されたオブジェクトは、元のシーン中の対応するオブジェクトのメッシュ、マテリアルなどのオブジェクトデータに対するリンクを持っています。これは、新しいシーンでは、元のシーンに影響を与えることなくオブジェクトの位置や向きが変更できるけれども、メッシュやマテリアルなどのオブジェクトデータに対する変更は元のシーンに影響を与えることを意味しています。(この理由は、オブジェクトデータの1つのインスタンスが、全てのシーン中の全てのオブジェクトによって共有されているからです。もし、新しいシーン中のオブジェクトを他のシーン中のオブジェクトとは別個に変更したいならば、プロパティウィンドウの Object Data パネルの数値を使って、新しいシーン中でのそのオブジェクトを手作業で single-user コピーにする必要があります。これによって、そのオブジェクトデータは新規の独立したものとなります。)



- Full Copy (フルコピー): 選択可能な中で最も深いコピー形式です。何も共有されません。このオプションでは、現在選択中のシーンのコンテンツのコピーを含む、完全に独立したシーンが生成されます。元のシーンの全てのオブジェクトは複製され、同様に、複製された専用のオブジェクトデータも生成されます。

Blender のデータ処理方法についてより深く理解するためには、[Blenderのライブラリとデータシステム](http://amrc.altervista.org) を参照してください。

A brief example

(手短な例)

映画のバーのシーンを考えてみましょう。初めに、きれいなバージョンのバーを作ります。このバージョンでは何も壊れておらずすべて適切な位置に置かれています。次に筋の展開にあわせた別のシーンを作ることに決めました。シーン内の動きはどのリンク方式がシーンに最適なのか(リンクを使う場合)を示しています。

Link Objects

すべてのオブジェクトが元のシーンにリンクされます。壁の配置を直すと、このバーを舞台として使う全シーンで壁が動きます。

Link Object Data

オブジェクトの位置を変更する必要があるものの、形やマテリアル設定は一定のままにしたいときに便利です。たとえば椅子は、「混雑するバー」のシーンには床の上に立たせ、「もうすぐ閉店」のシーンではテーブルの上に置くことができます。椅子が形を変えることはないので、メッシュをまるごとコピーしてメモリを浪費する必要はありません。


Full Copy

床に落ちて割れるグラス。メッシュの形を変えるのでコピーが必要です。

上記のすべてを同じシーン内で行うことはできませんが、ものによって別の方法でリンクするわけを理解するのに役立つでしょう。

Deleting a Scene

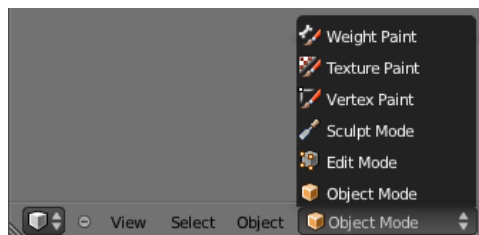
(シーンの削除)

シーンセレクトタから Delete datablock ボタン  をクリックすると、シーンを削除できます。(スクリーンとシーンセレクトタ)を見てください。

Modes

(モード)

モードとは、Blenderレベルのオブジェクト指向の機能で、これは *Blender アプリケーション全体* が常に *ただひとつのモード* にあり、選択しているアクティブオブジェクトのタイプによって、利用できるモードが変化するという事です。オブジェクトのほとんどは(たとえばカメラやランプなど)、デフォルトのオブジェクトモードだけを有効にします。モードはそれぞれ、選択しているオブジェクトのひとつの特徴を編集する設計になっています。詳細は下記 (*Blender のモード*)の表を参照してください。



モード選択の例(メッシュオブジェクト)

今いるモードは、3Dビュー のヘッダにある モード ドロップダウンリストで設定します (*モード選択の例(メッシュオブジェクト)*参照)。



オブジェクトの選択は オブジェクト モードでのみ行えます。他すべてでは、カレントオブジェクトの選択は “ロック”されています (例外として、ある程度までは、アーマチュアの ポーズモードでも行えます)。

モードは、Blenderの多くのものに影響を与えます:

- プロパティウィンドウの一部のコンテキストで、使用可能なパネルや操作項目を変更することができます。
- 例えば UV/Image Editor のように(もちろん 3Dビュー でも!)、ウィンドウ全体の振る舞いを変更することができます。
- 利用可能なヘッダーツールを変更することができます(メニューやメニュー項目だけでなく、その他の操作項目も...)。例えば 3Dビュー ウィンドウでは、オブジェクトモードの Object(オブジェクト) メニューは 編集 モードでは Mesh(メッシュ) メニューに変わり(アクティブなメッシュオブジェクトがあるとき!)、頂点ペイントモードでは Paint(ペイント) メニューに変わります...

Blenderのモード

アイコン	名前	ショートカット	備考
	Object オブジェクト モード	なし ¹	デフォルトのモードであり、すべてのオブジェクトタイプで使用可能。Object データブロックの編集が主用途(すなわち、位置/回転/サイズ)。
	Edit 編集モード	⇔ Tab ¹	すべてのレンダリング可能なオブジェクトタイプで使用できるモードで、“形状”ObData データブロックの編集が専らの用途(すなわちメッシュの頂点/辺/面、カーブ/サーフェスの制御点、等)。
	Sculpt スカルプトモード	なし ¹	メッシュ専用のモード、Blenderのメッシュ3D-スカルプトツールを利用できる。
	Vertex Paint 頂点ペイント モード	なし ¹	メッシュ専用のモード、メッシュの頂点の色を設定する(頂点を“ペイント”する)ことができます。
	Texture Paint テクスチャペ イントモード	なし ¹	メッシュ専用のモード、3Dビューで、モデル上にメッシュのテクスチャを直接ペイントできるようになります。
	Weight Paint ウェイトペイン トモード	Ctrl⇔ Tab ²	メッシュ専用のモード、頂点グループの重み付け専用。
	Particle パーティクル モード	なし ¹	メッシュ専用のモード、パーティクルシステム専用。編集可能なシステムで有用(髪)。
	Pose ポーズモード	Ctrl⇔ Tab ²	アーマチュア専用のモード、もっぱらアーマチュアにポーズをつける用途に用いられる。

モードのショートカットに関する注記:

1. ⇔ Tabは、デフォルトでは Edit(編集) モードをトグル切り替えます。
2. Ctrl⇔ Tabは、ウェイトペイント(メッシュ)/ポーズ(アーマチュア)モードと、現在の別のモード(デフォルトでは オブジェクト モード)を切り替えます。ただし同ショートカットは、一部のモード内で別の意味を持つことがあります。

ご覧のように、ショートカットを用いたモードの切り替えは、特にメッシュではかなり大変になることがあります...

モードの使用法のさらなる詳細は、ここでは解説しません。これについては大半が[モデリングの章](#)で議論されていますが、これはモードの使用が大抵の場合モデリングと関係があるためです。Particle(パーティクル)モードは [パーティクルの章](#)で、アーマチュア用の Pose(ポーズ) モードと Edit(編集) モードは [リギングの章](#)で解説されています。

備考

このマニュアルを読んでいて、一部のボタンやメニューオプションが見本どおりに画面に表示されない場合は、そのオプションの有効化に妥当なモードに

なっていない可能性があります。

Contexts










(コンテキスト)

プロパティ(またはボタン)ウィンドウには、コンテキストがいくつか表示されていて、その選択はヘッダー内に並んだアイコンを介して行えます(コンテキストボタンの例を参照)。



コンテキストボタンの例

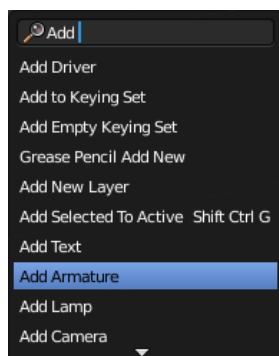
ボタンの種類と数は、有益なボタンだけを表示するように、選択したコンテキストによって変わります。ボタンの並びは下記の順に従います:

-  [Render\(レンダー\)](#): レンダリング出力に関連するすべてのもの(寸法、アンチエイリアシング、パフォーマンスなど)。
-  [Scene\(シーン\)](#): シーンの重力、単位及びその他の一般的な情報。
 - [World\(ワールド\)](#): 環境照明、空、霧、星やアンビエントオクルージョン。
 - [Object\(オブジェクト\)](#): トランスフォーメーション、表示オプション、可視性の設定(レイヤを経由して)、複製の設定とアニメーション情報(オブジェクトの位置関連)。
 -  [Constraints\(制約\)](#): オブジェクトのトランスフォーム(位置、回転、スケール)、トラッキング、それに関係の特性を制御するために使用します。
 -  [Modifiers\(モディファイア\)](#): ジオメトリを変更することなくレンダリングや表示方法を変えることで、非破壊的にオブジェクトに影響を与えられる処理(例えばミラーやスムージング)。
 -  [Object Data\(オブジェクトデータ\)](#): オブジェクト固有のデータ(ランプの色、カメラの焦点距離、頂点グループなど)をすべて含みます。アイコンはオブジェクトのタイプによって異なります(ここで示されているものは、メッシュオブジェクト用です)。
 -  [Materials\(マテリアル\)](#): 表面(カラー、スペキュラ、透明度、等)に関する情報。
 -  [Textures\(テクスチャ\)](#): 詳細(色、透明度、3次元深度)を追加するためにマテリアルが使用します。
 -  [Particles\(パーティクル\)](#): 不定量の(通常は小さな)オブジェクト、たとえばフォース・フィールドで操作可能なライトやメッシュ・オブジェクトと、その他の設定を追加します。
 -  [Physics\(物理演算\)](#): 布、フォースフィールド、衝突、流体と煙のシミュレーションに関連するプロパティ。


各コンテキストのボタンは、[パネル](#)にグループ化されています。


Menus

(メニュー)



The Space-menu

Blenderには多数のメニューがあり、各メニューへのアクセスは、ウィンドウのヘッダーか、ホットキーを使用してマウス位置から直接、またはウィンドウの境界上やボタンや画面上のどこかでRMB  クリックして行えます。文脈依存のメニューは、そのインターフェイス要素に対して使用可能な場合にだけ表示されます。

それだけでなく、Spaceを押すことで、すべての Blender コマンドへアクセスできるメニューを使えます(写真のように)。そのまま必要なコマンドの名前を入力し始めると、検索してメニューの残りの部分を補足してくれます。リストが十分に絞られたら、目的のコマンドでLMB 、または↓↑でハイライトを動かしてReturnで選択しましょう。

バージョン 2.4x の古いツールボックスにあったメニューを懐かしんでいるのであれば、同様のものを 3D View: Dynamic Spacebar Menu アドオンで追加することができます。これは Preferences(ユーザー設定)ウィンドウの Add-Ons(アドオン) タブからインストールすることができます。

[アドオンをインストールする方法の詳細を読む](#) »

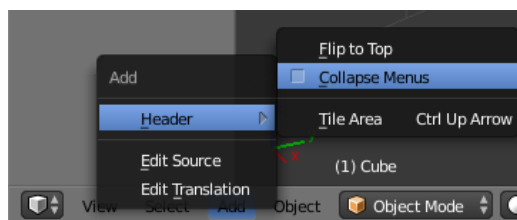
一部のメニューは文脈依存で、特定の状況下でのみ使用可能です。例えば specials(スペシャル)メニュー(W)は Edit(編集)モード がアクティブである間の 3Dウィンドウでのみ使用可能です。

Blender を使用している間は、どのモードがアクティブになっているか、そしてどのタイプのオブジェクトが選択されているかに気をつけましょう。これは、どの場面でのホットキーが動作するかを知るにも役立ちます。

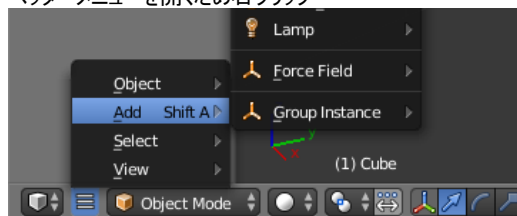
Collapsing Menus

(メニューの折り畳み)

メニューを折り畳んで、ヘッダーに横に余分なスペースが得られると便利ことがあります。この機能はヘッダーのコンテキストメニューからアクセスできます。単にヘッダーを右クリックして Collapse Menus(メニュー非表示) 設定を有効にします。



ヘッダーメニューを開くため右クリック



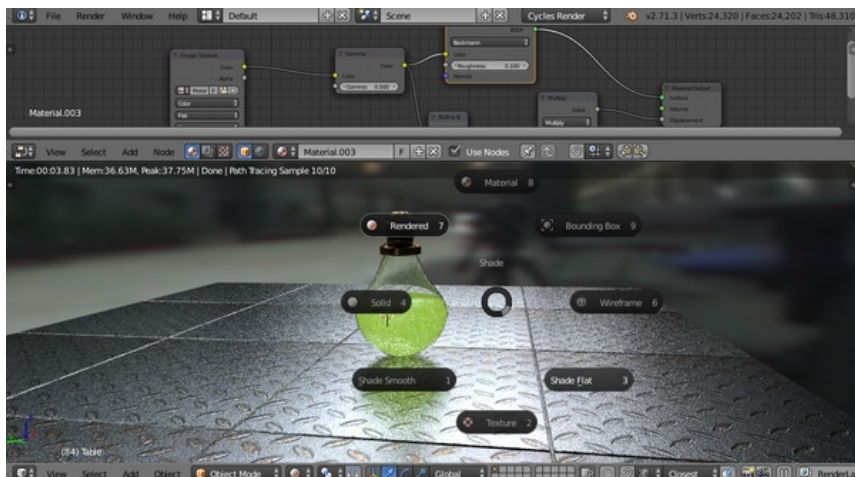
折り畳まれたアイコンからメニューを開く

Macでのメニュー

Blender は標準的な OS のメニューシステムを使用していないため、Macを使用している場合には、画面最上部でメニューバーが重なることがあります。これを取り除くには、Macworld誌の [この記事\(英文\)](#) を参照してください。ただし、やや複雑であることに注意してください。別の方法として、Lion 以降の Mac OS X では Blender ウィンドウのタイトルバー右端にある、フルスクリーン化のボタンをクリックします。それ以前は Blender の情報ウィンドウのヘッダの最後に、フルスクリーン化ボタンがありました。

パイメニュー

パイメニューとは、マウスの周囲に放射状にアイテムが広がるメニューのことです。(訳注:この機能はバージョン 2.72 からアドオンとして組み込まれています)



動作中のパイメニュー。現在有効なオプション "Rendered" が明るいアウトラインでハイライト表示されており、その一方で確定すると選択されるパイメニューアイテム "Shade Flat" が少し明るくハイライト表示され、円盤状のウィジェットはその方向を示しています

使用方法

パイメニューはデフォルトの Blender には含まれていません。アドオンによって利用可能になります。初回では、Blender に以下のパイメニューを提供する最小のアドオンが含まれています。

- ⇐ Tab : インタラクションモード
- Z : 描画モード+ソリッド・スムーズシェーディング
- Q : ビュー方向+透視/平行投影+カメラ
- ⇐ Tab⇧ ShiftCtrl : スナップ
- . : ピボット
- CtrlSpace : マニピュレータ

これらのメニューを有効にするには、User Preferences (ユーザー設定) の Add-Ons (アドオン) タブに行き、"User Interface" カテゴリの "Pie Menus Official" アドオンをチェックしてください。

パイメニューは上記のキーに割り当てられていたオペレータを完全に上書きします。ただしオペレータとパイを同じキーで併用できる **"Sticky"** オペレータが開発中です。

インタラククション

パイメニューを表示するには、単にメニューを表示するキーを押すだけです。

- マウスを移動せずにキーを離すとメニューが開いたままになり、その後、ユーザはマウスポインタをパイメニューアイテムのある方向に移動してクリックすることで選択が可能になります (クリックスタイルインタラククションと呼びます)。
- マウスをパイメニューアイテムの方向に移動後、キーを離すとメニューが閉じ、メニューアイテムがアクティブとして選択されます (ドラッグスタイルインタラククションと呼びます)。

パイメニューの中心の開いた円盤ウィジェットにより、ユーザはパイメニューの現在の方向を知ることができます。選択されたアイテムもハイライト表示されます。

パイメニューのアイテム選択の方向は、メニュー中心にある円盤ウィジェットにマウスカーソルが触れるか越えることで有効になります。このウィジェットの半径は、User Preferences (ユーザー設定) の Threshold (しきい値) オプションによりコントロール可能です。

すべてのパイメニューではないですが、現在アクティブなオプションも明るい縁によってハイライト表示されます(上の画像参照)。

リセンタリング

ウィンドウシステムはパイメニューをウィンドウ境界内に収めようとします。これにより、ユーザがウィンドウ境界の近くでパイメニューを表示しようとする時に、その初期位置も必然的に変更されることになります。その結果初期方向は中立ではなくなり、境界に一番近いアイテムを指します。その対策として、Recenter Timeout (リセータータイムアウト) オプションがあります。このタイムアウトを増やすと、パイメニューはこの期間、パイの方向を計算するのに初期マウス位置を使用するようになります。これにより、高速にドラッグ選択できるようになります。

アイテムの位置

体に覚えさせるため、パイメニューのアイテム位置は同じになるようにしてあります。つまり、もしコンテキストによっていくつかのオプションが欠けても、他のアイテムは空きを埋めるように移動したりしないということです。

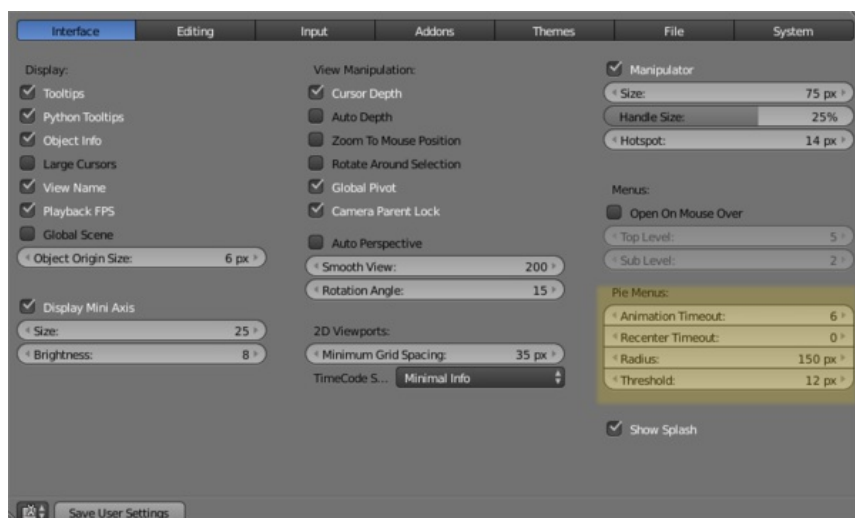
例: メッシュには PoseMode (ポーズモード) がありません。そのため、⇐ Tab のパイメニューはこのオプションを表示しません。しかし、Object Mode (オブジェクトモード) と Edit Mode (編集モード) オプションは同じ方向のままです。これは新しいアイテムを既存のメニューに追加した時にも重要で、この時も従来の位置を変更してはいけません。

キーインタラクション

パイメニューアイテムはキーアクセラレータに対応しています。キーアクセラレータは各メニューアイテムの、下線が引かれた文字になります(訳注:英語UIのみ)。また、数字によるアイテムの選択も可能です。位置同様、この番号もコンテキストの所為でアイテムが追加されたり欠けても変更されないよう、各数字は特定の方向に固定されています。ナンバリング方法は5を中心とした、テンキーの数字の方向に合致するようになっています。

オプション

User Preferences (ユーザー設定)にはパイメニューの操作を調整するオプションがいくつかあります。



User preferences

Animation Timeout(アニメーションタイムアウト)

この数値はその時間で、パイメニューが開くアニメーションが終了するまでを1/100秒単位で指定します。

Recenter Timeout(リセッタータイムアウト)

もしパイメニューがスクリーン境界に近い場合、常に表示されるようにシステムがパイメニューをセンタリングします。つまりユーザがドラッグした方向が、最初に意図していたのとは違う可能性があるということです。

しかし、この Recenter Timeout が過ぎる前に計算された方向は、つねにメニューが表示された初期位置を使用するため、メニューがリセッターされた後でも、すばやくジェスチャして使用することができます。

Radius(半径)

パイメニューの中心からメニューアイテムへの距離(ピクセル単位)。

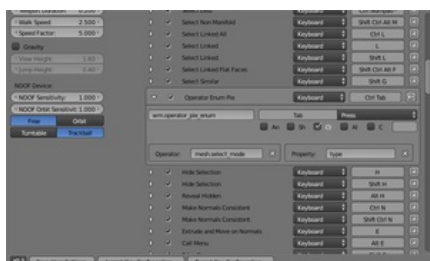
Threshold(しきい値)

パイメニューの中心から、最低この距離だけマウスカーソルが離れた時のみ、選択が有効になります。これにより、不用意にアイテムを選択してしまうことが少なくなります。

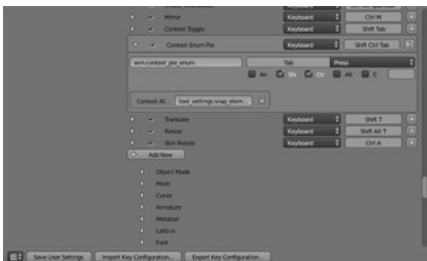
Blender へのパイメニューの追加

パイメニューを追加する方法は色々あります。ユーザ用にとって恐らく最も簡単なのは、あるオペレーターまたはコンテキスト列挙用のキーマップを変更し、パイメニューアイテムを表示する方法で、それぞれ専用のオペレーターがあります。

`wm.operator_pie_enum` はオペレーターのEnum(列挙)プロパティを元にしたパイを表示し、`wm.context_pie_enum` はコンテキストプロパティの値の列挙から選択するパイを表示します。下はこれら二つのサンプルです。



Operator Enum(オペレーター列挙)パイ。これは "type" 列挙プロパティをアイテム上に使用したパイメニューを、Edit Mode(編集モード)選択オペレーターで表示します

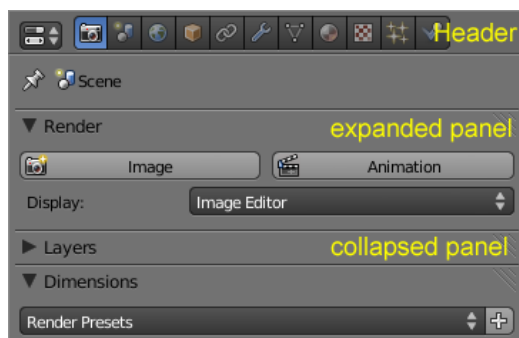


Context Enum(コンテキスト列挙)パイ。これは3Dビュー内で使用されるスナップモードのパイメニューを表示します

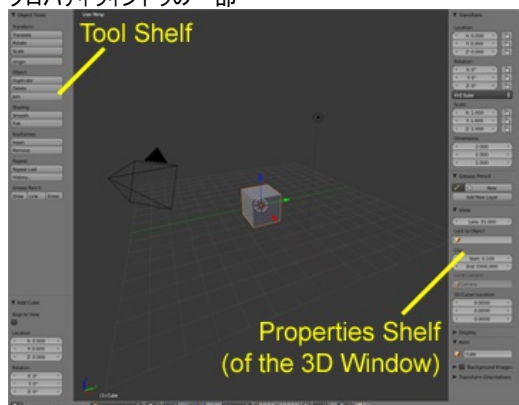
パイメニューは Python でコーディングすることもできます。詳細は[こちらのページ\(英文\)](#)をチェックしてください。

Panels

(パネル)



プロパティウィンドウの一部




3D ウィンドウ内にある二つのシェルフ





パネルは、一般的に Properties (プロパティ) ウィンドウ (バージョン 2.4x のボタンウィンドウ) に現れます。これはデフォルトの画面レイアウトの右側に表示されるものです (図 [プロパティウィンドウの一部](#) 参照)。

パネルはまた、表示非表示を切り替えられる 3D ウィンドウの部品 Tool shelf (ツールシェルフ) や Properties shelf (プロパティシェルフ) にもあります。ツールシェルフを表示するには、View (ビュー) » Tool (ツール) を使用するか、T を押しましょう。プロパティシェルフを表示するには、View (ビュー) » Properties (プロパティ) を使用するか、N を押しましょう (図 [3D ウィンドウ内にある二つのシェルフ](#) を参照)。

プロパティウィンドウに収められているヘッダでは、いくつかの [Contexts \(コンテキスト\)](#) を選択できます。各 Context にあるパネルの量や種類はさまざまです。例えば レンダーコンテキストには、レンダリング出力の大きさやアンチエイリアシングを変更できるパネルがあり、それに対してマテリアルコンテキストには、色、透明度、テクスチャなどを設定できるパネルがあります。

プロパティウィンドウ 内のパネルは、プロパティウィンドウ 内で RMB  で呼び出すメニューでお望みのオプションを選択すれば、水平または垂直に整列させることができます。ただし プロパティウィンドウ 内のパネルは垂直方向の配置に最適化されていることに注意してください。水平方向の配置は扱いにくいかもしれません。

パネルの配置とビューは、あなたの好みに変更することができます。たとえば、パネルには次の操作を行います:

- 右上隅にあるウィジェット (これはフレーム分割ウィジェットに似ており、三角形に3本のラインが入っています) を LMB  でクリック、ホールド、ドラッグすると、ウィンドウ (またはシェルフ) のあちこちに動かせます。
- Wheel  を使用すると、上下にスクロールします
- Ctrl MMB  を押したままマウスを左右に動かすと、ズームインとズームアウトをします。
- パネルのヘッダの左側にある黒一色の三角形を LMB  でクリックすると、折りたたみ/展開します。

各パネルについての更なる詳細は、[Panels](#) のリファレンスを参照するか (訳注: 2.6用の記事はまだありません)、またはマニュアルの該当セクションを調べてください。

Buttons and Controls

(ボタンと操作項目)


ボタンなどの操作項目は、Blenderインターフェースのほぼすべての [ウインドウ](#) で見つかります。さまざまなタイプの操作項目について以下に説明します。

Operation Buttons

(実行ボタン)



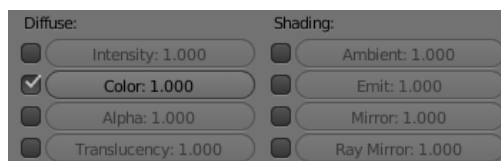
実行ボタン

LMB  でクリックしたときに、処理を実行するボタンです。デフォルトの Blender スキームでは、灰色の見た目で見分けられます。

これらのボタンの上で CtrlC を押すと、その python コマンドがクリップボードにコピーされ、python コンソールやテキストエディタでスクリプトを書くときに使用できます。

Toggle Buttons

(トグルボタン)



トグルボタン

トグルボタンはチェックボックスで構成されています。このタイプのボタンをクリックすると状態をトグル切り替えますが、命令を実行することはありません。場合によっては、ボタンにプロパティの影響を制御する数値ボタンがとりつけられています。

Radio Buttons

(ラジオボタン)

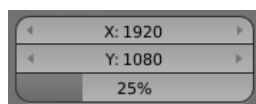


ラジオボタン

ラジオボタンを使用するのは "相互に排他的な" オプションの、数の限られた選択肢から選択するときです。





Number Buttons

(数値ボタン)



数値ボタン

数値ボタンはほとんどの場合、名前とコロンの続く、値を持ったラベルを持つのが特徴です。数値ボタンの操作は、複数の方法で行えます:

1. 一定間隔で値を変更するには、ボタンの両側にある小さな三角形を LMB  でクリックします。
2. 値を大幅に変更するには、LMB  を押したまま左右にマウスをドラッグします。Ctrl を押したままにしてから LMB  を押し続けると、値が一定間隔で変更されます。代わりに ⇧ Shift を押したままにすると、値をより細かく制御できます。
3. ↵ Enter か LMB  で、値を手入力することができます。

値を手入力するとき、⌘ Home または → End を押すと、範囲の先頭または末尾にカーソルを移動します。Escを押すと、編集がキャンセルされます。ボタンの値をコピーすることもでき、ボタンの上にマウスを滞空させて CtrlC 押すことで行えます。同様に、CtrlV で、コピーした値を貼り付けることができます。

Expressions

(式)

6の代わりに、 $3*2$ や $5/10+3$ のような数式を入力することもできます。pi (3.142) のような定数や、sqrt (2) (2の平方根) のような関数も使用することができます。

これらの式は Python によって評価されます。すべての利用可能な数学表現は以下を参照してください: [math module reference](#)

Units

(単位)

式と同様に、数字に単位を混在させることができます。これが動作するためには、シーンの設定で単位を設定する必要があります (メートル法、又は英

単位系)。

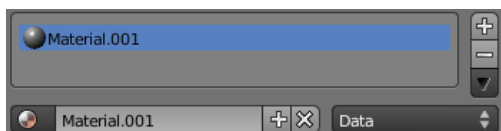
有効な単位の含め方は...

- 1cm
- 1m 3mm
- 1m, 3mm
- 2ft
- 3ft/0.5km
- 2.2mm + 5' / 3" - 2yards

コンマは省略可能であることを覚えておきましょう。同時に表示できる単位は一つだけですが、記述にはメートル法と英単位系を混在できるように注目してください。

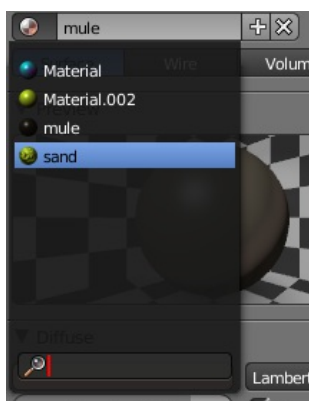
Menu Buttons

(メニューボタン)





データブロックのリンクボタン




メニューボタンは、動的に作成されるリスト上の項目を扱うのに使ってください。メニューボタンは、主にデータブロックを相互にリンクするために使用されます。データブロックとは、メッシュ、オブジェクト、マテリアル、テクスチャなどのような項目のことです。マテリアルをオブジェクトにリンクすると、選択状態のオブジェクトにそのマテリアルを割り当てます。



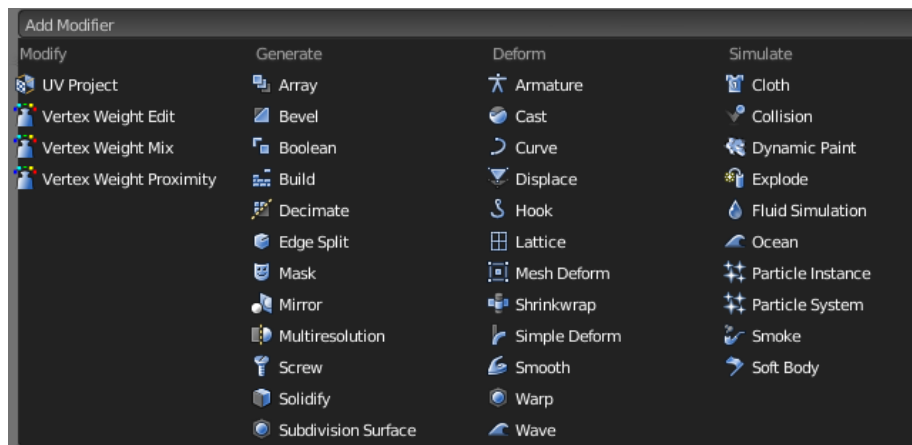
データブロックのリンクメニューと検索

1. 最初のボタン(データブロックの種類アイコン)が開くメニューでは、LMB  クリックして選んだデータブロックをリンクできます。このリストの下部には検索ボックスがあります。
2. 2番目のボタンには、リンクされたデータブロックの名前が表示され、LMB  クリックすると編集できます。
3. "+" ボタンは現在のデータブロックを複製し、それを適用します。
4. "X" ボタンはリンクを消去します。

適用されたデータブロックのリストが表示されることがあります(たとえば、オブジェクトで使われたマテリアルのリストなど)。上記の **データブロックのリンクボタン** を参照してください。

1. データブロックを選択するには、その上で LMB  クリックします。
2. 新しいセクションを追加するには(例えば、マテリアル、パーティクルシステムなど)、リストの右側にある "+" ボタンを LMB  クリックします。
3. セクションを削除するには、リストの右側の "-" を LMB  クリックします。

メニューボタンブロックのもう一つのタイプは、オプションの集まりを持つ静的なリストを表示します。たとえば Add Modifier ボタンは、使用可能なすべてのモディファイアをメニューに生成します。



モディファイアのオプション

リンクの解除されたオブジェクト

リンクの解除されたデータは、**Blenderを終了しない限り失われません**。これは強力なアンドゥ機能です。オブジェクトを削除すると、そこに割り当てられたマテリアルのリンクは解除されますが、まだそこにあります! Blender の終了後もデータを残したい場合は、別のオブジェクトに再リンクするか、またはそのマテリアルに "Fake User" を与える必要があります (アウトライナーのデータブロックビュー内の、該当のデータブロックでそのオプションをクリックします)。

[Fake Userについてもっと詳しく](#) »

Color Selector Controls

(色選択コントロール)

Blender では 4 種類のカラーピッカーを選べます

円形 (デフォルト)、正方形 (HS + V)、正方形 (SV + H) および 正方形 (HV + S)

カラーピッカーの種類の選び方について、詳しくは [システム](#) 設定のページをご覧ください。

どのカラーピッカーにも共通して、値の表示用の RGB、HSV、そして Hex オプションがあります。操作によっては、アルファの制御用スライダーがカラーピッカーの下部に追加されます。

Blender は RGB や HSV 用の色を表すのに、浮動小数点数を使います。Hex 値は HTMLカラーの表現と同じ方法で表現されます。

なお、Blender はデフォルトでガンマを補正します。Blender でガンマ補正を無効にする方法について詳しくは、[色調管理と露出](#) ページをご覧ください。



図2 - カラーピッカー - 円形

Circle (円形) (デフォルト)

中央から端までの色の全域が常に表示されます。中央は混合色です。

輝度 (brightness) は右の上下に伸びるバーで調整できます。

アルファを利用可能な操作では、別のスライダーがカラーピッカーの下部に追加されます。

図2 - カラーピッカー - 円形 をご覧ください。

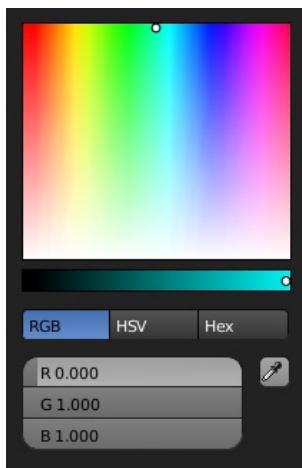


図3 - カラーピッカー
正方形 (HS + V)

Square (HS + V) (正方形 (HS+V))

Hue (色相)、Saturation (彩度) プラス Value (値) → 全色域が常に表示されます。
 スライダーを左に動かすと、カラーピッカーの正方形で選ばれた基本色から輝度が差し引かれます。
 アルファを利用可能な操作では、別のスライダーがカラーピッカーの下部に追加されます。
 カラーピッカー 正方形 (HS + V) をご覧ください。

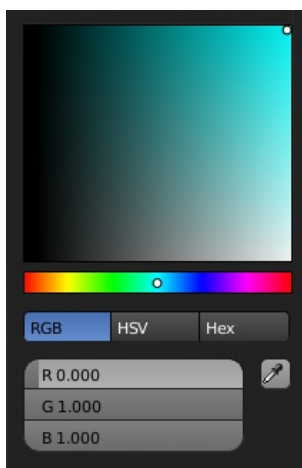


図4 - カラーピッカー
正方形 (SV + H)

Square (SV + H)

Saturation (彩度)、Value (値) プラス Hue (色相) → カラーピッカー中央にあるバーに、全色域が常に表示されます。
 色は、ピッカー中央のカラーバーで選ばれた基本色の輝度の範囲を使って調整されます。
 アルファを利用可能な操作では、別のスライダーがカラーピッカーの下部に追加されます。
 図4 - カラーピッカー 正方形 (SV + H) をご覧ください。



図5 - カラーピッカー
正方形 (HV + S)

Square (HV + S)


Hue (色相)、Value (値)、Saturation (彩度) → カラーピッカーの正方形に、全色域が常に表示されます。
 スライダーを左に動かすと、カラーピッカーの正方形で選ばれた基本色に輝度が足されます。
 アルファを利用可能な操作では、別のスライダーがカラーピッカーの下部に追加されます。
 図5 - カラーピッカー - 正方形 (HV + S)

- 全体的な輝度を変えるには Mouse wheel を使います。
- カラー-slider はデフォルト値を持ちません。前回の変更を行う前の値が代わりに使われます。
- ← Backspace は色をリセットしてデフォルトの色にします。

Eye Dropper

(スポイト)

スポイトを使用すると、Blender ウィンドウ内の任意の場所から色をサンプルすることができます。どのカラーピッカーからも、カラープロパティ上にマウスを乗せて E を押すと呼び出せます。

スポイトツールを LMB  クリックしてドラッグすると、ドラッグ中に通過した色が混合され、ノイズの多い画像でもサンプリングしやすくなります。スペースバー はリセットをし、再度始めから色を混合します。

Cascade Buttons

(カスケードボタン)

ボタンによっては実質的に追加のボタンを表示することがあります。例えば Ramps パネルには Ramp (カラーランプ/色傾斜) と呼ばれるカスケードボタンがあり、カラーバンドを扱う追加的なボタンが現れます。カラーバンド使用前と カラーバンド使用後の図を参照してください。



Color Ramps (カラーランプ)

カラーランプを使うと、カラーストップに基づく色の領域を指定できます。

カラーストップは、選択色が正確にどこにあるべきかを示す印のようなものです。

ランプに追加されたカラーストップ間には、色補間と選んだ補間方式の結果が現れます。カラーランプで使えるオプションには次のものがあります:

Add (追加) (ボタン)

クリックすると、カスタムのウェイトペイント図にカラーストップが追加されます。

カラーストップは最後に選択したカラーストップから隣のカラーストップに向けて左から右に追加され、両者の中点に置かれます。

Delete (削除) (ボタン)

選択したカラーストップをリストから削除します。

'F' (ボタン)

カラーバンドを反転し、ウェイトペイント域の値を反転します。

数値欄

カスタムのウェイトペイント域にカラーストップを追加すると、カラーストップは通し番号 (index) を受け取ります。この欄は追加された通し番号を表示し (反対側のストップまで矢印をクリックすることで) リストからカラーストップを選べます。選んだカラーストップは点線で表示されます。

補間のオプション

各カラーストップに対して色の補間計算を 4 種類から選べます。

次から選べます:

B-Spline (Bスプライン)

カラーストップに B-Spline 補間を使います。

Cardinal (カーディナル)

カラーストップに Cardinal 補間を使います。

Linear (リニア)

カラーストップに Linear 補間を使います。

Ease (イーズ)

カラーストップに Ease 補間を使います。

Constant (一定)

カラーストップに Constant 補間を使います。

Position(位置)

このスライダーは選択したカラーストップの位置を調整します。

Color Bar(カラーバー)

選択したカラーストップ用の色とアルファを指定するカラーピッカーを開きます。

アルファが使われている色なら、カラーバーが二つにわかれ、左側に基本色、右側にアルファ値をともなった色を表示します。

Blender Internationalization

バージョン2.60から、Blenderはインターフェイスとツールチップのための、国際的なフォントと幅広い言語オプションをサポートしています。これを有効にするには、Systemタブで、User Preferencesを開き、右下隅のInternational Fontsのオプションをトグル切り替えます。

こうすると3つの新しい設定が表示されます:



Enabling international fonts in the User Preferences window.

Language

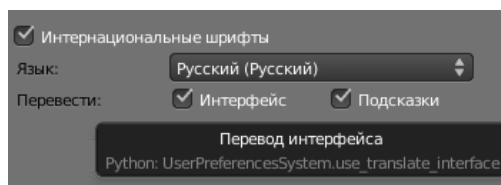
希望する言語を選択することができるドロップダウンメニュー。

Interface

ユーザーインターフェイス自体を訳します (例えば、操作項目やメニュー)。

Tooltips

ツールチップを訳します。



Blender with the Russian language enabled for the Interface and Tooltips.

インタフェースかツールチップ(あるいはその両方)を選択すると、Blenderは画面をリフレッシュし、新しい言語を表示します。いくつかの言語での翻訳はまだ完了していないことに注意してください。各翻訳の進行状況は、ドロップダウンメニューに表示されます。



Tip

チュートリアル的大部分はユーザーインターフェイスが英語のもので行われているので、ユーザーインターフェイスは英語のままにし、ツールチップだけを訳文表示すると便利かもしれません。

Quick Rendering

(クイックレンダリング)

What is rendering?

(レンダリングって何?)

レンダリングは、2D画像を作成するプロセスです。Blenderは、モデルや、マテリアル、テクスチャ、ライティング、合成のすべてを考慮に入れて、このイメージを作成します。

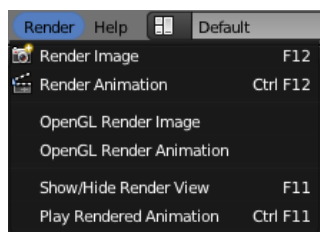
- Blender の内部レンダリングエンジンには、主に2つの種類があります。一方はフルレンダリングのためのもので、他方は OpenGLレンダリングのためのものです。このページでは、画像のレンダリングについての基本的な知識を説明します。Blender の内部に組み込まれている、Blender Internalと呼ばれるフルレンダリングエンジンに関してもっと深く知りたいのであれば、[内蔵レンダラを使ったレンダリング](#)についてのセクションを調べてみましょう。
- また、この wikiマニュアルには、新しい [Cycles](#)レンダリングエンジンという、バージョン2.61から Blender に組み込まれたレンダー専用のセクションもあります。

Rendering an image using Full Render - Blender Internal

(フルレンダリングを使った画像のレンダリング - Blender Internal)

モード: 全モード

ホットキー: F12



情報 (Info) ウィンドウのヘッダー

Blender Internal を使用したフルレンダリングは、次のいずれかのオプションで開始できます:

- F12キーを押す
- プロパティウィンドウ » Render(レンダー)コンテキスト » Render(レンダー)パネル に移動し、Render(レンダー) ボタンを押す
- Info (情報) ウィンドウ のヘッダから Render(レンダー) » Render Image (画像をレンダリング) と進む(参照: [情報 \(Info\) ウィンドウのヘッダー 画像](#))
- Spaceキーを押して、Blenderの検索を使用し、Renderと打ち込んで、Render をクリックする。

レンダリングを中止か終了するには、Esc を押します。

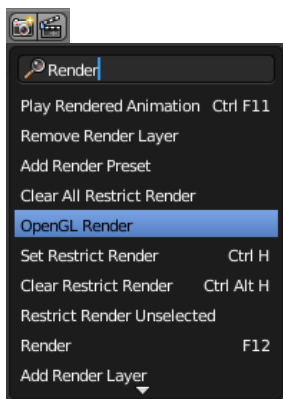
Rendering an image using OpenGL Render

(OpenGL レンダーを使った画像のレンダリング)

モード: 全モード

ホットキー: 未定義。[Keymap\(キーマップ\)](#) » で独自のホットキーを追加できます。

OpenGL レンダーは、次のオプションのいずれかで開始できます:



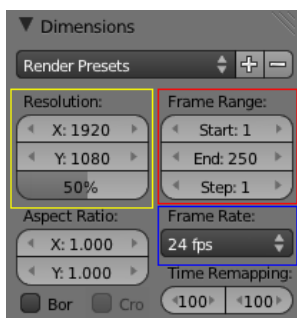
機能の検索

- 3Dウィンドウのヘッダーで *OpenGL Render Active Viewport (現在の視点を OpenGL レンダリング)* をクリックする。カメラの絵がついた小さなボタンで、カチンコの絵がついたボタンといっしょにあります
- Info (情報) ウィンドウのヘッダから Render(レンダー) » OpenGL Render Image (OpenGL 画像レンダリング) と進む(参照: [情報 \(Info\) ウィンドウのヘッダー 画像](#))
- Spaceキーを押してBlenderの検索を使用し、Renderと打ち込んで、OpenGL Render をクリックする。

レンダリングを中止か終了するには、Escを押します。

Adjusting the resolution

(解像度を調整)

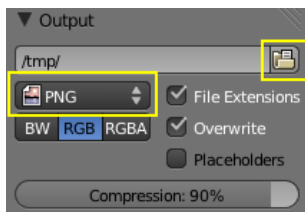


Dimensions(寸法)パネル

Render(レンダー)コンテキストの Dimensions(寸法)パネルでは、解像度を変更することができます。Blenderのデフォルトのインストールでは、**1920 x 1080の50%**に最初に設定されており、その結果 **960 x 540**の画像になっています (Dimensionsパネルのイメージ内の、黄色の強調表示)。高解像度かつ高いパーセンテージに調整すると、より詳細に表示されますが、レンダリング時間も長くなります。

Output format and output file

(出力形式と出力ファイル)

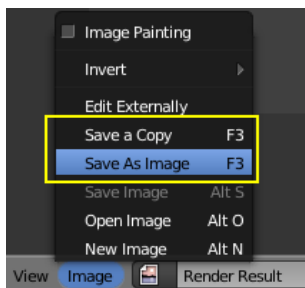


Output(出力)パネル

レンダリング画像やアニメーションの出力形式や出力場所を選択することもできます。デフォルトでは、それらは絶対パスを使用して、一時フォルダ (/tmp)に保存されます。[ファイルの設定の章](#)にあるような命令を使用してファイルパスをセットアップすることもできますが、Output(出力)パネルのフォルダアイコンをクリックすることで別のフォルダに変更することもできます。また、メニューボタンからあなたの作品の画像や動画の形式の種類を選択することもできます。

Saving your image

(画像を保存する)



「画像を別名保存」ダイアログ

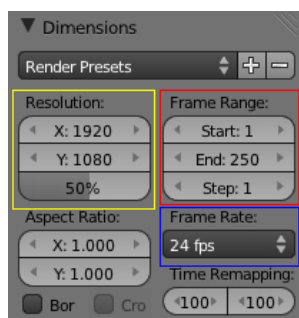
Blenderは画像を自動的に保存してくれません。画像を保存するには、F3を押すか、UV/Image エディタウィンドウのヘッダーの Image(画像)メニューから Save As Image(画像を保存)をクリックします。このアクションで、Blender内部のファイルブラウザが開き、レンダリングしたものを配置するフォルダを検索できます。

Rendering an animation using Full Render - Blender Internal

(フルレンダリングを使ったアニメーションのレンダリング - Blender Internal)

モード: 全モード

ホットキー: CtrlF12



Dimensions (寸法) パネル

アニメーションをレンダリングするのは簡単で、Output (出力) パネルの Frame Range (フレーム範囲) (図「Dimensions (寸法) パネル」で赤で強調表示された部分) を使用して、レンダリングされるアニメーションの **フレーム枚数** を定義します。**間隔** は **フレーム毎秒 (FPS)** によって定義され、Frame Rate (フレームレート) ドロップダウンリスト (図「Dimensions (寸法) パネル」の青で強調表示された部分) で定義します。標準では **24 FPS** かつ **250** フレームに設定されています。

これらの数字を理解するための簡単な例:

- このパネルでは、アニメーションをフレーム **1** で開始してフレーム **250** で終了させることを示しています。FPS の設定は **24** に設定されているので、標準の Blender のインストールでは、だいたい **10** 秒のアニメーションになります ($250 / 24 = 10.41$ 秒)。

Blender Internal エンジンで フルレンダリングを使用したアニメーションのレンダリングは、次のいずれのオプションで行うことができます:

- CtrlF12 を押す
- Properties (プロパティ) ウィンドウ » Render (レンダー) コンテキスト » Render (レンダー) パネルに移動し、Animation (アニメーション) ボタンを押す
- Info (情報) ウィンドウのヘッダから Render (レンダー) » Render animation (アニメーションレンダリング) と進む (参照: *Header of the Info Window* 画像)

アニメーションのレンダリングを中止か終了するには、Esc を押します。

Rendering an animation using OpenGL Render

(OpenGL Render を使用したアニメーションレンダリング)

モード: 全モード

ホットキー: 未定義。 [Keymap \(キーマップ\)](#) » で独自のホットキーを追加できます。

OpenGL Render を使用したアニメーションのレンダリングは、次のいずれかのオプションで行うことができます:



- 3Dビューのヘッダーで、スレートの絵がついた小さなボタンをクリックする (カメラの絵がついた小さな画像の隣にある)
- Info Window のヘッダから Render (レンダー) » OpenGL Render animation (OpenGL アニメーションレンダリング) と進む (参照: *情報 (Info) ウィンドウのヘッダー* 画像)

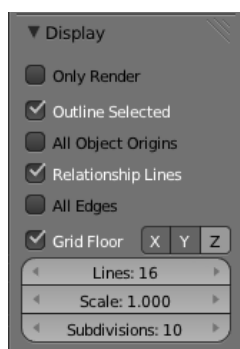
アニメーションのレンダリングを中止か終了するには、Esc を押します。

Showing Only Rendered Objects

(レンダリングされるオブジェクトのみを表示)

モード: 全モード

ホットキー: 未定義。 [Keymap \(キーマップ\)](#) » で独自のホットキーを追加できます。



プロパティシェルフ - Display (表示) パネル

レンダリング時には (Full または OpenGL のいずれも)、シーン内の一部のオブジェクトがレンダリングされない場合があります。その理由は、型のため (ボーン、エンブレ、カメラなど)、無効であるかまたは目に見えるジオメトリを持っていないため (全く頂点がないメッシュ、伸び幅のない曲線、等)、または単にそれらがレンダリングしないように設定されているため、のいずれかです。

Blenderは、レンダリングされるオブジェクトだけをシーン内に表示するオプションを持っています。

このオプションにアクセスするには、3Dビューにマウスをポイントし (それに焦点を当てて)、ショートカット **N** を使用するか右上の **+** 記号をクリックして、**プロパティシェルフ** を表示させます。オプションをスクロールさせながら探していくと、**Display** (表示) パネルが見つかるはずですが、このオプションは、オブジェクトが 3Dビューでどのように表示されるかを制御するためのものです。

さっそく **Only Render** (レンダリングするもののみ) オプションを有効化しましょう - これでもう、レンダリングされるオブジェクトのみが表示されます (図「プロパティシェルフ - Display (表示) パネル」)。このオプションは、OpenGL のレンダリングを使用して画像を生成するときにも作用します。選択表示のための他のすべてのオプションを無効にすることを忘れずに。

The purposes of OpenGL Rendering

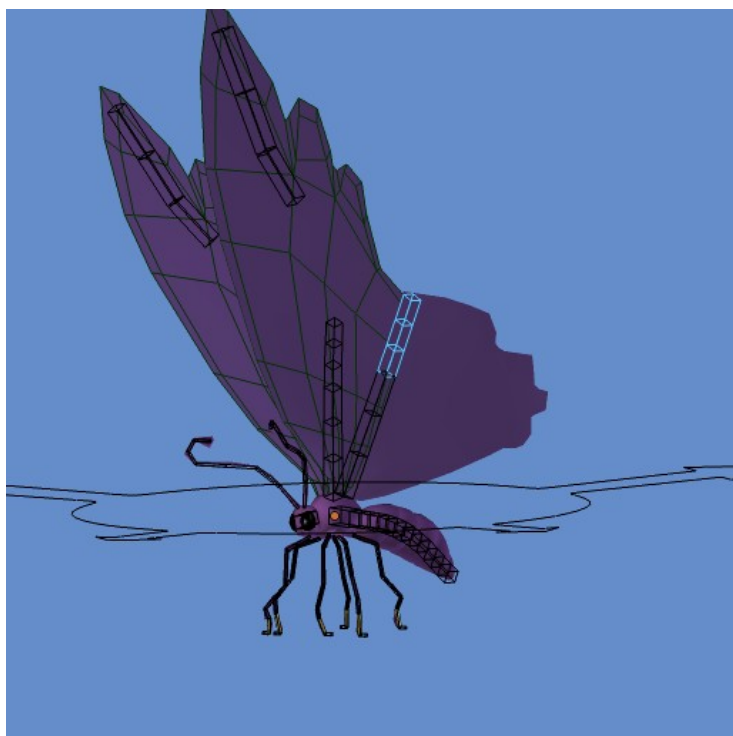
(OpenGLレンダリングの用途)

OpenGLレンダリングは、現在のビューポートのドラフト品質のレンダリングを与えることで、アニメーターが **animatic** (オブジェクトの動き、互いの角度、などのようなもの) をすばやく点検できるようにしてくれます。

OpenGL だけを使用してレンダリングされるので、見た目は 3Dビューポートに表示されるものと同じくらいでしかないとはいえ、生成がとても高速です。

これを使うと、たとえシーンの複雑さのためにリアルタイムで行えないような時でも (すなわち: **AltA** を押したときでは結果の **FPS** が低すぎて、アニメーションの感触がうまくつかめないような時でも)、アニメーターはアニメーションを流動的に再生しながらプレビューすることができます。

これが OpenGLレンダリングイメージの例です:



そして、Blender内部レンダリングエンジンを使用したフルレンダリングがこちらです:



OpenGLは、画像とアニメーションの両方のレンダリングに使用できます。そして上で説明したのと同じ手順を使用して、寸法を変更することができます。また、通常のレンダリングと同様に、Escを押して中止することができます。

ミスからの回復

Blenderは、あなたの作業工程を失うことがないようにする多くのオプションと機能を備えています。これには Undo History, Auto Save および Save on Quit などの機能が含まれます。これらの機能はすべて、デフォルトで有効にされています。Undo History はあなたの行動をリスト内に保存し、Auto Save はあなたの作品をバージョンごとに保存する数々の機能を担い、Save on Quit はBlenderを閉じるときのファイルを自動保存してくれます。

アンドゥ

Mode: All modes

Hotkey: CtrlZ

ほとんどのプログラムのように、あなたの最後の操作をアンドゥする場合は、CtrlZを押します。

リドゥ

Mode: All modes

Hotkey: ⇧ ShiftCtrlZ

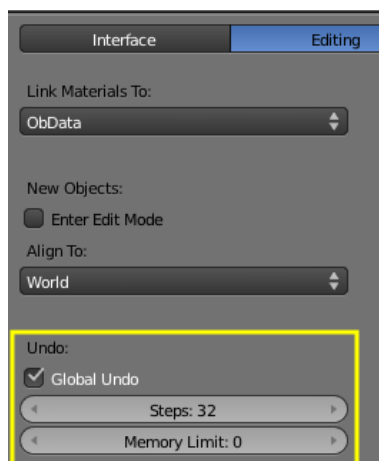
最後に取り消したアクションを再実行するには、⇧ ShiftCtrlZを押します。

アンドゥの履歴

Mode: All modes

Hotkey: CtrlAltZ

Blenderは最後から32までの行動の記録を保持し、CtrlAltZ押すことで表示できます。任意のアクションをクリックすると、後続のアクションをすべてアンドゥして、指定した状況に戻ります。2つの別々の履歴があることに注意してください: ひとつは エディット mode, and aモードに専用のもので、そして Global Undo リストはそれ以外のすべてのモードのためのものです。あなたの行動の記録を維持してくれますが、その代償としてシステムメモリがかさみます。



Undo options

アンドゥを有効または無効にするには、User Preferences ウィンドウに移動して Editing タブをクリックします。このセクションで設定することができるのは:

Steps

この数値フィールドは、ステップまたはアクションをどれだけの数保存するかを示します。デフォルト値の **32** では、実行したうちの最後から三十二までのアクションを元に戻すことができます。

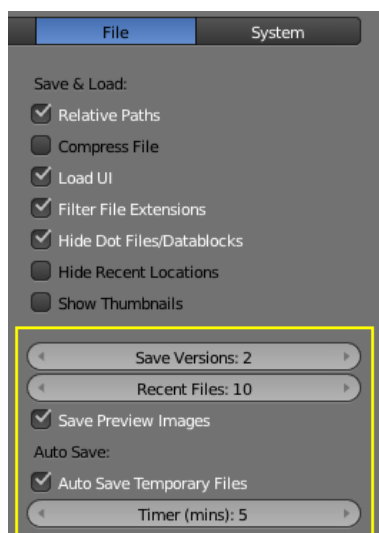
Memory Limit

この数値フィールドでは、アンドゥシステムに使用を許可するメモリをメガバイト単位で定義することができます。デフォルト値の **0** は制限がないことを示します。

Global Undo

Edit Mode でないとき、Blenderがアクションを保存することができます。たとえば、オブジェクトを複製したり、パネルの設定を変更したり、モードを切り替えるときです。

Save and Auto Save



Auto Save options

コンピュータのクラッシュ、停電、あるいは単に保存を忘れることで、あなたの作業が損失または破壊されるおそれがあります。この問題が発生する可能性を減らすために、Blenderの Autosave 機能を利用することができます。User Preferencesウィンドウの File タブでは、Blenderが提供する、前のバージョンへの作業の退縮を、2つの方法で設定することができます。

Save Versions

手動でファイルを保存するとき、このオプションは、手動で保存されたバージョンを指定された数だけ、現在の作業ディレクトリのファイルに保持するよう、Blenderに指示します。これらのファイルは拡張子を持つことになります: `.blend1`, `.blend2`, という具合にバージョンの数をあなたが指定した数まで増加させていきます。古いファイルは、より高い数字で名前が付けられます。例えば、デフォルト設定である **2**で、あなたがファイルを3つのバージョンで持っているとする: `*.blend` (最後のセーブ), `*.blend1` (最後から2番目のセーブ) and `*.blend2` (最後から3番目のセーブ)。

Auto Save Temporary Files

このチェックボックスをオンにすると、進行中の作業のバックアップコピーを一時ディレクトリに *自動的に* 保存するようにBlender指示します (その場所についてはUser Preferencesウィンドウの File パネルを参照)。また、これによりそれぞれの自動保存の間隔を分単位で指定するタイマー (分) 制御が可能になります。自動保存されたファイルは、乱数を使用して名前が付けられ、`.blend` 拡張子を持ちます。

Recovering Auto Saves

Recover Auto Save

File » Recover Auto Save... で自動保存ファイルを開くことができます。自動保存されたバージョンをロードする際には、で、作業ディレクトリの現在のファイルに、通常の `.blend` ファイルとして上書きすることもできます。

Recover Last Session

Blenderを終了するときに Temp ディレクトリに保存される`quit.blend` は、File » Recover Last Session で開きます。リポートすると、Temp ディレクトリ内のファイルが削除されることに注意してください。

Important Note

自動保存されたファイルを復元するとき、最後のAuto Save が実行されて以降にあなたが行ったあらゆる変更は失われます。

各プロジェクトに **ひとつ** だけ自動保存ファイルが存在することになります (すなわち、Blenderは古いバージョンを保持しません – したがって、このツールを使って、数分前より過去にさかのぼることはできなくなります)。

その他のオプション

Recent Files

この設定では、File » Open Recent サブメニューにいくつかのファイルを一覧表示するかを制御します。

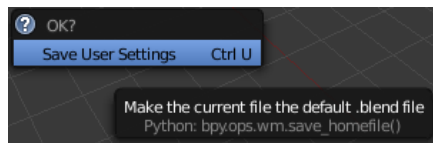
Save Preview Images

ファイルブラウザ ウィンドウ内のイメージや素材のプレビューがオンデマンドで作成されます。これらのプレビューを `.blend` ファイルに保存するには、このオプションを有効にしてください (代償として `.blend` ファイルのサイズは増えます)。

デフォルトのシーンを設定する

あなたが Blender を起動するか、メニューのエントリの File » New または CtrlN で新しいプロジェクトを開始すると、新しいシーンが User Preferences (ユーザープリファレンス) に保存されているデフォルトのシーンから作成されます。

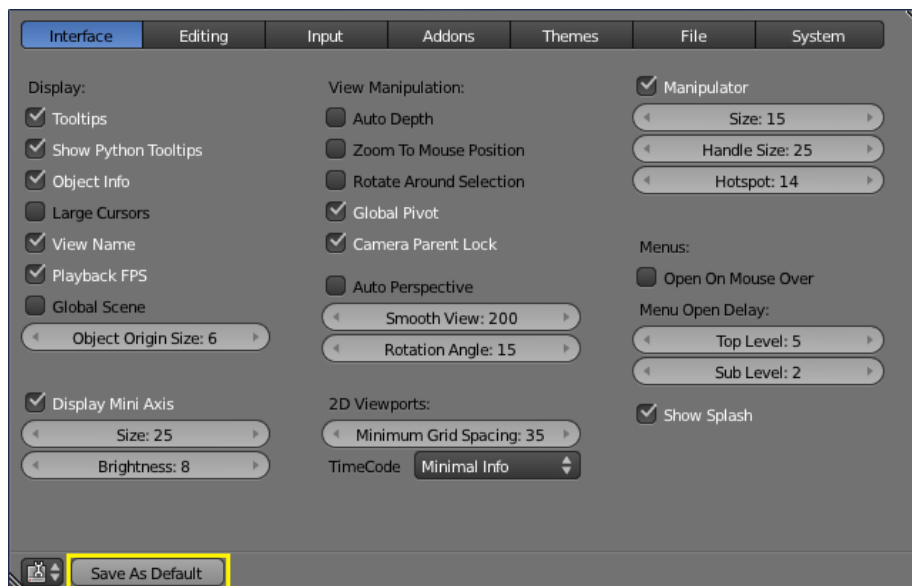
ホットキー デフォルトのシーンを変更するには、現在のシーンに必要なすべての変更を行い CtrlU キーを押します。



ポップアップの Save User Settings を LMB (左マウスクリック) するか、または ↵ Enter キーを押します。キャンセルは Esc キーを押します。

メニューから デフォルトのシーンを変更するには、現在のシーンに必要なすべての変更を行い File » User Preferences を LMB (左マウスクリック)。User Preferences 画面が表示されます。

Save As Default ボタンを LMB (左マウスクリック) で変更を保存します。デフォルトのシーンを変更しない場合は、User Preferences ウィンドウを (Default Scene ボタンをクリックせずに) 閉じてください。



結果として

現在のシーン、すべてのオブジェクト、および設定は、ユーザ設定 で保存されます。

工場出荷時設定へのデフォルトのシーンの復元

工場出荷時の設定にデフォルトのシーンを復元するには File » Load Factory Settings と LMB (左マウスクリック) します。これは、すべての User Preferences を元の工場出荷時設定に復元します。

スクリーンショット

チームでの作業やラピッドプロトタイピングを容易にするために、ウィンドウやBlenderのウィンドウセットアップ全体の写真をすばやく撮れたらと思うことがあるかもしれません。

File » Screenshot Subwindow は現在メニューオプションでは使用できなくなっています。CtrlF3 でアクティブウィンドウの写真を撮るという表示が出て、そして"保存" でFile Browser を開けば、ファイルの場所と名前を指定することができます。この機能はBlender 2.55ベータ版を一貫してクラッシュさせます。この機能を無効にしたり、GIMPなど、画面キャプチャ機能を持つほとんどのイメージエディタによって回避する必要があります。

File » Screenshot All はすべて、メニューオプションでは使用できなくなっています。◊ ShiftCtrlF3 では無効になっていると表示されます。(確認を求めてきます)

OSの画面キャプチャ

クリップボードに画面をキャプチャするためにはオペレーティングシステムを使用してください。その後、クリップボードから画像エディタにイメージを貼り付けることができます。

Windowsの画面キャプチャ

クリップボードに画面をキャプチャするには、CtrlPrtscr を押してください。

Macintoshのスクリーンキャプチャ

⌘ Cmd◊ Shift3 を押すと画面キャプチャをデスクトップ上のファイルに作成する。

Ctrl⌘ Cmd◊ Shift3 を押すとクリップボードに画面をキャプチャする。

⌘ Cmd◊ Shift4 を押すと画面の領域キャプチャをデスクトップ上のファイルに作成する。

Ctrl⌘ Cmd◊ Shift4 を押すとクリップボードに画面の領域をキャプチャする。

GNU/Linux Screen Capture

Ubuntu などのユーザビリティベースのシステムでは、Prtscr を押すと画面をファイルにキャプチャすることができます。

あなたのアクションをビデオで撮るには、Ubuntu の公式リポジトリから入手できるgtk - recordmydesktop アプリケーションを使用できます。

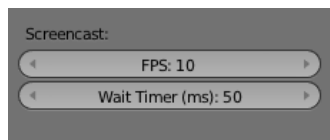
Blender Screencasts

(Blenderの画面を録画する (スクリーンキャスト))

モード: すべてのモード

ホットキー: AltF3

ショートカット AltF3 でスクリーンキャスト機能を開始します。スクリーンキャストは画面操作を動画や連番画像に記録し続けます。出力形式や出力場所は [レンダー \(Render\) コンテキスト](#) ウィンドウの [出力パネル](#) で指定します。スクリーンキャストはデフォルトでは 50ミリ秒ごとに PNG 画像 をキャプチャーし、/tmp フォルダに保存します。動画に記録したい場合には 出力 (Output) パネルで、お使いのシステムが対応している 動画ファイル形式 を選びます。お使いのシステムのサポートするビデオコーデックがわからないときは、AVI JPEG を選んでください。



ユーザー設定エディタにあるオプション

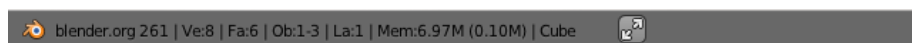
動画の FPS または 連番画像の撮影間隔は [ユーザー設定](#) ウィンドウの [システムパネル](#) で指定します。(図「ユーザー設定エディタにあるオプション」をご覧ください。)

音声対応

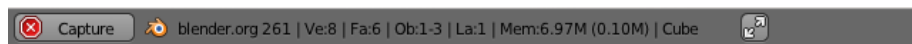
Blender のスクリーンキャストは録音に対応していないので、他のソフトウェアを使って自分で録音する必要があるでしょう。例えば [Audacity](#) を Blender と組み合わせて使います。

Blender のスクリーンキャストを開始すると 情報 (Info) ウィンドウ のヘッダーが変わり、録画停止ボタンが表示されます。

下の画像「通常操作時の情報ウィンドウ」は、Blender を普段操作しているときの 情報ウィンドウのヘッダーです。スクリーンキャストモードに入ると、録画停止ボタンが現れます (画像「録画停止ボタンのある情報ウィンドウ」)。



通常操作時の情報ウィンドウ

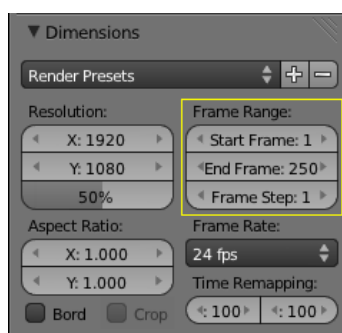


録画停止ボタンのある情報ウィンドウ

(備考: ヘッダー画像は Blender 2.61 のものです)

録画を止める唯一の方法

情報ウィンドウのヘッダーにある停止ボタン以外の方法で、録画を止めることはできません。Esc を押すショートカットは通常のインタフェースの Blender で行った操作には効果がありますが (アニメーションなどを止めたりはできます...)、録画は止められません。



寸法パネルのフレーム範囲

ファイル名の後ろにはフレーム番号が入ります。フレーム番号は [レンダー \(Render\) コンテキスト](#) の寸法 (Dimensions) パネルにある、*開始フレーム*と*終了フレーム*で与えられた範囲内の数字になります (画像「寸法パネルのフレーム範囲」で黄色にハイライトされた部分)。



重要:

寸法パネルのフレーム範囲の終了フレームの設定は、録画を自動的に終了させません。常に、停止ボタンを使って手動で停止する必要があります。

動画の各フレームは作業中のウィンドウの幅と高さを使って、内部的には スクリーンショットと同じ手順で生成されます。動画ファイルへの保存を選ぶと、生成されたフレームがビデオコーデックに渡されます。

警告: コーデックによっては出力の幅や高さ、ビデオの品質に制限のあるものがあります。

- 連番画像として スクリーンキャスト を保存すると、画像は Blender のウィンドウ全体の幅と高さを持ち、画像品質は形式 (JPG、PNG他) と設

定 (JPG形式の「品質」スライダー) で変わります

- 動画形式でスクリーンキャストを保存すると、各フレームがコーデックに渡されます。コーデックによっては、出力される動画が縮小される可能性があります。また、ウィンドウの特定の幅と高さの組み合わせを処理できないコーデックがあります。この場合 スクリーンキャストは開始しようとしてすぐに停止します。解決策として、他のウィンドウサイズや他のコーデックを選んでください。

Blender Window Dimension

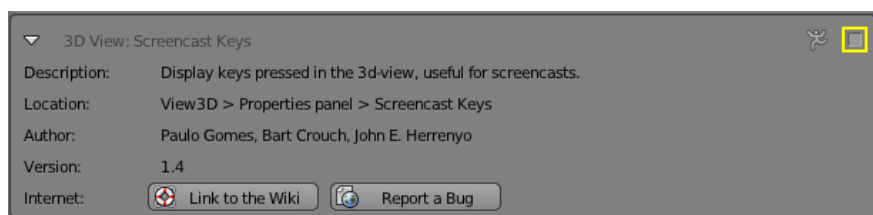
(Blender のウィンドウサイズ)

Blender のスクリーンキャストの出力を標準的な寸法 (NTSC、HD、Full HD など) にして、Blender のウィンドウ寸法と出力動画の寸法を一致させる方法があります。コマンドラインから Blender を起動すると、Blender のウィンドウの幅や高さを制御可能です。コマンドラインからの起動について詳しく学ぶには [コンソールウィンドウ](#) をご覧ください。

Addon: 3D View: Screencast Keys

(アドオン)

コミュニティ発のアドオン 3D View:Screencast Keys は、スクリーンキャストの録画中に入力されたキーやマウスボタン、またはその組み合わせを、3D 画面の左下隅に表示します。このアドオンは デフォルトで Blender に同梱されています。下の画像はこのアドオンのタブを開いたところです。このアドオンを有効にするには、ユーザー設定 を開いて CtrlAltU、アドオン タブに行き、3D Viewのアドオンを探します。チェックボックス(下の画像では黄色にハイライトされています)をクリックして、アドオンを有効にします。

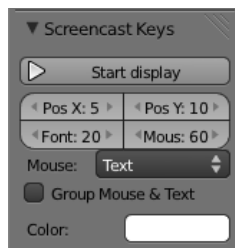


アドオン: 3D View: Screencast Keys

モード: すべてのモード → アドオン有効化

ホットキー: Nでプロパティシェルフを表示 → Screencast Keys タブ

メニュー: View(ビュー) » Properties(プロパティ) → Screencast Keys タブ



プロパティシェルフの Screencast Keys アドオンのタブ

アドオンが有効化されると、プロパティシェルフの最後に Screencast Keys セクションが追加されます。

解説

- **Start display** ボタン: このボタンを押すと、入力したキーやキーの組み合わせが 3Dウィンドウの左下隅にフローティングテキストとして表示されます。同じキーやキーの組み合わせを繰り返すと、その表示の後ろに繰り返した回数 n を示す「xn」が付きま。
- **Stop display** ボタン: は ScreenCast Keys の表示を止めます。
- **PosX**: Screencast テキストの X 座標です。
- **PosY**: Screencast テキストの Y 座標です。
- **Font**: Screencast テキストのフォントサイズです。
- **Mouse**: Screencast のマウスアイコンのサイズです。
- **Mouse display**: Screencast テキストの表示方法を選びます。
- **Text**: 押されたキーやマウスボタンをテキストとして表示します。
- **Icon**: マウスをアイコンとして、キー入力をテキストとして表示します。
- **None**: キー入力の情報のみ表示し、マウスボタンの情報は表示しません。
- **Group Mouse & Text Check box**: 有効にすると、Screencast テキストの周りに枠をつけて見やすくします。
- **Color**: Screencast テキストの色を選びます。

💡 新しいコミュニティアドオン

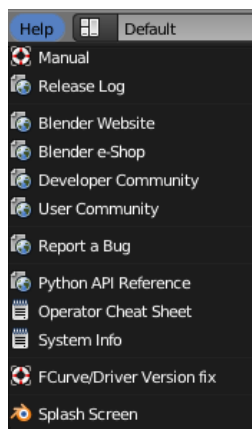
Blender 2.5/2.6 用のアドオンで、マウスで選んだ好きな部分のスクリーンショットをとり、直接 [Pasteall](#) にアップロードするものがあります。このアドオンには現在開発に関するページがありませんが、完成すればここにリンクします。

Help system

Blender は、ビルトインおよびWebベースのヘルプオプションを呼び出せるようになっています。Info ウィンドウのヘッダーにあるHelp メニューを介してアクセスできます。

Web Access

一部の方式のヘルプは、Webブラウザを起動して、ブレンダー財団のWebサーバにアクセスします。これを行うためには、あなたのオペレーティングシステムにおける、デフォルトのWebブラウザを設定しておく必要があります。あなたがダイヤルアップ接続を使用している場合で、使用可能なアクティブなインターネット接続がない場合は、起動時に自動的にダイヤルアウトするようにWebブラウザを設定する必要があります。



Help menu

Help Menu

Web-based Help options

Manual

[Wiki Manual](#)の目次ページのメインテーブルに連れて来ます。

Release Log

利用中のバージョンの[release notes](#)のWebページに連れて来ます。

Blender Website

[Blender](#)のホームページに連れて来ます。

Blender e-Shop

[Blender Store](#)のWebページに連れて来ます。

Developer Community

blender.orgの[Get Involved](#)ページに連れて来ます。この導入ページにはブレンダーの、ソフトウェアの開発、バグの追跡、パッチやスクリプト、教育やトレーニング、マニュアルの開発と機能性の研究などがあります。

User Community

Blenderに貢献している[User Community](#)のウェブサイトのリスト。

Report a Bug

[Blender Bug Tracker](#)のページに連れて来ます。

Python API Reference

このヘルプページで説明している[Python application programming interface \(API\)](#)というのはBlenderとPythonが互いに通信するために使用するものです。

Internal Help options

Operator Cheat Sheet

利用可能なPythonの演算子を一覧表にして、`operatorList.txt`ファイルとして生成します。このリストは、テキストエディタを介してアクセスできます。Browse Text to be linked メニューリストをクリックして (Text Editor ウィンドウのヘッダ上の New ボタンの隣)、利用可能なTextBlockを表示させてください。

System Info

System Info は、システムとブレンダーのさまざまなキーのプロパティを一覧表にして、`system-info.txt`ファイルとして生成します。問題の診断に役立ちます。このリストは、上記の Operator Cheat Sheet にリストされているものと同じ命令を介してアクセス可能です。


FCurve/Driver fix

shapekey driversドライバの修正。

Splash Screen

ここでは、パッケージとバージョンを識別する、スプラッシュスクリーンのイメージが表示されます。

Open User Preferences

Blenderの User Preferences editor を開くには、File » User Preferences と進むか、CtrlAltUを押しましょう。Macユーザーは ⌘ Cmd,を押すことで開きます。また、どのウィンドウにもPreferences editor をロードさせることができ、Window type selection メニューから  User Preferences を選択することで行えます。



このエディタでは、Blenderがどのように機能するかを設定できます。使用可能なオプションは7つのタブに分類され、ウィンドウの上部でアクセス可能です。オプションは次のとおりです: *Interface*, *Editing*, *Input*, *Add-Ons*, *Themes*, *File* and *System*.

Configure

さて、User Preferences editorが開いたら、あなたの好みに合わせてBlenderを設定することができます。変更したいものを次のリストから選択しましょう:

[インターフェイス](#) • [編集](#) • [入力](#) • [アドオン](#) • [テーマ](#) • [ファイル](#) • [システム](#)

Save the new preferences

一度プリファレンスを設定した後は、それらを手動で保存する必要があります。そうしないと、再起動した場合や、新しいシーンを起動した後に、新しい設定が失われてしまいます。Blenderはユーザー設定をそれぞれのシーンと一緒に保存します。こうなっていることで特定のレイアウトや特殊なアドオンを必要とするとき便利なことがあります。

User Preferencesウィンドウのヘッダ(この場合はフッター)で、Save As Defaultをクリックします。これで保存されるのは:

- 新しい設定のすべて。
- 現在開いているシーンを、デフォルトのシーンとして。

またUser Preferences の保存は、いつでも CtrlUを押すことで行えます。

工場出荷時設定をロードする

デフォルトのBlenderの設定を復元するには、2つの方法があります:

1. File » Load Factory Settings と進んで、CtrlU か User Preferences editorでユーザー設定を保存する。
2. コンピュータ上の次の場所から startup.blendファイルを削除する:
 - Linux: `/home/$user/blender/'Version Number'/config/startup.blend` (隠しファイルを表示する必要があります)。
 - Windows 7 and Windows Vista: `C:\Users\%user%\AppData\Roaming\Blender Foundation\Blender\'Version Number'\config\startup.blend`
 - MacOS: `/Users/$user/Library/Application Support/Blender/'Version Number'/config/startup.blend` (隠しファイルを表示する必要があります)。

Blenderのconfigフォルダ内をいじるときには、Blenderの設定ファイルを別のフォルダにコピーしておくことが大事です。万一設定を失った場合に、バックアップコピーを使用してBlenderの設定ファイルを復元することができます。

表示

Tooltips

有効にすると、マウスポインタを操作項目の上のせたときにツールチップが表示されます。このチップでは、ポインタの下に何の機能があるかを説明し、関連するホットキー（もしあれば）とそれを参照するPythonの関数を提示します。

Object Info

3Dビューの左下にアクティブオブジェクトの名前とフレーム番号を表示します。

Large Cursors

利用可能なときに大きいマウスカーソルを使用します。

View Name

3Dウィンドウの左上隅に現在のビューの名前と種類を表示します。For example: User Persp or Top Ortho.

Playback FPS

アニメーションが再生される間、画面のリフレッシュレートを秒間あたりのフレーム数で表示します。ビューポートの四隅のいずれかに表示されません。

Global Scene

現在のシーンが他のすべてのシーンで表示されるように強制します (プロジェクトは複数のシーンで構成されます)。

Object Origin Size

ビューポートでの3Dオブジェクトの中心の直径 (value in pixels from 4 to 10).

ミニ軸を表示

ビューポートの左下にミニ軸を表示します。

Size

ミニ軸のサイズを指定します。

Brightness

ミニ軸の明るさを調整します。

Properties Window

Width Check

Properties windowの幅がこの値以下のとき、ボタンが2列ではなく1列で表示されます。

View manipulation

Auto Depth

マウス下の深さを使用して、ビューのパン/ 回転/ズームの機能を向上させます。

Zoom to Mouse Position

有効にすると、マウスポインタの位置が、2Dウィンドウの中央の代わりに、ズームの焦点になります。ズームインやズームアウトを頻繁にする場合、パンをせずに済みます。

Rotate Around Selection

選択しているオブジェクトが、ビューポートの回転中心になります。

Global Pivot

すべての3Dビューで、回転/スケーリングのピボットを同じにロックします。

Auto Perspective

User Orthographicを使用した後に、自動的に視点をトップ/サイド/フロントビューにする。無効にすると、トップ/サイド/フロントビューは、正投影または遠近法ビューに保たれます (そのビューへの切り替え時にアクティブであった方)。

Smooth View

テンキーでビュー (Top/Side/Front/Camera...)を変更するときのアニメーションにかかる時間の長さ。ゼロに減らすと、アニメーションをとばすことができます。

Rotation Angle

回転させるときのステップサイズの各度で、4 NumPad, 6 NumPad, 8 NumPad, 2 NumPadでの3Dビューの回転のときに使用されます。

2D Viewports

Minimum Grid Spacing

2D (すなわち上部正投影)ビューポートのグリッド線の間のピクセルの最小数。

TimeCode Style

フレーム区切りのタイミングで表示されていないときの、タイムコード表示の形式。この形式は、ミリ秒のフレーム番号の区切り文字として '+' を使用し、必要に応じてタイムコードの左側と右側が切り捨てられます。

Manipulator

3D変形マニピュレータのコンフィギュレーションを可能にするもので、オブジェクトのドラッグ、回転、サイズの変更 (Size, Handle size)に使用されます。

Menus

Open on Mouse Over

このオプションを選択すると、エントリの上にマウスポインタを置くだけで、クリックしなくてもメニューが開くようになります。

Menu Open Delay

Top Level

1/10秒区切りでの、メニューが開く前の遅延時間 (Open on Mouse Overを有効にする必要があります)。

Sub Level

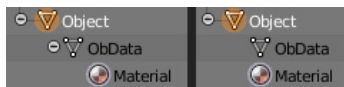
サブメニューについての、上記と同じもの (for example: File » Open Recent)。

Link Materials To



このオプションを正しく理解するには、オブジェクトでのBlenderの動作方法を理解する必要があります。Blenderのほとんどあらゆるものは、データブロックの序列階層で構成されています。データブロックは、特定の情報のコンテナと考えることができます。たとえば、Objectのデータブロックには、オブジェクトの位置に関する情報が含まれていて、一方Object Data (ObData) のデータブロックには、メッシュに関する情報が含まれています。

マテリアルのリンクは、2つの異なる方法で行われます:



A material linked to ObData (left) and Object (right).

ObData

作成されるマテリアルがすべて、ObDataのデータブロックの一部として作成されます。

Object

作成されるマテリアルがすべて、Objectのデータブロックの一部として作成されます。

[Blenderのデータシステムの詳細を読む](#) »

新規オブジェクト

Enter Edit Mode

選択されている場合、新しいオブジェクトを作成するときに、編集モードが自動的にアクティブになります。

Align To

World

新しいオブジェクトがワールド座標に揃えられます。

View

新しいオブジェクトが、ビューの座標に揃えられます。

Undo

Global Undo

メモリ内のファイルを完全なコピーとして保持することで動作します (したがって、より多くのメモリを必要とする)。

Step

利用可能なアンドゥのステップ数。

Memory Limit

Mbでのメモリの最大使用量 (0 は無制限)。

[取り消しと再実行のオプションについての詳細を読む](#) »

Grease Pencil

Grease Pencil を使うと、鉛筆のようなツールで3Dビューポートに描画することができます。

Manhattan Distance

マウス移動が記録されるまでに、水平方向または垂直方向に移動する必要があるピクセルの最小数。

Euclidian Distance

マウス移動が記録されるまでに、移動する必要がある最小距離。

Eraser Radius

grease pencilで使用する消しゴムのサイズを指定します。

Smooth Stroke

鉛筆のストロークの後の仕上げを滑らかにします。

Playback

Allow Negative Frame

設定されると、負のフレーム番号も使用できるようになります。

Keyframing

多くの状況では、アニメーションはキーフレームで制御されます。状態の値 (すなわち位置)はキーフレームに記録され、2つのキーフレーム間のアニメーションはBlenderが補間してくれます。

Visual Keying

拘束のついているオブジェクトに対して、自動的にVisualキーを使用します。

Only Insert Needed

有効にすると、必要なときにだけ新しいキーフレームが作成されます。

Auto Keyframing

オブジェクトとボーンに、自動的にキーフレームを挿入します。Auto Keyframe はデフォルトでは有効になっていません。

Only Insert Available

利用可能な曲線に、自動的にキーフレームを挿入する。

New F-Curve Defaults

Interpolation




これが制御するのは、2つのキーフレーム間の状態の計算方法です。新しいキーフレームのデフォルトの補間はベジエで、線形または定数がより急激であるのに対し、加速と減速が滑らかになります

XYZ to RGB

X, Y, Z のアニメーションカーブ (位置、スケール、回転) の色を、X, Y, Z 軸の色と同じにします。

Transform

Release confirm

オブジェクトの上で LMB  ドラッグすると移動するようになります。この (およびその他の) 変形の確定は、デフォルトでは LMB  を押す必要があります。このオプションが有効になっていると、LMB  を離すことが変形の確定として機能します。

Duplicate Data

'Duplicate Data' のチェックボックスは、複製するオブジェクトのどのデータをコピーして、どのデータをリンクのままにするかを定義します。チェックマークの付いたボックスはすべて、データがオブジェクトの複製と一緒にコピーされます。チェックされていないボックスのところはすべてその代わりに、複製されるソースオブジェクトからそれらのデータがリンクされることになります。

たとえば、メッシュにチェックしている場合、メッシュデータの完全なコピーが新しいオブジェクトとともに作成され、そして各メッシュは複製とは独立して動作します。メッシュボックスがオフのままの場合は、1つのオブジェクトのメッシュを変更するときに、変更が複製したオブジェクトにミラーリングされます。

同じ規則が 'Duplicate Data' リストのチェックボックスそれぞれに適用されます。

Managing presets

Blenderでは、入力の構成を複数のPresetで定義することができます。デフォルトのキーマップを削除して作成するのではなく、マウスとキーボードの両方のための新しいPresetsをそのまま追加することができます。マウスオプションは、ウィンドウの左側、および上の写真のキーボードオプションの右側に記載されています。

Adding and deleting presets



デフォルトの設定に何か変更をする前に、図に示すように"プラス"記号をクリックして、新しいPreset追加しておきましょう。Blenderが新しいプリセットの名前を指定するよう求めてきて、その後ならリストからPresetを選択して編集できるようになります。Presetを削除したい場合は、リストからそれを選択し、"マイナス"記号をクリックします。

Selecting presets

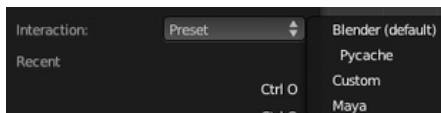
使用するプリセットの変更は、次のいずれかの方法で行うことができます:

- 起動時かHelp » Splash Screenを選択して、スプラッシュ画面のInteractionメニューから設定を選択する。
- User Preferences Inputウィンドウから設定を選択する。

Note

上記の各オプションが変更するのは、現時点のファイルのプリセットのみだということに注意してください。File » New か File » Openと選択すると、デフォルトのプリセットが再ロードされます。


Setting presets to default



一旦マウスとキーボードのPresetsを設定できたら、これをデフォルトの設定にしていけることができます。:

- User Preferences Inputエディタを開き、プリセットリストからプリセットを選択するか、
- スプラッシュ画面からプリセット設定を選択します。
- User Preferencesウィンドウで Save As Defaultオプションを使用するか、CtrlUを押して、コンフィグを保存します。

Export/Import key configuration

場合によっては、外部ファイルに設定を保存する必要があります(例えば、新しいシステムをインストールしたり、あなたのキーマップ設定をコミュニティと共有する必要がある場合)。InputタブのヘッダーでExport Key Configurationをシンプルに LMB をクリックすると、ファイルブラウザが開き、コンフィグを格納する場所を選択できるようになります。Import Key Configuration ボタンをクリックすると、お使いのコンピュータ上にある、Blender以外のキーマップの構成をインストールします。

Mouse

Emulate 3 Button Mouse

3ボタンマウスがなくてもBlenderを使用することが可能になります(例えば、2ボタンマウスや、アップルのシングルボタンマウス、またはラップトップなど)。この機能は、キー/マウスボタンのコンボでエミュレートすることができます。このオプションは、Select Withが Rightに設定されている場合にのみ利用可能です。

[Read more about emulating a 3 button mouse »](#)

Continuous Grab

ビューの外でもマウスを動かせるようにします(たとえば、平行移動、回転、拡大縮小)。

Drag Threshold

ユーザーインターフェイス要素が移動されたときBlenderが認識するまでのピクセル数。

Select with

どのボタンを選択に使用するかを選べます(もうひとつは、3Dカーソルの配置に使用されます)。

Double Click

ダブルクリックの時間(ミリ秒)。

Note


マウスの代わりにグラフィックタブレットを使用していて、圧力が正常に動作しない場合は、マウスポインタをBlenderのウィンドウに配置して、グラフィックタブレットを抜いたり/再接続したりしてみてください。これで治ることもあります。

Numpad emulation


テンキーは、Blenderではかなり頻繁に使用されます。通常の数字キーと同じキーとしてではなくです。テンキーの無いキーボード(ラップトップなど)を使用している場合は、標準の数字キーをテンキーの役割のキーとして扱うようにBlenderに指示することができます。Emulate Numpadを確認してみてください。

View manipulation



Orbit Style

3Dビューを回転させるときにBlenderがどのように動作するかを選択します (デフォルトは MMB で)。2つのスタイルが用意されています。あなたがMayaかCinema 4Dから来た方なら、Turntableのほうが好みかもしれません。


Zoom Style

ズームインやズームアウトをCtrl MMB でするときの、お好みのスタイルを選択します

Scale

Scale方式でのズームは、ビューの最初にクリックした場所に依存します。ズームアウトするには、画面の端で Ctrl MMB でドラッグしたまま中心の方へ動かします。ズームインするには、画面の中心で Ctrl MMB でドラッグしながら画面端のほうへ動かします。

Continue

Continueズームオプションを選択すると、Ctrl MMB で最初のクリックポイントから遠くに移動することで、ズームの速度を (および値ではなく)制御することができます。最初のクリックポイントから上か右に動かすとズームアウトし、下か左に動かすとズームインします。遠くに動かすほど、ズームの動きが速くなります。方向を変えることもでき、Vertical と Horizontal のラジオボタンと Invert Zoom Directionオプションによって行えます。

Dolly

Dollyズームの動作は、ズーム速度が一定であることを除けば、Continueズームと似ています。

Vertical

上に移動するとズームアウトし、下に移動するとズームインします。

Horizontal

左に動かすとズームインし、右に動かすとズームアウトします。

Invert Zoom Direction

Dollyと Continueのズーム方向を反転します。

Invert Wheel Zoom Direction

マウスホイールのズーム方向を反転します。

NDOF device

3Dマウスの感度を設定します。

Keymap editor



キーマップエディタでは、デフォルトのホットキーを変更できます。各ウィンドウでのキーマップを変更することができます。

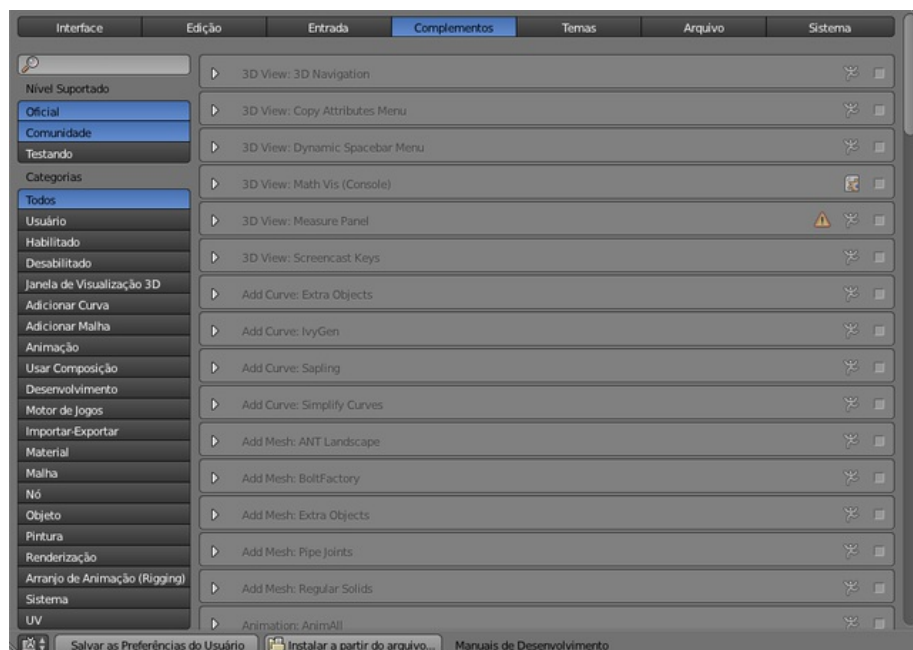
1. 変更したいキーマップを選択し、白の矢印をクリックするとマップツリーが開きます。
2. どのInputで機能を制御するかを選択します
 - Keyboard: ホットキーまたはコンボホットキーのみ (E または ⇧ ShiftE)。
 - Mouse: 左/中央/右クリック。CAlt, ⇧ Shift, Ctrl, ⌘ Cmdと組み合わせることができます。
 - Tweak: クリックとドラッグ。前項の4つのキーと組み合わせることもできます。
 - Text input: テキストで入力することにより、この機能を使用する
 - Timer: 時間間隔に基づいたアクションの制御に使用します。例えば、デフォルトではアニメーションのステップにはタイマ0を使用していて、タイマ1を使用するとビューがスムーズになります。
3. 必要に応じてホットキーを変更します。ショートカットの入力をクリックして、新しいショートカットを入力するだけです。

キーマップのデフォルト設定を復元する場合は、このキーマップの右上にある Restoreボタンをクリックするだけです。

インターフェイス・編集・入力・アドオン・テーマ・ファイル・システム

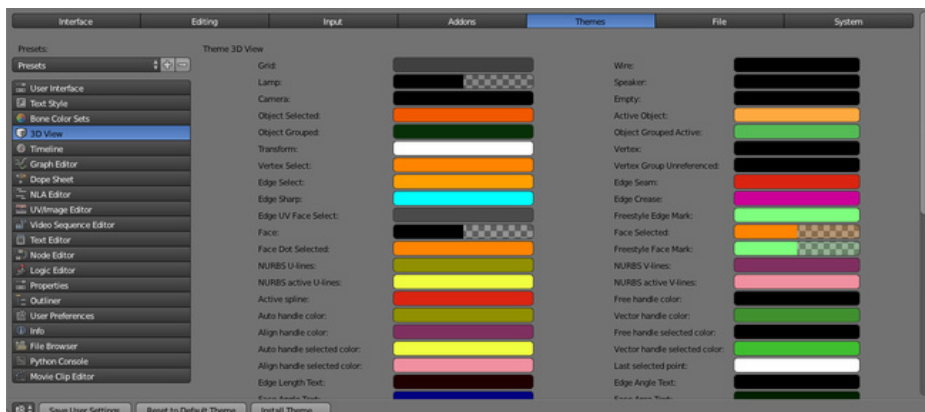
ADD-Onsタブでは、Blenderのデフォルトでは有効化されていない副次的なオプションを管理できます。*Install Add-Ons*では、新機能を追加することもできます。アドオンの数はますます多くなり、Blender-communityが作成していくため、見逃していた機能がないか探してみましょう(または自作するのもよいかもしれません)。

アドオンの使用方法の詳細については、[アドオンのページ](#)を参照してください。



テーマのカスタマイズ

前述のように、Blenderはカスタマイズの幅が非常に広くできています—いくつかの設定は、インターフェイスの外観と色に影響を与えます。これらの設定は、Themesタブの下にあります。



各エディタの色が別個に設定できます—左のマルチ選択リストで変更したいエディタを選んで、必要に応じて色を調整しましょう。変更が画面上でリアルタイムに表示されることも注目点です。さらに、変更できる詳細には、3D View や Graph Editor でのドットサイズのようなものもあります。

Customizing themes with Python

スクリプトからテーマを制御したい場合は、パラメータを個別に記載したサンプルスクリプトを以下からどうぞ:

```
# A simple way to make global theme color changes.
```

```
import bpy
```

```
#Shared elements
```

```
back = (0.2667,0.2667,0.2667)
```

```
viewbg = (0.2667,0.3,0.34)
```

```
back_alt = (0.1647,0.1647,0.1647)
```

```
button = (0.2667,0.2667,0.2667)
```

```
grid = (0.5,0.5,0.5)
```

```
header = (0.2667,0.2667,0.2667)
```

```
panel = (0.2667,0.2667,0.2667,0.75)
```

```
text = (0.8,0.8,0.8)
```

```
text_hi = (1,1,1)
```

```
wire = (1,1,1)
```

```
lamp = (1,1,1,1)
```

```
#UI shared
```

```
inner = (0.2667,0.2667,0.2667,1)
```

```
inner_sel = (0.2667,0.2667,0.2667,1)
```

```
item = (0.39,0.39,0.39,1)
```

```
outline = (0.1,0.1,0.1)
```

```
theme = bpy.context.user_preferences.themes['Default']
```

```
theme.view_3d.act_spline
```

```
theme.view_3d.back = viewbg
```

```
theme.view_3d.bone_pose
```

```
theme.view_3d.bone_solid
```

```
theme.view_3d.button= button
```

```
theme.view_3d.button_text = text
```

```
theme.view_3d.button_text_hi = text_hi
```

```
theme.view_3d.button_title = text
```

```
theme.view_3d.edge_crease
```

```
theme.view_3d.edge_facesel
```

```
theme.view_3d.edge_seam
```

```
theme.view_3d.edge_select
```

```
theme.view_3d.edge_sharp
```

```
theme.view_3d.editmesh_active
```

```
theme.view_3d.face
```

```
theme.view_3d.face_dot
```

```
theme.view_3d.face_select
```

```
theme.view_3d.facedot_size
```

```
theme.view_3d.frame_current
```

```
theme.view_3d.grid = grid
```

```
theme.view_3d.handle_align
```

```
theme.view_3d.handle_auto
```

```
theme.view_3d.handle_free
```

```
theme.view_3d.handle_sel_align
```

```
theme.view_3d.handle_sel_auto
```

```
theme.view_3d.handle_sel_free
```

```
theme.view_3d.handle_sel_vect
```

```
theme.view_3d.handle_vect
```

```
theme.view_3d.header = header
```

```
theme.view_3d.header_text = text
```

```
theme.view_3d.header_text_hi = text_hi
```

```
theme.view_3d.lamp = lamp
```

```
theme.view_3d.lastsel_point
```

```
theme.view_3d.normal
theme.view_3d.nurb_sel_uline
theme.view_3d.nurb_sel_vline
theme.view_3d.nurb_uline
theme.view_3d.nurb_vline
theme.view_3d.object_active
theme.view_3d.object_grouped
theme.view_3d.object_grouped_active
theme.view_3d.object_selected
theme.view_3d.panel = panel
theme.view_3d.text = text
theme.view_3d.text_hi = text_hi
theme.view_3d.title = text
theme.view_3d.transform
theme.view_3d.vertex
theme.view_3d.vertex_normal
theme.view_3d.vertex_select
theme.view_3d.vertex_size
theme.view_3d.wire = wire

theme.console.back= back_alt
theme.console.button = button
theme.console.button_text = text
theme.console.button_text_hi = text_hi
theme.console.button_title = text
theme.console.cursor
theme.console.header = header
theme.console.header_text = text
theme.console.header_text_hi = text_hi
theme.console.line_error
theme.console.line_info
theme.console.line_input
theme.console.line_output
theme.console.text = text
theme.console.text_hi = text_hi
theme.console.title = text

theme.dopesheet_editor.active_channels_group
theme.dopesheet_editor.back= back
theme.dopesheet_editor.button = button
theme.dopesheet_editor.button_text = text
theme.dopesheet_editor.button_text_hi = text_hi
theme.dopesheet_editor.button_title = text
theme.dopesheet_editor.channel_group
theme.dopesheet_editor.channels
theme.dopesheet_editor.channels_selected
theme.dopesheet_editor.dopesheet_channel
theme.dopesheet_editor.dopesheet_subchannel
theme.dopesheet_editor.frame_current
theme.dopesheet_editor.grid = grid
theme.dopesheet_editor.header = header
theme.dopesheet_editor.header_text = text
theme.dopesheet_editor.header_text_hi = text_hi
theme.dopesheet_editor.list = button
theme.dopesheet_editor.list_text = text
theme.dopesheet_editor.list_text_hi = text
theme.dopesheet_editor.list_title = text
theme.dopesheet_editor.long_key
theme.dopesheet_editor.long_key_selected
theme.dopesheet_editor.text = text
theme.dopesheet_editor.text_hi = text_hi
theme.dopesheet_editor.title = text
theme.dopesheet_editor.value_sliders
theme.dopesheet_editor.view_sliders

theme.file_browser.active_file
theme.file_browser.active_file_text
theme.file_browser.back= back
theme.file_browser.button = button
theme.file_browser.button_text = text
theme.file_browser.button_text_hi = text_hi
theme.file_browser.button_title = text
theme.file_browser.header = header
theme.file_browser.header_text = text
theme.file_browser.header_text_hi = text_hi
theme.file_browser.list = button
theme.file_browser.list_text = text
theme.file_browser.list_text_hi = text
theme.file_browser.list_title = text
theme.file_browser.scroll_handle = button
theme.file_browser.scrollbar = button
theme.file_browser.selected_file
theme.file_browser.text = text
theme.file_browser.text_hi = text_hi
theme.file_browser.tiles = back
theme.file_browser.title = text

theme.graph_editor.active_channels_group
theme.graph_editor.back= back alt
```

```
theme.graph_editor.button = button
theme.graph_editor.button_text = text
theme.graph_editor.button_text_hi = text_hi
theme.graph_editor.button_title = text
theme.graph_editor.channel_group
theme.graph_editor.channels_region
theme.graph_editor.dopesheet_channel
theme.graph_editor.dopesheet_subchannel
theme.graph_editor.frame_current
theme.graph_editor.grid = grid
theme.graph_editor.handle_align
theme.graph_editor.handle_auto
theme.graph_editor.handle_free
theme.graph_editor.handle_sel_align
theme.graph_editor.handle_sel_auto
theme.graph_editor.handle_sel_free
theme.graph_editor.handle_sel_vect
theme.graph_editor.handle_vect
theme.graph_editor.handle_vertex
theme.graph_editor.handle_vertex_select
theme.graph_editor.handle_vertex_size
theme.graph_editor.header = header
theme.graph_editor.header_text = text
theme.graph_editor.header_text_hi = text_hi
theme.graph_editor.lastsel_point
theme.graph_editor.list = button
theme.graph_editor.list_text = text
theme.graph_editor.list_text_hi = text
theme.graph_editor.list_title = text
theme.graph_editor.panel = back
theme.graph_editor.text = text
theme.graph_editor.text_hi = text_hi
theme.graph_editor.title = text
theme.graph_editor.vertex
theme.graph_editor.vertex_select
theme.graph_editor.vertex_size
theme.graph_editor.window_sliders
```

```
theme.image_editor.back= back_alt
theme.image_editor.button = button
theme.image_editor.button_text = text
theme.image_editor.button_text_hi = text_hi
theme.image_editor.button_title = text
theme.image_editor.editmesh_active
theme.image_editor.face
theme.image_editor.face_dot
theme.image_editor.face_select
theme.image_editor.facedot_size
theme.image_editor.header = header
theme.image_editor.header_text = text
theme.image_editor.header_text_hi = text_hi
theme.image_editor.scope_back
theme.image_editor.text = text
theme.image_editor.text_hi = text_hi
theme.image_editor.title = text
theme.image_editor.vertex
theme.image_editor.vertex_select
theme.image_editor.vertex_size
```

```
theme.info.back= back
theme.info.button = button
theme.info.button_text = text
theme.info.button_text_hi = text_hi
theme.info.button_title = text
theme.info.header = header
theme.info.header_text = text
theme.info.header_text_hi = text_hi
theme.info.text = text
theme.info.text_hi = text_hi
theme.info.title = text
```

```
theme.logic_editor.back= back
theme.logic_editor.button = button
theme.logic_editor.button_text = text
theme.logic_editor.button_text_hi = text_hi
theme.logic_editor.button_title = text
theme.logic_editor.header = header
theme.logic_editor.header_text = text
theme.logic_editor.header_text_hi = text_hi
theme.logic_editor.panel = back
theme.logic_editor.text = text
theme.logic_editor.text_hi = text_hi
theme.logic_editor.title = text
```

```
theme.nla_editor.back= back
theme.nla_editor.bars
theme.nla_editor.bars_selected
```

```
theme.nla_editor.button = button
theme.nla_editor.button_text = text
theme.nla_editor.button_text_hi = text_hi
theme.nla_editor.button_title = text
theme.nla_editor.frame_current
theme.nla_editor.grid = grid
theme.nla_editor.header = header
theme.nla_editor.header_text = text
theme.nla_editor.header_text_hi = text_hi
theme.nla_editor.list = button
theme.nla_editor.list_text = text
theme.nla_editor.list_text_hi = text
theme.nla_editor.list_title = text
theme.nla_editor.strips
theme.nla_editor.strips_selected
theme.nla_editor.text = text
theme.nla_editor.text_hi = text_hi
theme.nla_editor.title = text
theme.nla_editor.view_sliders
```

```
theme.node_editor.back= back_alt
theme.node_editor.button = button
theme.node_editor.button_text = text
theme.node_editor.button_text_hi = text_hi
theme.node_editor.button_title = text
theme.node_editor.converter_node
theme.node_editor.group_node
theme.node_editor.header = header
theme.node_editor.header_text = text
theme.node_editor.header_text_hi = text_hi
theme.node_editor.in_out_node
theme.node_editor.list = button
theme.node_editor.list_text = text
theme.node_editor.list_text_hi = text
theme.node_editor.list_title = text
theme.node_editor.node_backdrop
theme.node_editor.operator_node
theme.node_editor.selected_text
theme.node_editor.text = text
theme.node_editor.text_hi = text_hi
theme.node_editor.title = text
theme.node_editor.wire
theme.node_editor.wire_select
```

```
theme.outliner.back= back
theme.outliner.button = button
theme.outliner.button_text = text
theme.outliner.button_text_hi = text_hi
theme.outliner.button_title = text
theme.outliner.header = header
theme.outliner.header_text = text
theme.outliner.header_text_hi = text_hi
theme.outliner.text = text
theme.outliner.text_hi = text_hi
theme.outliner.title = text
```

```
theme.properties.back= back
theme.properties.button = button
theme.properties.button_text = text
theme.properties.button_text_hi = text_hi
theme.properties.button_title = text
theme.properties.header = header
theme.properties.header_text = text
theme.properties.header_text_hi = text_hi
theme.properties.panel = back
theme.properties.text = text
theme.properties.text_hi = text_hi
theme.properties.title = text
```

```
theme.sequence_editor.audio_strip
theme.sequence_editor.back= back
theme.sequence_editor.button = button
theme.sequence_editor.button_text = text
theme.sequence_editor.button_text_hi = text_hi
theme.sequence_editor.button_title = text
theme.sequence_editor.draw_action
theme.sequence_editor.effect_strip
theme.sequence_editor.frame_current
theme.sequence_editor.grid = grid
theme.sequence_editor.header = header
theme.sequence_editor.header_text = text
theme.sequence_editor.header_text_hi = text_hi
theme.sequence_editor.image_strip
theme.sequence_editor.keyframe
theme.sequence_editor.meta_strip
theme.sequence_editor.movie_strip
theme.sequence_editor.plugin_strip
theme.sequence_editor.scene strip
```

```
theme.sequence_editor.text = text
theme.sequence_editor.text_hi = text_hi
theme.sequence_editor.title = text
theme.sequence_editor.transition_strip
theme.sequence_editor.window_sliders
```

```
theme.text_editor.back= back_alt
theme.text_editor.button = button
theme.text_editor.button_text = text
theme.text_editor.button_text_hi = text_hi
theme.text_editor.button_title = text
theme.text_editor.cursor
theme.text_editor.header = header
theme.text_editor.header_text = text
theme.text_editor.header_text_hi = text_hi
theme.text_editor.line_numbers_background
theme.text_editor.scroll_bar = back
theme.text_editor.selected_text
theme.text_editor.syntax_builtin
theme.text_editor.syntax_comment
theme.text_editor.syntax_numbers
theme.text_editor.syntax_special
theme.text_editor.syntax_string
theme.text_editor.text = text
theme.text_editor.text_hi = text_hi
theme.text_editor.title = text
```

```
theme.timeline.back= back_alt
theme.timeline.button = button
theme.timeline.button_text = text
theme.timeline.button_text_hi = text_hi
theme.timeline.button_title = text
theme.timeline.frame_current
theme.timeline.grid = grid
theme.timeline.header = header
theme.timeline.header_text = text
theme.timeline.header_text_hi = text_hi
theme.timeline.text = text
theme.timeline.text_hi = text_hi
theme.timeline.title = text
```

```
theme.user_preferences.back= back
theme.user_preferences.button = button
theme.user_preferences.button_text = text
theme.user_preferences.button_text_hi = text_hi
theme.user_preferences.button_title = text
theme.user_preferences.header = header
theme.user_preferences.header_text = text
theme.user_preferences.header_text_hi = text_hi
theme.user_preferences.text = text
theme.user_preferences.text_hi = text_hi
theme.user_preferences.title = text
```

```
theme.user_interface.icon_file
```

```
theme.user_interface.wcol_box.inner = inner
theme.user_interface.wcol_box.inner_sel
theme.user_interface.wcol_box.item = item
theme.user_interface.wcol_box.outline = outline
theme.user_interface.wcol_box.shadedown
theme.user_interface.wcol_box.shadetop
theme.user_interface.wcol_box.show_shaded
theme.user_interface.wcol_box.text = text
theme.user_interface.wcol_box.text_sel = text_hi
```

```
theme.user_interface.wcol_list_item.inner = inner
theme.user_interface.wcol_list_item.inner_sel
theme.user_interface.wcol_list_item.item = item
theme.user_interface.wcol_list_item.outline = outline
theme.user_interface.wcol_list_item.shadedown
theme.user_interface.wcol_list_item.shadetop
theme.user_interface.wcol_list_item.show_shaded
theme.user_interface.wcol_list_item.text = text
theme.user_interface.wcol_list_item.text_sel = text_hi
```

```
theme.user_interface.wcol_menu.inner = inner
theme.user_interface.wcol_menu.inner_sel
theme.user_interface.wcol_menu.item = item
theme.user_interface.wcol_menu.outline = outline
theme.user_interface.wcol_menu.shadedown
theme.user_interface.wcol_menu.shadetop
theme.user_interface.wcol_menu.show_shaded
theme.user_interface.wcol_menu.text = text
theme.user_interface.wcol_menu.text_sel = text_hi
```

```
theme.user_interface.wcol_menu_back.inner = inner
theme.user_interface.wcol_menu_back.inner_sel
```

```
theme.user_interface.wcol_menu_back.item = item
theme.user_interface.wcol_menu_back.outline = outline
theme.user_interface.wcol_menu_back.shadedown
theme.user_interface.wcol_menu_back.shadetop
theme.user_interface.wcol_menu_back.show_shaded
theme.user_interface.wcol_menu_back.text = text
theme.user_interface.wcol_menu_back.text_sel = text_hi
```

```
theme.user_interface.wcol_menu_item.inner
theme.user_interface.wcol_menu_item.inner_sel
theme.user_interface.wcol_menu_item.item = item
theme.user_interface.wcol_menu_item.outline = outline
theme.user_interface.wcol_menu_item.shadedown
theme.user_interface.wcol_menu_item.shadetop
theme.user_interface.wcol_menu_item.show_shaded
theme.user_interface.wcol_menu_item.text = text
theme.user_interface.wcol_menu_item.text_sel = text_hi
```

```
theme.user_interface.wcol_num.inner = inner
theme.user_interface.wcol_num.inner_sel
theme.user_interface.wcol_num.item = item
theme.user_interface.wcol_num.outline = outline
theme.user_interface.wcol_num.shadedown
theme.user_interface.wcol_num.shadetop
theme.user_interface.wcol_num.show_shaded
theme.user_interface.wcol_num.text = text
theme.user_interface.wcol_num.text_sel = text_hi
```

```
theme.user_interface.wcol_numslider.inner = inner
theme.user_interface.wcol_numslider.inner_sel
theme.user_interface.wcol_numslider.item = item
theme.user_interface.wcol_numslider.outline = outline
theme.user_interface.wcol_numslider.shadedown
theme.user_interface.wcol_numslider.shadetop
theme.user_interface.wcol_numslider.show_shaded
theme.user_interface.wcol_numslider.text = text
theme.user_interface.wcol_numslider.text_sel = text_hi
```

```
theme.user_interface.wcol_option.inner = inner
theme.user_interface.wcol_option.inner_sel
theme.user_interface.wcol_option.item = item
theme.user_interface.wcol_option.outline = outline
theme.user_interface.wcol_option.shadedown
theme.user_interface.wcol_option.shadetop
theme.user_interface.wcol_option.show_shaded
theme.user_interface.wcol_option.text = text
theme.user_interface.wcol_option.text_sel = text_hi
```

```
theme.user_interface.wcol_progress.inner = inner
theme.user_interface.wcol_progress.inner_sel
theme.user_interface.wcol_progress.item = item
theme.user_interface.wcol_progress.outline = outline
theme.user_interface.wcol_progress.shadedown
theme.user_interface.wcol_progress.shadetop
theme.user_interface.wcol_progress.show_shaded
theme.user_interface.wcol_progress.text = text
theme.user_interface.wcol_progress.text_sel = text_hi
```

```
theme.user_interface.wcol_pulldown.inner = inner
theme.user_interface.wcol_pulldown.inner_sel
theme.user_interface.wcol_pulldown.item = item
theme.user_interface.wcol_pulldown.outline = outline
theme.user_interface.wcol_pulldown.shadedown
theme.user_interface.wcol_pulldown.shadetop
theme.user_interface.wcol_pulldown.show_shaded
theme.user_interface.wcol_pulldown.text = text
theme.user_interface.wcol_pulldown.text_sel = text_hi
```

```
theme.user_interface.wcol_radio.inner = inner
theme.user_interface.wcol_radio.inner_sel
theme.user_interface.wcol_radio.item = item
theme.user_interface.wcol_radio.outline = outline
theme.user_interface.wcol_radio.shadedown
theme.user_interface.wcol_radio.shadetop
theme.user_interface.wcol_radio.show_shaded
theme.user_interface.wcol_radio.text = text
theme.user_interface.wcol_radio.text_sel = text_hi
```

```
theme.user_interface.wcol_regular.inner = inner
theme.user_interface.wcol_regular.inner_sel
theme.user_interface.wcol_regular.item = item
theme.user_interface.wcol_regular.outline = outline
theme.user_interface.wcol_regular.shadedown
theme.user_interface.wcol_regular.shadetop
theme.user_interface.wcol_regular.show_shaded
theme.user_interface.wcol_regular.text = text
theme.user_interface.wcol_regular.text_sel = text_hi
```

```
theme.user_interface.wcol_scroll.inner = inner
theme.user_interface.wcol_scroll.inner_sel
theme.user_interface.wcol_scroll.item = item = inner
theme.user_interface.wcol_scroll.outline = outline
```

```
theme.user_interface.wcol_scroll.shadedown
theme.user_interface.wcol_scroll.shadetop
theme.user_interface.wcol_scroll.show_shaded
theme.user_interface.wcol_scroll.text = text
theme.user_interface.wcol_scroll.text_sel = text_hi

theme.user_interface.wcol_state.inner_anim
theme.user_interface.wcol_state.inner_anim_sel
theme.user_interface.wcol_state.inner_driven
theme.user_interface.wcol_state.inner_driven_sel
theme.user_interface.wcol_state.inner_key
theme.user_interface.wcol_state.inner_key_sel

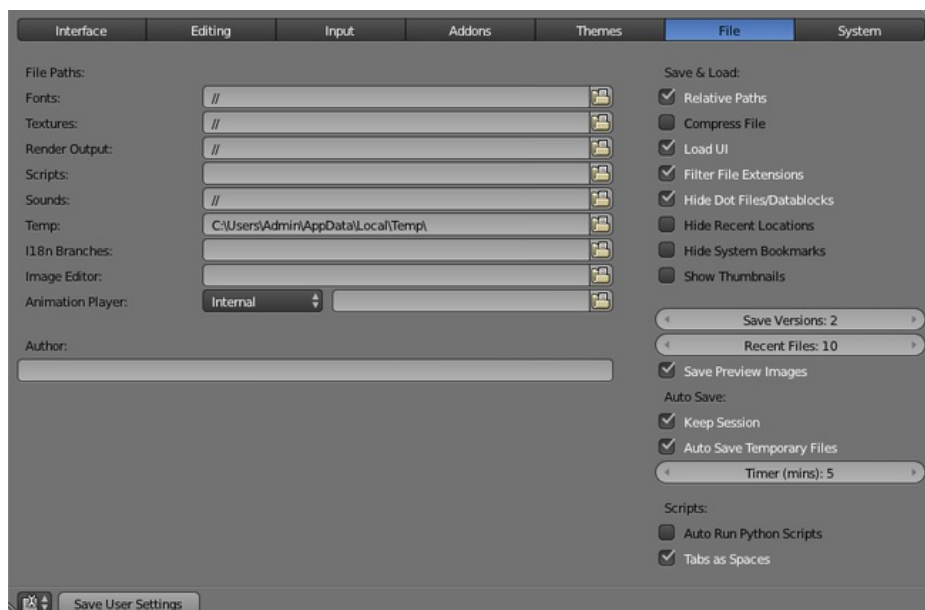
theme.user_interface.wcol_text.inner = inner
theme.user_interface.wcol_text.inner_sel
theme.user_interface.wcol_text.item = item
theme.user_interface.wcol_text.outline = outline
theme.user_interface.wcol_text.shadedown
theme.user_interface.wcol_text.shadetop
theme.user_interface.wcol_text.show_shaded
theme.user_interface.wcol_text.text = text
theme.user_interface.wcol_text.text_sel = text_hi

theme.user_interface.wcol_toggle.inner = inner
theme.user_interface.wcol_toggle.inner_sel
theme.user_interface.wcol_toggle.item = item
theme.user_interface.wcol_toggle.outline = outline
theme.user_interface.wcol_toggle.shadedown
theme.user_interface.wcol_toggle.shadetop
theme.user_interface.wcol_toggle.show_shaded
theme.user_interface.wcol_toggle.text = text
theme.user_interface.wcol_toggle.text_sel = text_hi

theme.user_interface.wcol_tool.inner = inner
theme.user_interface.wcol_tool.inner_sel
theme.user_interface.wcol_tool.item = item
theme.user_interface.wcol_tool.outline = outline
theme.user_interface.wcol_tool.shadedown
theme.user_interface.wcol_tool.shadetop
theme.user_interface.wcol_tool.show_shaded
theme.user_interface.wcol_tool.text = text
theme.user_interface.wcol_tool.text_sel = text_hi
```


File Preferences

画像でお見せしているのは、下文で説明するファイル環境の設定です。



File Paths

重要なプロジェクトで作業するときには、これを設定しておくが賢明です。使用するそれぞれのファイルの種類ごとにデフォルトパスを設定しておきましょう。

こちらが設定の例です:

```

フォント      //fonts/
テクスチャー  //textures/
テクスチャプラグイン //plugins/texture/
シーケンスプラグイン //plugins/sequence/
レンダリングの出力 //renders/
スクリプト   //scripts/
サウンド     //sounds/
テンポラリ   //tmp/

```

blenderは自動的にプロジェクトの構造を作成しないことに注意してください。ファイルブラウザに手動ですべてのディレクトリを作成する必要があります。

Save & Load

Relative Paths

デフォルトでは、外部ファイルには相対パスが使用されます。これは、Blenderのファイルが保存されている場合にのみ動作します。

Compress File

.blend ファイルを保存するときに圧縮します

Load UI

デフォルト設定では、保存したファイルのウィンドウレイアウト ([Screens](#)) をロードします。これは個別に変更することができ、File BrowserウィンドウのOpen Blender File/パネルからファイルをロードするときに行えます。



File extension filter

Filter File Extensions

これを有効にすることによって、ファイルのダイアログウィンドウが、適合性のあるファイルだけを表示するようになります (すなわち .blend のファイルであり、完全なBlenderとしての設定でロードできるときにだけ)。ファイルタイプのこの抜粋は、ファイルダイアログウィンドウでも変更できます。

Hide Dot File/Datablocks

ファイルブラウザ上の ".*" で始まるファイルを非表示にする (LinuxおよびAppleのシステムでは、".*" ファイルは非表示になっている)。

Hide Recent Locations

File BrowserウィンドウのRecent/パネルという、最近アクセスしたフォルダを表示するパネルを非表示にします。

Show Thumbnails

File Browserを使用するときに、画像や動画のサムネイルを表示します。

Auto Save

Save Versions

同じファイル用に作成されたバージョンの数 (バックアップ用)。

Recent Files

File » Open Recentに表示されるファイルの数。

Save Preview Images

File Browserウィンドウ内の画像やマテリアルのプレビューがオンデマンドで作成されます。こういったプレビューを .blendファイルに保存するには、このオプションを有効にしてください(その代わりに .blendファイルのサイズは増えます)。

Auto Save Temporary File

自動保存を有効にします(一時ファイルを作成する)。

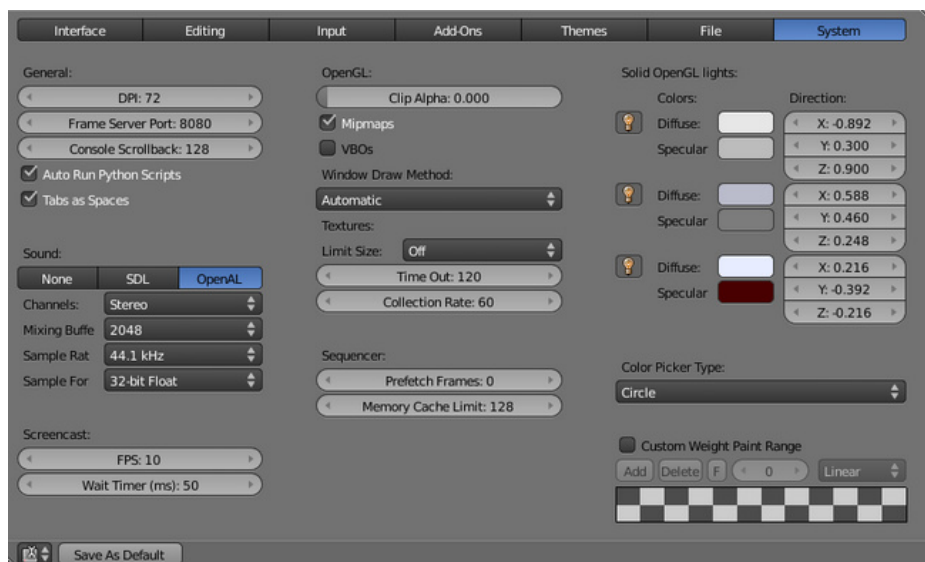
Timer

自動セーブごとの間の待ち時間。

[自動保存オプションの詳細についてを読む »](#)

System preferences

画像に示しているのは、BlenderのUser PreferencesウィンドウにあるSystemタブです。いくつかのオプションを以下に説明します。



General

DPI

この値は画面の解像度で、インターフェイスのフォントのサイズを制御します。インタフェースの部分のサイズ変更は、Ctrlを押してMMBドラッグし、パネルの上で左右に動かして画面内のサイズを変更するのもよいかもしれません。

Frame Server Port

フレームサーバーレンダリングのためのフレームサーバのポートを指定します。分散レンダリングで作業するときに使用されます。

Console Scrollback

コンソールウィンドウのメモリに一時保存される行数。

Auto Run Python Scripts

.blendファイルのスキプトの自動的実行を許可します (入手元が定かではないファイルは安全でない可能性があります)。

Tabs as Spaces

テキストファイルの内のタブをスペースに変換します。タイピング中やテキストファイルがロードされたときに変換されます。

Sound

Sound

オーディオ出力デバイスを設定します。

Channels

オーディオのチャンネル数を設定します。

Mixing Buffer

オーディオミキシングバッファで使用されるサンプル数を設定します。

Sample Rate

オーディオサンプルのレートを設定します。

Sample Format

オーディオサンプルの形式を設定します。

Screencast

FPS

スクリーンキャスト再生のフレームレート

Wait Timer

スクリーンキャストのための各フレームの記録の間隔(ミリ秒単位)。

Open GL

Clip Alpha

3Dビューポートでこのしきい値よりも下のアルファ値を刈り取る。

Mipmaps

ミップマップフィルタリングを使用した3Dビューのテクスチャの縮尺。これは表示品質を向上させますが、使用されるメモリも多くなります。

Anisotropic Filtering

異方性フィルタリングのレベルを設定します。

VBOs

(OpenGLが)サポートされていない場合に、Vertex Buffer Objectsかvertex arraysを、ビューポートのレンダリングに使用します。

Window Draw Method

Automatic

グラフィックスカードとドライバに基づいて自動的に設定します。

Triple Buffer

第3のバッファを使用して、再描画を最小限にします。その代わりメモリの消費量は多くなります。

Overlap

すべての重なり合う領域を再描画します。メモリ使用量は最小限になりますが、再描画はより多くなります。

Overlap Flip

すべての重なり合う領域を再描画します。メモリ使用量は最小限になりますが、再描画は多くなります(フリッピングを行わないグラフィックドライバ用)。

Full

完全な再描画を毎回行います。参照のみ場合、または他のすべてでは不調のときに使用します。

Text Draw Options

インターフェイスのテキストのアンチエイリアスを有効にします。

Limit Size

テクスチャ表示に使用する画像の最大解像度を制限して、メモリを節約します。

Time Out

GLテクスチャへの最後のアクセスからそれが解放されるまでの時間(秒)。0に設定すると、テクスチャが割り当てられたまま保たれます。

Collection Rate

GLテクスチャのガベージコレクタの各実行間の秒数。

Sequencer

Prefetch Frames

再生中にあらかじめレンダリングするフレームの数。

Memory Cache Limit

シーケンサーのメモリキャッシュの上限(メガバイト)。

Solid OpenGL lights

3Dウィンドウでの表示をSolid viewに切り替えたとき、3つの仮想光源があり(それらはレンダリングでは表示されません)、シーンを照らすために使用されます。これらのライトの方向、ディフューズとスペキュラのライティングは、ここで変更できます。

Miscellaneous

Color Picker Type

どの種類のカラーダイアログにするのかを好きなものから選択する - 任意のカラーフィールドで LMB をクリックしたときにそれが表示されます。

Custom Weight Paint Range

Mesh skin weightingは、ボーンがキャラクターのメッシュを变形させる量を制御するために使用されます。これらの重みを可視化するため、Blenderは色勾配を使用します (from blue to green to yellow to red)。ここで独自に作成することもできます。オプションの下にあるコントロールを使用すると、色勾配のColor stopsの色の追加、削除、変更をすることができます。

30プラス30分でできる初めてのアニメーション Part I

この章では、小さな"ジンジャーブレッドマン"というキャラクターでアニメーション制作を解説していきます。ひとつひとつの手順を端々まで説明するようになりますが、インターフェイスの章 [interface](#) を読んだことを前提としています。そちらに目を通せば本書で使用される用語も理解できるはずです。

このチュートリアルPart Iでは *静止した*ジンジャーブレッドマンを構築していきます。続くPart IIでは、彼に歩いてもらうことにしましょう。

Note

キャラクターアニメーションに焦点を当てた、もっと詳細なBlenderの手引きなら、

[ブレンダー・サマー・オブ・ドキュメンテーション キャラクターアニメーションをはじめよう](#) というチュートリアルをチェックしてみましょう(これはV2.4xに向けて書かれたものですが、今でも大いに有益でしょう)。

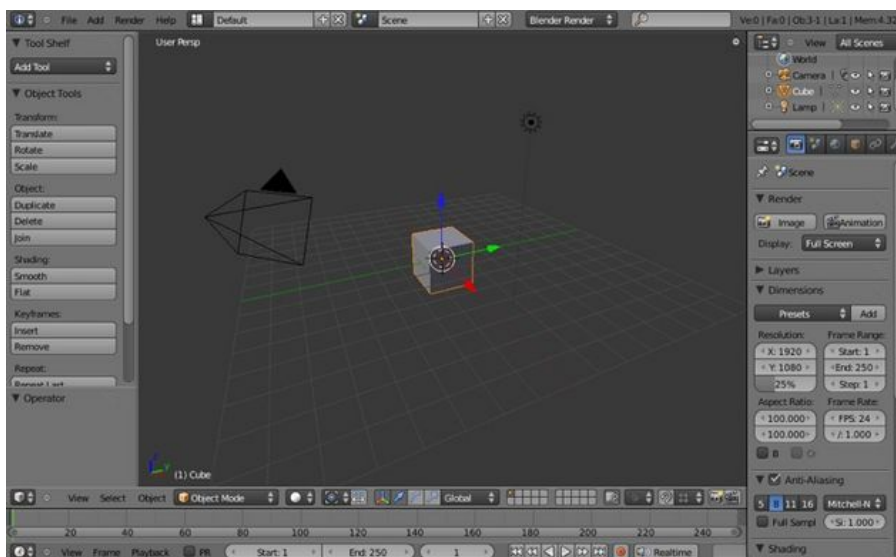


今ご覧になっている"Gus the Gingerbread Man"チュートリアルと同様に、the BSoD Intro to Character Animation tutorialは事前知識を必須としません。キャラクターを歩かせたりしゃべらせたりのイチからの制作過程を通してガイドし、ここでは出会えないBlenderの多くの強力な機能を取り扱っています。

The BSoD Intro to Character Animation は、オフラインでも読めるよう、ダウンロード可能な [PDF version \(3.75 MB\)](#) も用意されています。

ウォーミングアップ

Blenderを起動してみましょう。画面には、デフォルトのセットアップが表示されるはずですが。カメラ、ライト、およびキューブです。キューブはすでに選択されていて、そのオレンジ色の輪郭線で表示されているはずですが。(Default Blender screen as soon as you start it.)






Default Blender screen as soon as you start it (small).

オブジェクトは、それを隠せる別のレイヤーに置いて作業エリアを整理でき、さらに必要になったらいつでも、現在のシーンへそれらを戻すことができます。レイヤーを操作するにはこのようにします：

重要 : BlenderはThe BSoD Intro to Character Animation キーボードショートカット(慌てないで、すぐに慣れます)で操作します。これらのショートカットのほとんどは、対応するフレームにマウスポインタがのっているときにだけ上記の動作をします。だから、もしM が(ちょっと下に書いてあるように)思いどおりに動かないからといってくじけないで -- あとはマウスを3D-viewの上に移動させてもういちど。






Layer visibility controls.

Blenderはあなたの作業を整理できるように20のレイヤーを用意しています。3D window headerの20個のボタンから、現在どのレイヤーが表示中なのか確認できます (Layer visibility controls)。LMB  で表示レイヤーを変更することができ、トグル表示には  Shift LMB  します。最後にオンにしたレイヤーは、アクティブなレイヤーとなります。アクティブなレイヤーとはすべての新規に作成されるオブジェクトが格納される場所です。

まずはじめに、カメラとランプを別のレイヤーに移動させて、その場所をまっさらにししましょう。



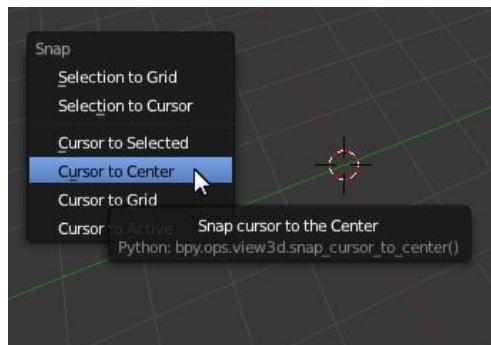
The Move to Layer popup.

RMB  でカメラを選択します。その後、 RMB  で選択範囲にランプを追加します。M を押すと小さなツールボックス、(*The Move to Layer popup*)の画像内にあるようなもの、がマウスのもとに現れます。最初のボタンがチェックされた状態になっていて、選択されたオブジェクトはレイヤ1に格納されるということを示しています。一番上の行の右端のボタンをチェックしてください。こうすると、カメラとランプがレイヤー10に移動します。


さてレイヤ1だけが表示されていることを確認してください。というのもカメラやランプを誤って消去したくないからです。Aですべてのオブジェクトを選択し、X >> Delete Selected Object(s)で消去されます。モデリングの作業を開始する前に知っておいていただきたいことはこのくらいです。

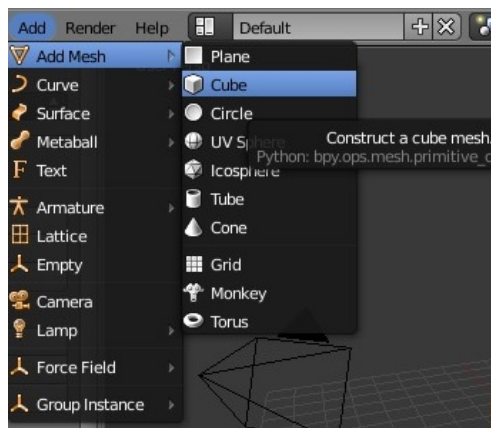
胴体の構築

1 NumPadでフロントビューに切り替え (Num lock を有効にしておいて！)、もしまだ斜行視点(Persp)になっていたら(確認するには3D Windowの左上隅を見て)、5 NumPad で直行視点(Ortho)に切り替えてください。




Snap cursor to the Center

スクリーンにキューブがない場合、追加する必要があります。カーソルが中心部(0,0,0)に位置しているかを確認し、もしずれていたらメニューから Object » Snap » Cursor to Center とするか、 Shift+S » Cursor to Center でカーソルを中心部にセットしましょう。

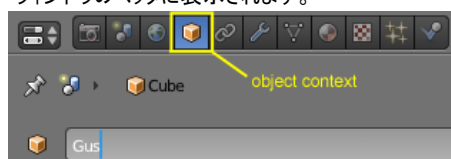


Add a cube to the scene.

メニューから Add » Mesh » Cube とするか、 Shift+A » Mesh » Cube で、キューブを追加しましょう。キューブがオレンジで表示され、アクティブであることを示しています。⇨ Tab を押して Edit Mode に入りましょう。

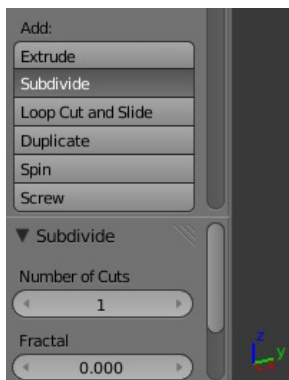
Edit Mode と Object Mode

Edit Mode は、メッシュの頂点を編集できるモードです。デフォルトでは、オブジェクトを新規作成するとすべての頂点が選択された状態になります (選択された頂点はオレンジ色で強調表示され、選択されていない頂点は黒く表示されます)。Object Modeでは、頂点を個別に選択・編集することはできません;オブジェクトは全体としてのみ変更することができます。⇨ Tab キーを押すことでこれらのモードを切り替えられます。現在のモードは、3D ウィンドウのヘッダに表示されます。



Naming Gus

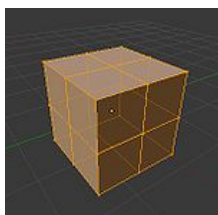
我々のジンジャーブレッドマンを"Gus"と呼ぶことにしましょう。そのために、object-contextに切り替えてください(*Naming Gus*参照)。標準レイアウトでは右側の下のほうにあるはずですが、最初の行にある通り、Gusが適切に命名されました。



Part of the Tool Shelf

私たちの最初の仕事は、キューブの頂点を動かして、Gusの胴体を構築することです。これを行うためのツールは3D Windowの(左側の) Tool Shelfにあります。もし Tool Shelfが見当たらなければ、シンプルに Tを押してみましょう。

さて Subdivide(Addの下あたり)の Tool Shelf ボタンのところへ行きそれを一度クリックします (*Part of the Tool Shelf*)。これにより、キューブの各辺が二つに分割され、新しい頂点と面が作成されます。結果は下図のとおりです。(同じような視点から見たければ、Num5で視点を変更し、MMB をクリックしたりドラッグしてビューを回転させましょう。-- Num1 (前面視点)と Num5 (斜行視点/直行視点)で視点を戻すのも忘れずに。)



カーソルを3Dウィンドウの上にもっていった状態で A を押すと、すべての要素の選択状態が解除されます。頂点は黒色になっているはずですが。

Box Select -- Selection Behaviors -- Limit Selection to Visible

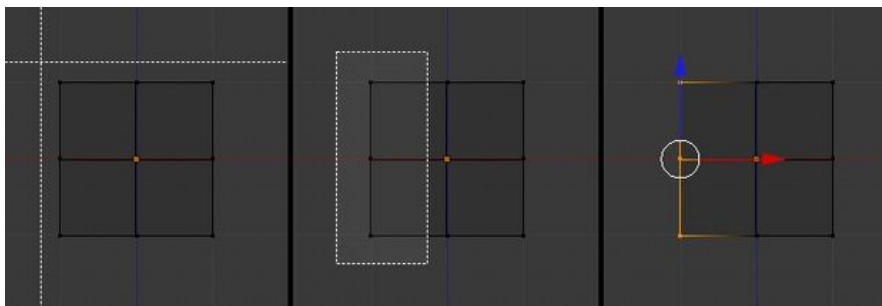
直行視点(5 NumPadで切り替え)では特に、頂点が他の頂点の背後に隠れてしまうことがあります。

我々の分割キューブは26個の頂点を持っていますが、それにもかかわらず直行正面視点ではわずか9個しか見ることができず、他は隠れてしまっています。

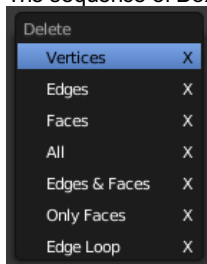
RMB をこれらの重なった頂点のうち1つだけを選択するのに対し、ボックス選択ではすべて選択します。これがあてはまるのは、Bounding Boxが Wireframe表示メソッドのいずれかがアクティブになっているときです。もしSolid か Texturedのメソッドが選ばれていたら、Limit Selection to Visible button を無効にする必要があり、そうしないと見えている頂点しか選択されません。(The Limit Selection to Visible button は、Solid か Textured いずれかの表示メソッドがアクティブになっていれば、3Dウィンドウのヘッダーに見つかるはずですが。)

このチュートリアルを進めるときは必ず Limit Selection to Visible button を *unselected* にしてください。

さて B を押しBox Select/Border Selectモードをアクティブにしましょう。カーソルが直交する点線に変わっているはずですが。カーソルをキューブの左上隅よりも上に移動させ、LMB を押したままに、マウスを下方方向と右方向にドラッグして、点線の長方形が左端のすべての頂点を包含するようにします。ここで LMB を離します (*The sequence of Box selecting a group of vertices*)。)







The sequence of Box selecting a group of vertices.



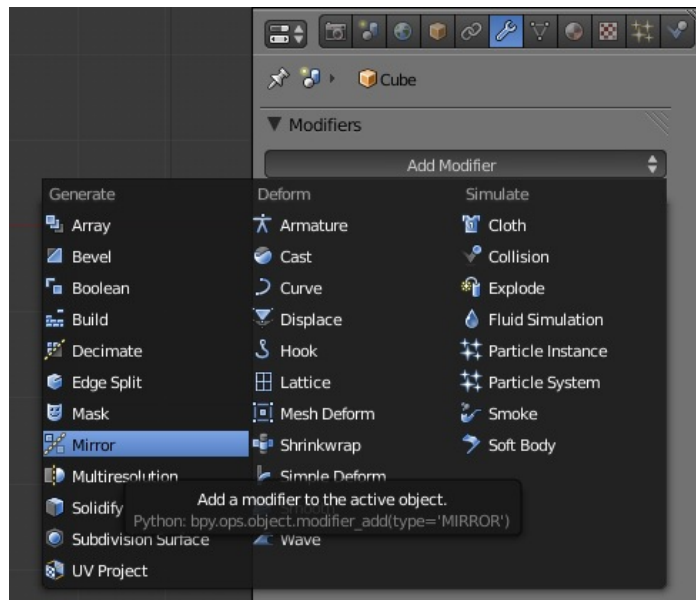
The pop-up menu of the Delete (X) action.

Xを押し、ポップアップメニューから Vertices を選んで、選択している頂点を消去しましょう (*The pop-up menu of the Delete (X) action*).{clr}}

Tip: 頂点の選択や解除をするツールには他にもCircle Selectがあります。Cキーを押すことでアクティブになります。Circle Selectツールがアクティブな状態で、LMB  でクリックかドラッグすると頂点が選択されます。MMB  は頂点の選択を解除します。マウスホイールを使うと選択円の大きさを変えられます。RMB  か  Enter で選択を確定しCircle Selectモードを終了します。上記のやりかたでの頂点の削除にも挑戦してみましょう。

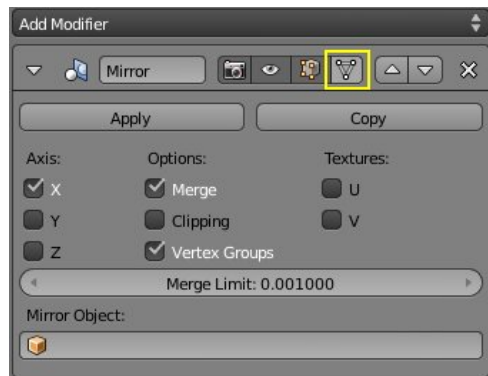
ミラーモデリング

オブジェクトを左右対称にモデリングするために、Mirror modifierを使用できます。Gusの片側をモデリングするだけで、Blenderが同時にもう片方を作成してくれます。Properties editorに移動して、Modifiers contextを見つけてください。 (*The modifiers context*)



The modifiers context

いまはまったくのからっぽです。Add Modifier と記されているボタンをクリックしてリストを開き、Mirrorを選択します。 (*The modifiers context*)



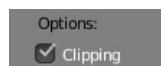
Cage Mode button.

何も起こらないように見えますが、これはmodifierが何を表示して何を表示しないのかということ、あとほんのちよっぴりの操作で指示してやる必要があるからです。我々のケースでは Cage Mode button をチェックして、Edit Modeでミラーされた面が透明に見えるようにします。 (*Cage Mode button*)

我々がキャラクターをモデリングしているサイドから、Blenderが補ってくれるサイドへと伸びる軸を、X,Y,Zいずれかのボタンをチェックして選択してください。ミラー面は、その軸に垂直になります。我々のケースでは、X軸にします。

The Merge Limits button (*Cage Mode button*参照) は安全ネットとして機能します。我々が設定する制限範囲よりもミラー面に近い頂点はどれも、ミラーの平面上にぴったりと配置されます。制限範囲は **0.000** から **1.000** の間で設定でき、どの程度の大きさにするかはその時々作業の性質や規模に依存します。

Gusのモデリングにおいては、ミラー面から **0.1** 単位ほど頂点が離れていれば気づけるかもしれませんが、それより近くなるとそうはいかないかもしれません。頂点が鏡面に接していない場合、メッシュが途中で引き裂かれてしまうおそれがあります。うっかり頂点を迷子にさせないように、Merge Limits を **0.1** に設定しておくべきでしょう。




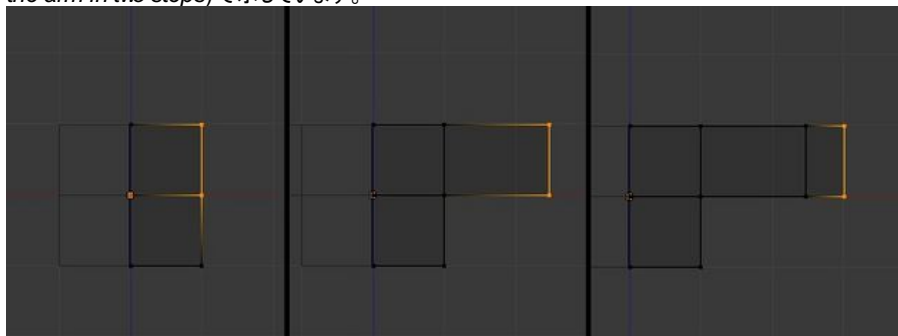
Clipping button.

最後に、Clipping button をチェックすると (*Clipping button*)、私たちのミラーは頂点が境界を横断できなくなります。もし横断できてしまうと、とても乱雑になりやすくなってしまいます。また、Clipping をアクティブにしておくと、ミラーの表面にある頂点はそこに固定されます。

ご覧のように、Mirror modifier は、多くの機能で私たちが生活をやすくしてくれます。

腕と脚

Gusの腕と脚を作成していきましょう。つい先ほど学んだ手順で、右側の上2つの頂点を Box Select しましょう (*Extruding the arm in two steps*)。実際には背後にある別の4つも選択されていて、頂点は合計で6つとなります。E を押してそれらの頂点を押し出していきます (もしくは Tool Shelf の Extrude Region button を使用)。こうして動かせる頂点と面が新しく作成され、それらはマウスで動かせます。それらを正方形1.5個ぶん右に移動させ、LMB  で位置を確定します。再度 E で押し出しを行い、新しい頂点を正方形の半分だけ右に動かします。この手順は (*Extruding the arm in two steps*) で示しています。



Extruding the arm in two steps.

アンドリドゥ

Blenderには2つのUndo機能があり、ひとつは Edit Mode のものでもうひとつは Object Mode のものです。

Edit Mode で取消しを行うには CtrlZ を押します。CtrlZ を押し続けるとバッファの限り取消を行い、変更をロールバックします。⇧ ShiftCtrlZ では変更をRe-doします。

(Macでは ⌘ Cmd の代わりに Ctrl を使用します。)


次の2つの点に注意:

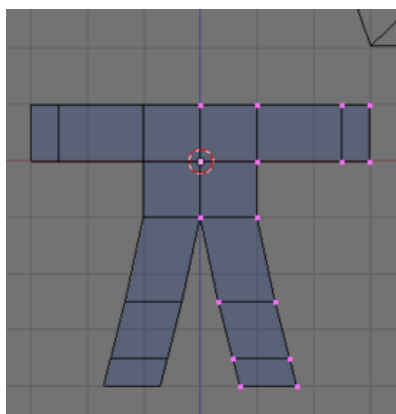
- Edit Mode での取消しは、その時のモードでのオブジェクトに対してのみ動作します。
- UNDOデータは Edit Modeの外に切り替えても失われませんが、それはEdit Modeで 別のオブジェクトの編集を始める前までに限ります。

Object Mode でも同じショートカットが適用されます。CtrlZ でundo、⇧ ShiftCtrlZでredoです。Edit Mode でオブジェクトに加えた変更が消えることなく、CtrlZ 一発でその回の段階でのすべてが一度に取り消されます。

行動の途中で気が変わってキャンセルしたいときは、多くの場合 Esc か RMB  を押すだけで直前の状態にすぐ戻れます。



頂点の一括移動

押し出しの加工を加えるため、まず新しい頂点を作成し、そしてそれらを移動させます。移動の途中で気が変わったらEsc か RMB  を押せばキャンセルされます。新しい頂点は元の位置に、初期位置にもどります! 押し出しを始める前の状態に戻るには、Undo (CtrlZ)を用いるのが最もシンプルな方法でしょう。このような意図で新規頂点の作成や移動をするには、G や S や Rでの拡大縮小や回転も役に立つときがあります。



Body.

現在Gusには、あなたがモデリングした左腕(彼は僕らと向かい合わせです)と、Blenderが追加した右腕があります。同じように下の頂点を3回押し出し、左脚を造ります。(Body)で示しているのとおなじように作成してみましょう。Extrude - Regionを用いる場合は、一時的にMirroring modifierのAxisの下にあるxオプションのチェックを外し、押し出し後に再びチェックしましょう。(そうしないとGusくんはズボンでなくスカートをはく羽目になってしまいます。)

押し出した頂点の移動時、E を押してからLMB  をクリックするまでの間にMMB  をクリックすると任意の方向に押し出せるようになります。こうしないと足がまっすぐ下にのびてしまいます。(Body)にあるように下だけでなく横へも動かしましょう。

Tip: 正確に配置したいときは、Ctrl を押しっぱなしにして周りからぐるっと見回してみましょう。


ミラーでのモデリング作業はここまでです。次のステップでは他のテクニクを試みてみましょう。その前にモデルの右側を *real* させておく必要があります。modifierは、変更を *apply*しないと永久に何もしてくれないからです。Gusを Object Modelにして (まだ Edit Mode であれば ⇧ Tab を押して)、

Mirror modifierの *Apply* button ボタンをクリックしましょう。

頭部

ガスには頭が必要です。



Edit Modeに戻しましょう (⇐ Tabを押します)

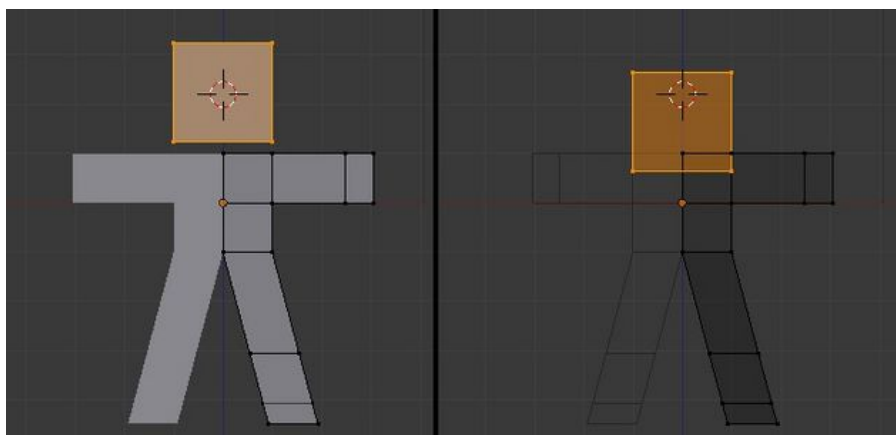
Gusの体から正方形ちょうど1つぶん上の位置にカーソルを移動します。カーソルを特定の格子点に配置するには、LMB  で移動させたい場所の近くにカーソルを置き、その後 ⇧ ShiftS で Snap Menuをよびだします。Cursor to Gridでカーソルは正確に格子点に配置されます。そうこれが今まさに欲しいものです。Cursor to Selectionは選択しているオブジェクトの位置に正確に配置するもので、これも便利です。

Gusの頭として新しくキューブを追加しましょう(⇧ ShiftA >> Add >> Cube) (*Adding the head*の左端の画像)。

Object Creation

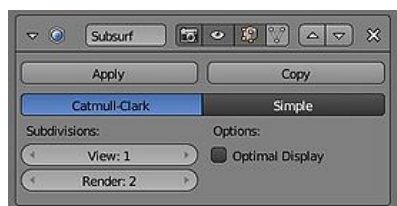
Edit Mode に別のオブジェクトがあるときにオブジェクトを追加すると、新しいオブジェクトは、既存オブジェクトの一部になります。したがって編集モードでガスの体に追加されたこのキューブは、自動的に彼の体の一部となります。

ここで G を押して Grab Mode に切り替え、新しく作成されたキューブを下の方へ下げます。頭を下のほうへ移動する際、MMB  のクリックで移動を直線状に制限することができます。Gusの新しい頭を格子の三分の一ほど下げたら、LMB  を押して位置を確定します (*Adding the head*の右端の画像)。



Adding the head.

サブサーフェス (サブサーフ)



The Subsurf modifier

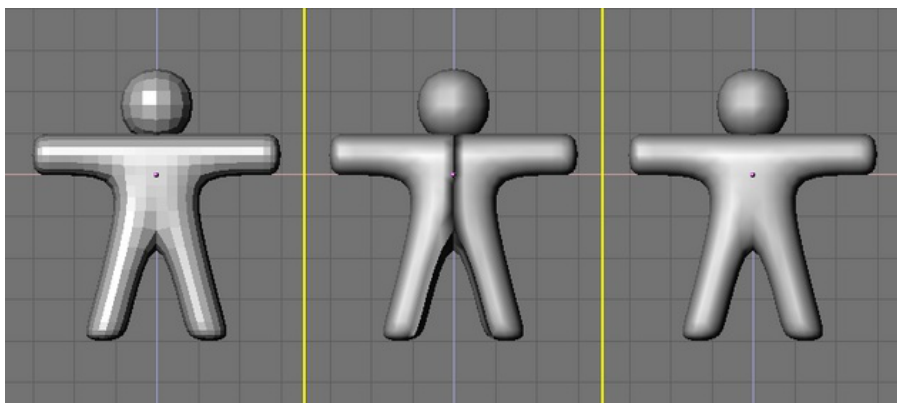
次のステップでは、彼の頭だけではなく、Gusのすべてを選択する必要があります (use A - maybe twice)。

これまで私たちが制作してきた形状は、最高に粗い状態です。滑らかにするために、Modifier のところにある Subdivision Surface modifierを追加しましょう (*The Subsurf modifier*)。View と Render の数値ボタン (Subdivisionsの下あたりにある) の両方が2かそれより下の数値になっているか確認してください。View では3D表示域の細分化のレベルを設定します。Render ではレンダラーで使用される細分化のレベルを設定します。

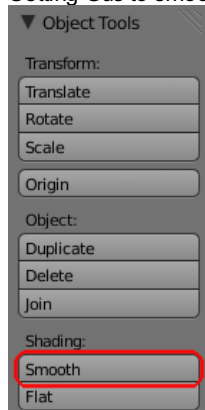
SubSurfaces

SubSurfaceは粗いメッシュを動的に磨き上げてくれる、先進的なモデリングツールです。密度の高いメッシュを作成し、この細かいメッシュに頂点を配置することによって動作し、それに従って元の粗いメッシュをなめらかにしていきます。オブジェクトの形状を粗いメッシュの頂点の位置によって制御する点は変わりませんが、レンダリングされた形状は細かく滑らかなメッシュになります。

Gusの姿を見るために、Edit Mode を抜け(⇐ Tab)、Zを使用して Solid display mode にします(アクティブになっていなかった場合)。(Setting Gus to smooth)のようにになっているはずで




Setting Gus to smooth

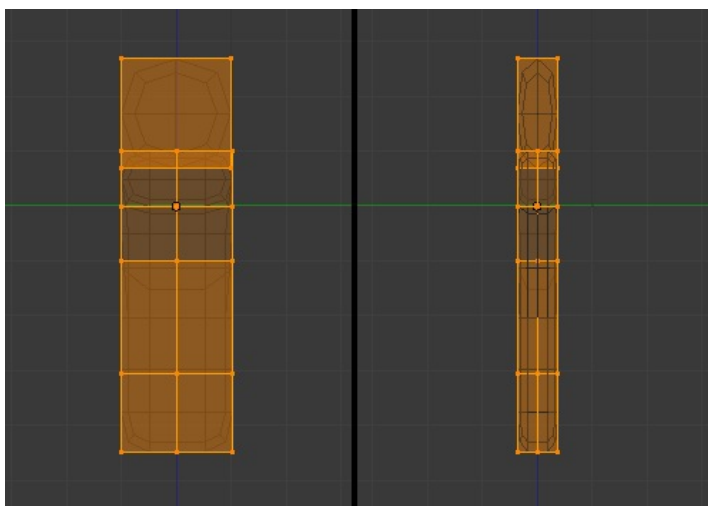


Object Tools.

Gusの外見をすべすべなめらかにするには、3DウィンドウのTool Shelf (T)のShadingのもとにあるSmooth button を押します。Gusがなめらかになって現れますが、けれども彼の中央部になにかおかしい黒い線が出ているかもしれません。この現象は Mirror modifierを使っても多くの場合避けられるのですが、Mirrorを追加する前に押し出しや反転が行われていると起こることがあります (*Setting Gus to smooth.*、中央)。これらの線が現れるのは、SubSurfの細かいメッシュが、粗いメッシュの法線方向(面に垂直な方向)に関する情報を使用して計算されるため、一部の点が誤った方向を向いてしまうことがあるからです(いくつかの面の法線は外向きを指し、あとのいくつかは内側を指すというぐあい)。法線をリセットするには、Edit Mode (⇐ Tab)に戻し、すべての頂点を選択し (A)、そして Ctrl+Nを押します。こうしてGusは見事になめらかになりました (*Setting Gus to smooth.*、右)。

MMB  を押してドラッグしながらGusをすべての角度からぐるっと見回してみましょう。おっと、彼は厚すぎますね！

縮尺の拘束



Slimming Gus using constrained scaling

Gusを薄くしていこう:



Parameters of the last action in the Tool Shelf

- Edit Mode になっていなければ(⇔ Tabで)切り替え、それから Wireframe mode に戻します。(Z)、Num3 をつかってサイドビューに切り替え、Aですべての頂点を選択します。以降のステップは Object Modeでも同じようにできますので、お好きなほうどうぞ。
- S を押してマウスの水平移動をスタートさせます。(MMB クリックでスケーリングをひとつの軸だけに制約できます。また、Yを押しても同様の結果が得られます)。ここでマウスをGusのほうへ動かすと彼が薄くなっていきますが、身長はそのまま変わらずとのままになります。
- 3DWindow toolbar のヘッダには倍率が表示されます。Ctrlを押したままにします: 拡大縮小の倍率が、とびとびの0.1ステップごとに変わります。Gusを縮小して係数が0.2になったら、LMB して寸法を確定します。その最後の変形で間違ってしまったても、まだパラメータを変更することができます。それらは Tool Shelf の下部に表示されます (*Parameters of the last action in the Tool Shelf*参照)。
- 正面視点に戻り (Num1)、Solid mode(Z)にして、MMB で視点を回転させましょう。Gusはさらにそれらしくなりましたね!

Let's see what Gus looks like

私たちの最初のレンダリングにお目にかかる準備ができましたが、その前にいくつかちょっとした作業をしておきましょう。

- Object Mode でない場合は切り替えておきます (⇔ Tab)。

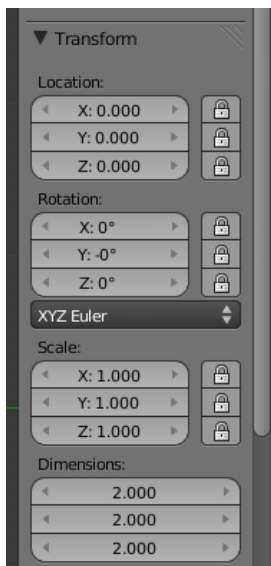


Making both layer 1 and 10 visible.

- 3DWindowツールのlayer visibilityボタンの上段右端の小ボタン(*Making both layer 1 and 10 visible.*)を⇧ Shift LMB して、レイヤー1(Gusのレイヤー)とレイヤー10(カメラと照明があるレイヤー)をvisibleにしておきます。

A Tip

最後に選択されたレイヤーはアクティブになるので、それ以降の追加はすべて自動的にレイヤー10にされることを覚えておきましょう。

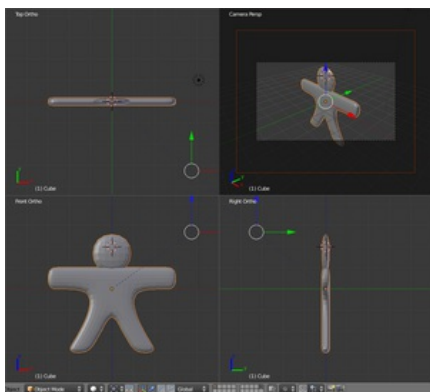


The Transform Panel

- Nを押して Properties Shelf をよびだし、Transform パネルをさがしましょう (*The Transform Panel*)。位置情報は、X、Y、Zの値によって指定されます。
- (RMB で)カメラを選択し、(x=7, y=-10, z=7)のような場所に移動させます。G を押してからカメラをドラッグしてもこれを行えます。視点を変えて、2回カメラを移動し、3つすべての座標を調整する必要があります。オブジェクトの場所を数値で入力するほうがお好みでしたら、数値を LMB すればお望みの値を入力できます。

カメラのセットアップ

カメラがGusのほうを向くようにするには、カメラを選択状態にしたまま続けて \diamond Shift RMB でGusも選択します。カメラは暗いオレンジ色(選択状態)になり、Gusはとても明るいオレンジ(選択かつアクティブ)になっているはずですが。ここで CtrlT を押し、ポップアップにある TrackTo Constraint entry を選びましょう。これにより、カメラがGusを追跡し常に彼を向くよう強制します。これで、あなたがカメラをどこに移動させても、Gusは常に必ずカメラの視野の中心になります。



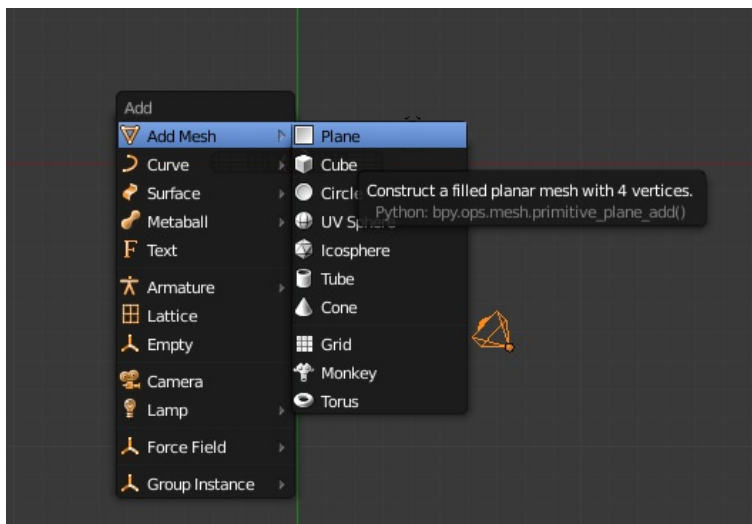
Camera position with respect to Gus.

(*Camera position with respect to Gus*) には、Gusの上面、前面、側面およびカメラビューをお見せしています。カメラビューにするには、0 NumPad かもしくは View>>Camera のようにします。画像でお見せしているような Quad View にするには、CtrlAltQ かもしくは View>>Quad View と選びましょう。

地面

Gusが上に立つための地面を作成する必要があります。

- 上面視点(7 NumPad または View>>Top)で、かつオブジェクトモードで、平面の追加をします
- 平面を追加しましょう (\diamond ShiftA か >>Add>>Mesh>>Plane)。

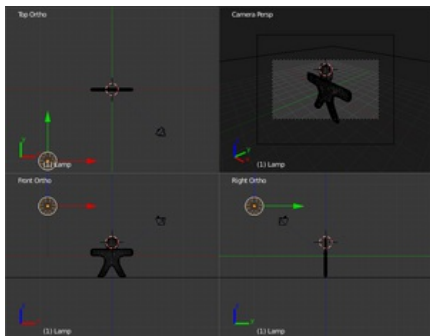


Note
 Edit Modeから出ておくことは重要です。そうしないとGusの頭を追加したときのように、新しく追加したオブジェクトが、その時のEdit Modeで編集中のオブジェクトの一部になってしまいます。

- 正面視点に切り替え(1 NumPad か View>>Front)、平面をGusの足の下まで移動 (G) させ、Ctrl を使ってGusとそろるようにします。
- カメラ視点にして (0 NumPad か View>>Camera)、平面が選択されている状態で、S を押してスケーリングを開始します。
- 平面を拡大して端がカメラの視野よりも外側にくるようにします。視野はカメラビューで明るく表示されています。

照明

さあ、光を加えていきましょう!



Inserting a Lamp

- 上面視点 (7 NumPad)で、いまあるランプ光源 (ランプが舞台にないときは \diamond ShiftA >>Add>>Lamp>>Lamp で追加できます) をGusの正面に移動させます。ですがこのときカメラよりも向こう側の位置になるようにしてください。例えば (x= -9, y= -10, z=7)のような位置に (*Inserting a Lamp*.)。



The object data button of a lamp

- オブジェクトモードでランプを選択して、[プロパティ]ウィンドウのオブジェクトデータのアイコンを選びましょう(小さい太陽のような見た目です)。ランプのサブメニューを見ると、Point, Sun, Spot, Hemi, Area choicesなどの項目があるでしょう。 (*The object data button of a lamp*).



The Spot light settings

- プロパティウィンドウの Lamp パネルで、ランプをスポットライトに変えるため Spot トグルボタンを押し (*The Spot light settings.*)、白いボタンをクリックして淡い黄色 (R=1, G=1, B=0.9) にします。実はカラーピッカーなんです。Spot Shape で Size を 40 ほどに調整し、Blend: は 1.0 にします。
- スポットライトが Gus を追いかけるようにするには、カメラでやったときと同様に、スポットを選択して \diamond Shift を押しながら Gus を選択、そして CtrlT >> Track To Constraint と押します。
- 第2のランプとして、全体をより均一に照らす光源を追加しましょう (\diamond ShiftA >> Add >> Lamp >> Hemi)。Energy の設定は 0.2 にします (*Hemi lamp settings*)。それをカメラの少し上に移動させ (x= 7, y= -10, z=9)、前にやったように Gus を追従するように設定します。

Two lamps?

2つかそれ以上のランプを使用すると、ソフトでリアルな照明の演出に役立ちます。というのも、現実の自然光がひとつの点から来ることはないからです。

レンダリング



The Render context button

レンダリングする準備がほとんど整いました。最初のステップとして、Properties ウィンドウのヘッダーにあるレンダーコンテキストボタンを押します (*The Render context button*)。



The Render context

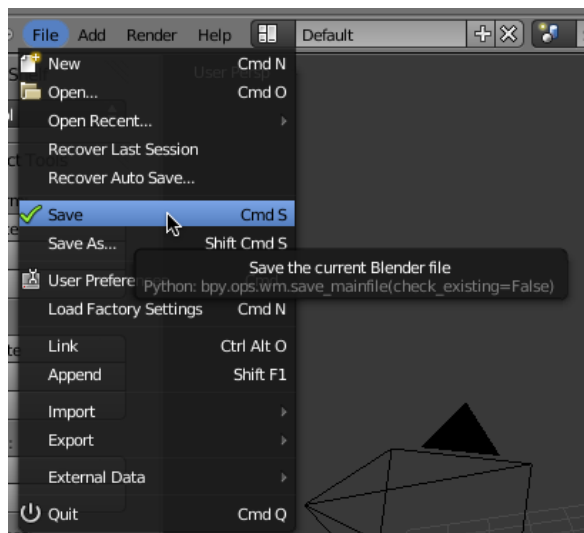
デフォルトのレンダリング設定を使用します。(The Render context)に示しています。

さあ Image button か F12を押しましょう。結果は、(Your first rendering. Congratulations!)のようになりますが、まだまだとても貧相です。材質の設定や、目などの様々なディテールを作っていく必要があります。



Your first rendering. Congratulations!

作品の保存



The Save menu

まだしてないのであれば、今は作業を保存するのにつけてのタイミングでしょう。 *The Save menu*. で示しているように、メニューからFile>>Save とするか、CtrlSで行えます。既存のファイルに上書きする場合、Blenderは警告を出します。

Blenderはあなたのシステムの一時ディレクトリへの保存も自動的にしてくれます。デフォルトでは、これは5分ごとに行われ、ファイル名は数字です。これらの保存の読み込みは、望まない変更を元に戻すもう一つの方法です。

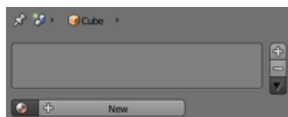
マテリアルとテクスチャ

ここからは、Gusをクッキーのような素敵な材質にいきましょう。



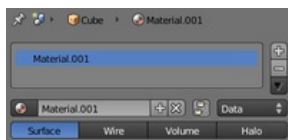
The Material context Button.

- Gusを選択。次に、[プロパティ]ウィンドウヘッダの中から、the Materials button (*The Material context Button.*)を選んで、Material パネルを呼び出します。



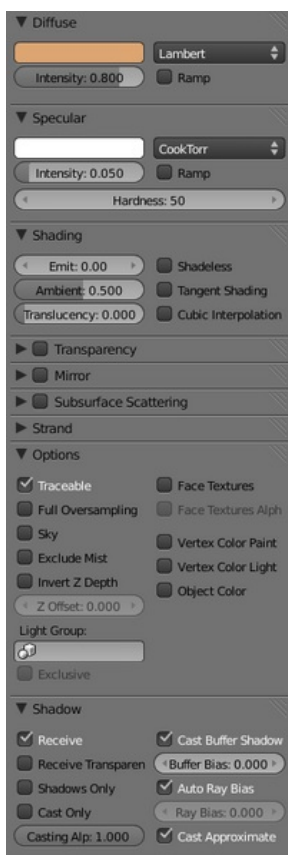
The (almost) empty Material context

- [プロパティ]ウィンドウはまだからっぽなので、ガスにはまだマテリアルが設定されていないことになります。マテリアルを追加するため、Material パネルの + Newbuttonを押します (*The (almost) empty Material context*)。



Top of the filled material context

- [プロパティ]ウィンドウはパネルとボタンによって構成されています。マテリアル名をもった一行、多くの場合"Material.001"、がリストのワクの中に現れます。ワク内で一意のデータブロックIDがつきます。下のほうにある名前をクリックして (*Top of the filled material context*の下部)、なにかもうすこし意味のある名前に変えましょう。"GingerBread"のような名前に(引用符はタイプしないように)。



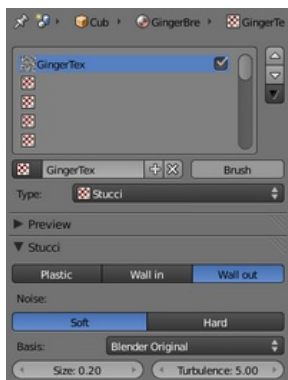
A first gingerbread material.

- 大まかな最初のmaterialを得るため、デフォルトの値を (*A first gingerbread material*) のように変更します。場合によりヘッダーの横の小さい三角形を LMB  クリックしてパネルを広げる必要があることに気をつけてください。



The Texture context button

- Properties ウィンドウのヘッダーの Texture context Button を押して (*The Texture context button*)、Add newを選びます。最初のチャンネルにテクスチャを追加します。"GingerTex"と呼ぶことにしましょう。



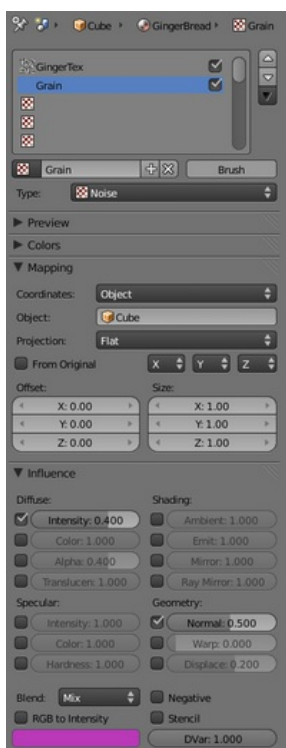
The Texture context

- Type を Clouds から Stucci に変えて、すべてのパラメータを (*The Texture context*)とおなじように設定します。



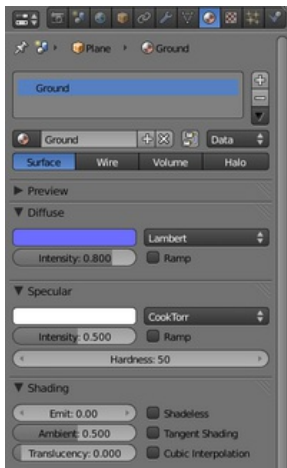
Settings for the Stucci texture

- Texture context の Mapping と Influence のパネルを (*Settings for the Stucci texture*) とおなじように設定します: Color のチェックボックスをはずし、Normal のチェックボックスをセットしたら、Normal のスライダーを **0.75**に変更します。これらの変更によってStucciテクスチャーは"パンブマップ"として振る舞い、Gusをよりビスケットのように見せてくれます。



Settings for an additional Noise texture

- さて Textureコンテキストのテクスチャリスト内の2行目を選択し、第2のテクスチャを追加します。それを"Grain"と名付けて、(*Settings for an additional Noise texture*) に合わせて調整します。テクスチャ自体はプレーンな Noise テクスチャです。



A very simple ground material


- 地面をそれにふさわしいマテリアル、(A very simple ground material)のような暗い青色にします。お好みの色合いの青をご自由にご自由に選んでください。

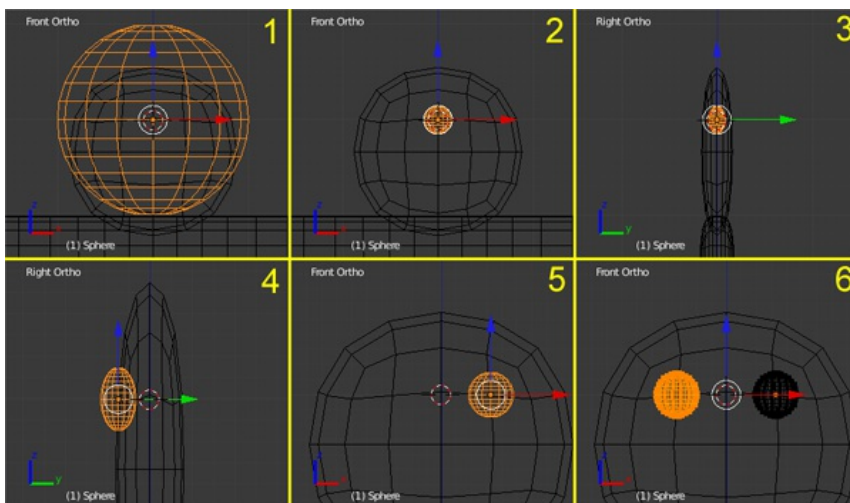
目とディテール



仕上げとして目やディテールを追加していきましょう。



Layer visibility

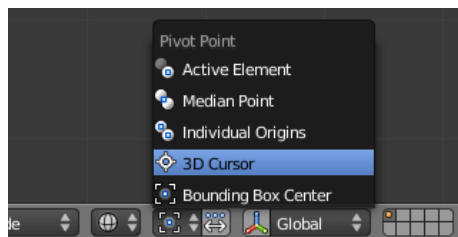
- まず最初に、LMB  クリックでレイヤ1ひとつだけを可視状態にします (Layer visibility)。これにより、ランプ、カメラ、地面は非表示になります。
- カーソルをGusの頭の中心に置きます。(あなたは3Dにいるのですから、残りの2方向のビューを確認することもお忘れなく!)
- Object Modeで球体を追加します (⇧ ShiftA >>ADD>>Mesh>>UVsphere)。F6 を押し、segments (meridians) の数字を 16 に変更します。結果は *Creation of the eyes*の1でお見せしているようになります。
- (S) で球体を縮小していき、factor で直径 **0.15** ほどにします。それから側面視点に切り替え (3 NumPad)、水平方向にだけ (Y) 縮小を **0.05**になるまでしていきます (*Sequence for creation of the eyes*の2と3も参照)。



- 必要なら + NumPad, Wheel , Ctrl MMB  などで少しズームし、球体をドラッグ (G) して左にもっていき半分だけが頭部にめりこむようにし、*Creation of the eyes*の画像4のようにします。
- 正面視点に戻り (1 NumPad) 球体を横に移動させ、右にもっていきます。Gusの目のあるべき位置に設置しましょう (*Creation of the eyes*の画像5)。

Flipping a duplicate around the cursor

- Edit Modeに切替えます (⇔ Tab)。3Dウィンドウのヘッダーから、十字線のはいったpivotボタン (*pivot: 3D Cursor*)を選びます (the 3D Transforms Manipulator jumps from the sphere to the cursor)。目のすべての頂点を必ず選択し (していない場合は A ですべてを選択)、この状態で ⇧ ShiftD を押すか Tool ShelfのDuplicate を使って複製をします。Esc でマウスによるコピーの設置をストップできます。



Pivot: 3D Cursor

- CtrlM を押してミラーし、X でミラーをX軸まわりにしたら、つづいて LMB か ↵ Enter でミラーを確定します。pivotボタンはデフォルトに戻しておきましょう (Median Point)。結果は *Creation of the eyes*の画像 6のようになります。

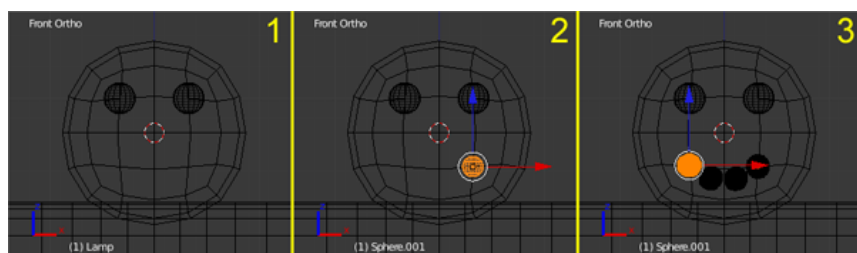
Mirroring

ミラーリングはオブジェクトモードでも可能で CtrlMを使用して行います。

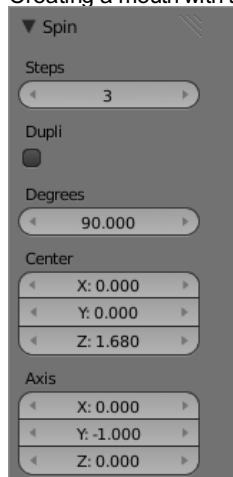
これでGusの目は2つになりました。

□

- Edit Modeから出て (⇔ Tab)、カーソルをGusの中心にできるだけ近い位置に置きましょう (⇧ ShiftS キーを思い出して)。新しい球を追加し、拡大縮小や移動はまえに目でやったのと同じようにします。ただしスケールだけは小さく (0.15ではなく0.1に) します。それをカーソルの右下に置き、カーソルは SubSurfed meshの頂点の中心に合わせ、*Creating a mouth with the Spin tool*の画像2のようにします。



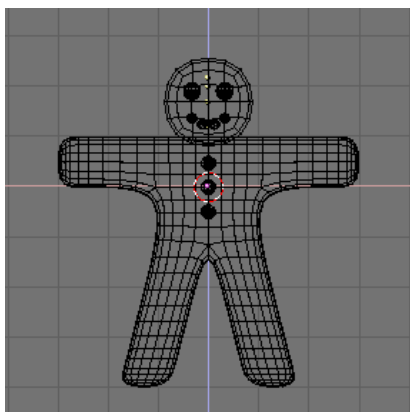
Creating a mouth with the Spin tool



The Spin Tools options on the Tool Shelf

- Edit Mode に切り替えます (⇔ Tab)。ここで AltRを使うかまたは Tool ShelfのSpin を LMB クリックします。球のコピーがいくつか出現します。
- Tool Shelfの下部で、(visibleでないときはF6を押します)spinの詳細設定をします: Degr: を 90に、Steps: を 3 に設定します。結果は、*Creating a mouth with the Spin tool*の画像3のように、Gusの口になっているはずで

(EDIT REMARK: プロパティ内の最後のコマンド (上記の *Spin* のような) は必ずしも3Dウィンドウにあるとは限りません (少なくとも最新のバージョンの筆者のBlenderでは (V2.5 beta, Mac)) ので、混乱なさらならないように。-どこかにあるはず (少なくともデフォルトスクリーン内に))



The complete Gus!

- さてオブジェクトモードに戻って、球をもう3つ追加して(頭の下かつZ軸に中心が沿うように)Gusの服にボタンをつくっていきます。ボタンが1つできたら、Edit Mode から出て、 \diamond ShiftD で複製をつくり、そして *The complete Gus!*で示したような位置に複製を置きましょう。

Attaching the spheres

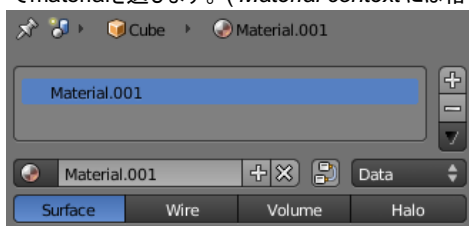
Gus全体をひとかたまりとしてつかんだり動き回したりしたければ(チュートリアル2部のアニメーションでも役立ちます)、目、口、ボタンとなる小さな球を体とくくりつける必要があります。オブジェクトモードに入り、何も選択されていない状態でAを押します。ここで1つの球を右クリックします(複数の場合はグループとして選択され、それでもokです)。 \diamond Shiftを押しながら、体を選択してください。そしてCtrlPと打ち、ポップアップのObjectを左クリックします。すべての選択を解除し、各要素に取り付けプロセスを繰り返します。

目のマテリアル

目をチョコレートのような質感にしていきます。口は白い砂糖のような質感にし、Gusの服のボタンにも少しこの色を使いましょう。Gus自体に質感を与えたときのことを思い出し、ほんの少しの創造性で - まもなくGusはジンジャーブレッドファッションという狭き門の頂点を目にすることでしょう。

Objects sharing a material

ひとつのオブジェクトに適用した材質を他のオブジェクトにも適用するには(言い換えれば口の白い質感をボタンのひとつにも再使用するには)、Properties window のMaterial context の Data Block ID Nameの横にあるBrowse ID Data のボタンを押すと現れる**Material Menu list**でmaterialを選びます。(*Material context* には格子模様の小さい球があり、お求めの *Material.001* はその横に表示されます)。



レンダリング

質感の割り当てが完了したら、レイヤー**10**を再び可視にしておきます(どうやるのか覚えています? ヒント、3Dウィンドウヘッダーを見てみましょう)。そして照明とカメラを出現させておき、(F12)で新しくレンダリングをしましょう。

結果はだいたい (*The complete Gus still rendering*)のようになります。



The complete Gus still rendering.

保存

レンダリング結果が表示されているUV/Image Editorのところで F3 を使って画像を保存しておきましょう。ファイルウィンドウで画像の名前を入力し、保存先を選択して保存します。

画像のフォーマットを選ぶことができ (JPEG, PNG, など) File Browserの右のshelfで設定します。

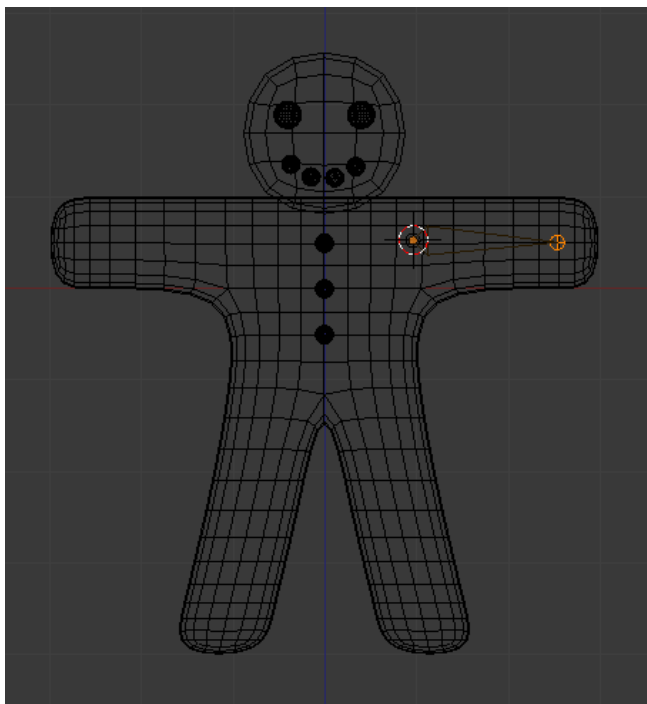
Blenderはファイル名に拡張子を追加します。; あなたもたぶんおなじみのように (これはV2.5での新要素です。V2.4xではあなたが手動でしなければなりません)。

30プラス30分でできる初めてのアニメーション Part II

静止画を求めているのであれば、ここまでの私たちの作業で十分なのですが、しかし我々はGusを動かしたいのです！次のステップでは彼に骨格やアーマチュアなどの、動かすのに必要なものをつけていきましょう。これは美術分野ではリギングと呼ばれています。GUSがもっているのは非常にシンプルなりギング:四肢(二本の腕と二本の脚)といくつかの関節(肘はなく、膝のみ)で、足先や手はありません。

リギング

リギングの追加:



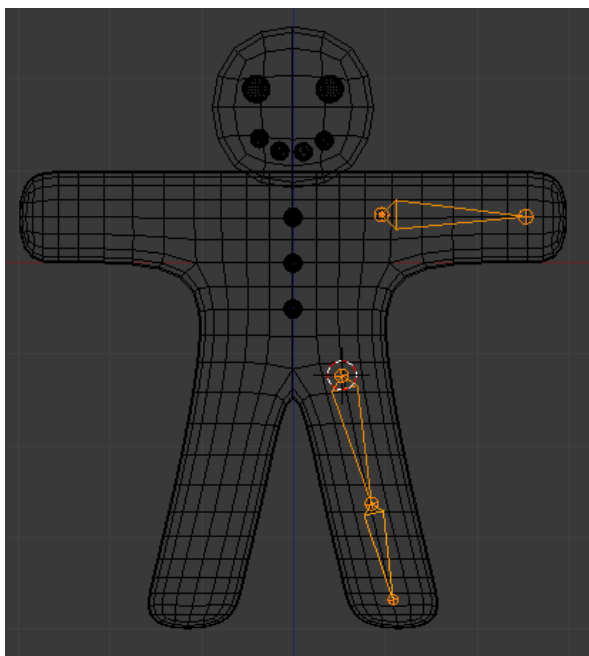
Adding the first bone, an elbowless arm.

- Object modeで、カーソルをGusの肩の位置に置き、**⇧ ShiftA >> Add >> Armature >> Single bone** と押します。菱形のオブジェクトが出現し、それがアーマチュアシステムの骨格となります。Edit mode に入ってください。骨のend末端、または *tip*先端が選択されます(黄色)。
- さて Edit modeで、(G) でボーンをつかんで動かし、その先端をGusの手のおきましよう、(Adding the first bone, an elbowless arm)。他のボーンは今必要ではありません。これで、肩から手の範囲をつらぬくボーンを得ました。先端を移動すると、ボーンが大きくなることに気づくでしょう - 実際ボーンはスケールアップされます。

プロセスをより効果的にすすめるために、ジンジャーブレッドマンとアーマチュアをいろいろな方向から定期的に見て、アーマチュアがジンジャーブレッドマンの内部にあるかを確認するようにしましょう。ちょうど骨が人体の内側にあるのと同じように。ボーンが、例えば体の前方や後方にあると、スキニングで失敗してしまうでしょう。多くの異なる視点から点検することは、3Dモデルを作成するうえで共通のテクニックです。

About Bones' Ends

ボーンの両端は、いろいろな名前ではばれます。Blenderでは、いまのところ"head"/"tail"と呼ばれています(前者が"太い"端部で、後者が"細い"端部です)。しかしながら歴史的には、"root" / "tip"という名称もあり、そのほうがいくらか混乱が少ないと考えられることもあって...



Adding the second and third bones, a leg bone chain.

- Edit modeのまま、股関節の位置にカーソルを移動し、 \diamond ShiftAで新しくボーンを追加します。
- (G)でつかみ、新しいボーンの黄色いウチバシが膝のあたりにくるように動かします。
- さて足の領域でCtrl LMB $\left[\text{LMB} \right]$ して、新しいボーンを膝から足に“chain”させていきましょう。新しくチェーンされたボーンは、膝から足のさきっぽまでが自動的につながった状態で現れます (*Adding the second and third bones, a leg bone chain*)。新しいボーンをチェーンにするもう一つの方法は、E ショートカットを使用して押し出しをすることです。このパリエーションで新しい骨を作成すると、あなたは自動的にGrabモードに移されます。LMB $\left[\text{LMB} \right]$ クリックして、骨の先端の現在の位置を有効にしましょう。

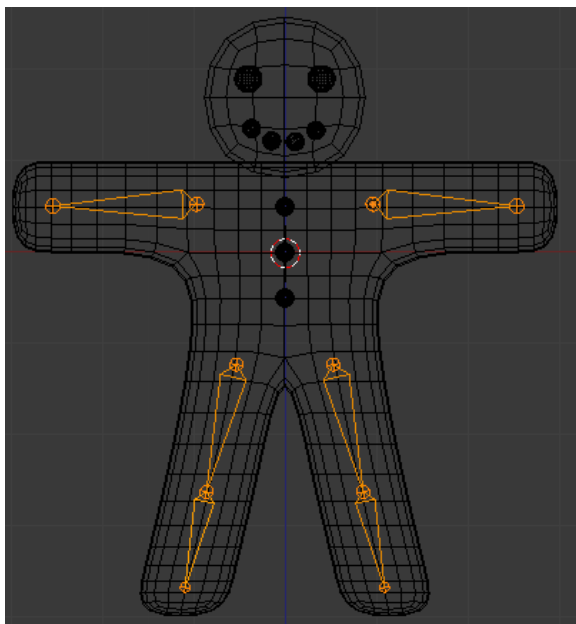
Bone position

我々がボーン追加すると、Gusの体のメッシュはゆがみます。きちんとした結果を生成するには、図に示すように、骨の関節を配置してみてください。

Bone roll

ボーンを (*Adding the second and third bones, a leg bone chain*) のように整列させなければ、下腿のボーンを選択して CtrlN, 3 を押し、Bone Rollを調整する必要があります。

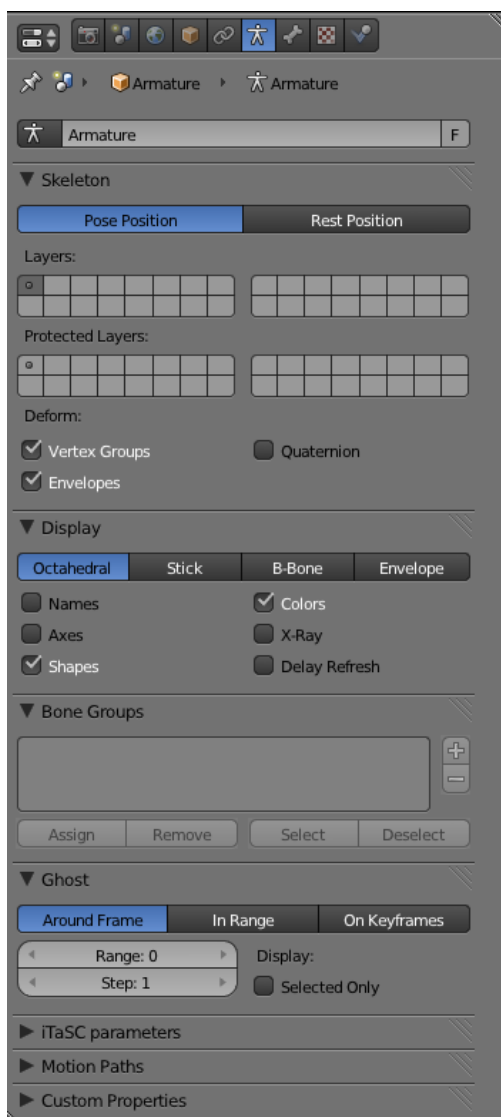
我々は現在、Gusのアーマチュアを構成する3つのボーンを持っています。



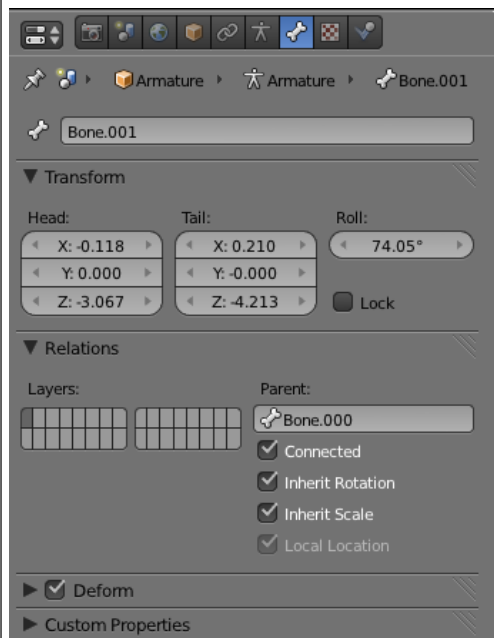
The complete armature after duplicating and flipping.

- では中央にカーソルを置き(\diamond ShiftC)、すべてのボーンを選択します A。 \diamond ShiftD でそれらを複製したら、すぐに EscでGrabモードを終了します。カーソルが回転/拡大縮小ピボットとして選択されていることを確認します (3Dウィンドウのヘッダー内の Pivot ドロップダウンリスト)。CtrlM 続いて Xで、複製されたボーンをX軸周りに反転させます。LMB $\left[\text{LMB} \right]$ クリックしてミラー化の操作を確定します。完了すると (*The complete armature after duplicating and flipping*)のようになります。

いずれかのボーンが選択されているとき、Object Data context にはアーマチュア全体の設定が表示され、名前も Armature Object と同じになってアーマチュア表示のためのセッティングになり、Object Data にはアクティブな Bone およびボーン固有の設定が表示されます。



Armature context.



Armature Bones context.

Names チェックボックスにチェックをいれて (Armature context, Display panel)、3Dビュー内のボーンの名前を見えるようにし、それぞれのボーンを選択して、Bones contextのボーンの名前を LMB クリックし、(Armature Bones context)をみながら、ボーンの名前を Arm.R, Arm.L, UpLeg.R, LoLeg.R, UpLeg.L, LoLeg.L のようないくらか相応しいものに変えましょう。⇔ TabでEdit mode から抜けておきます。

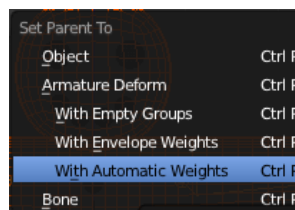
Naming Bones

ボーンに名前をつける際、左右を区別するために末尾に ".L" や ".R" とつけることはとても重要です。そうしておけば Action Editor で自動的にポーズ反転できるようになるようになります。

スキニング

さて、私たちは、アーマチュアの変形と、それが引き起こすボディの変形を一致させていく必要があります。skinningでこれを行います。頂点をボーンに割り当て、前者は後者の動きに従属します。

- Object mode でGusの体を選択し、続けて **Shift** でアーマチュアも選択すると、ボディは濃いオレンジ色になりアーマチュアは明るいオレンジ色になります。
- さて、私たちは、ボディとアーマチュアを親子関係にする必要があります。これは **Ctrl P** を押すことによって実現できます。(Parenting menu) が表示されるでしょう。Armature Deform >> With Automatic Weights と選択しましょう。

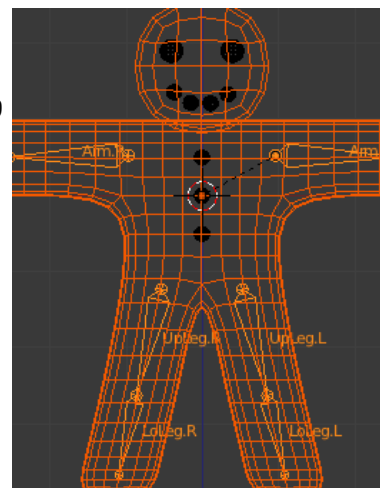


Parenting menu.

Vertex Groups, Envelopes and the order of modifiers

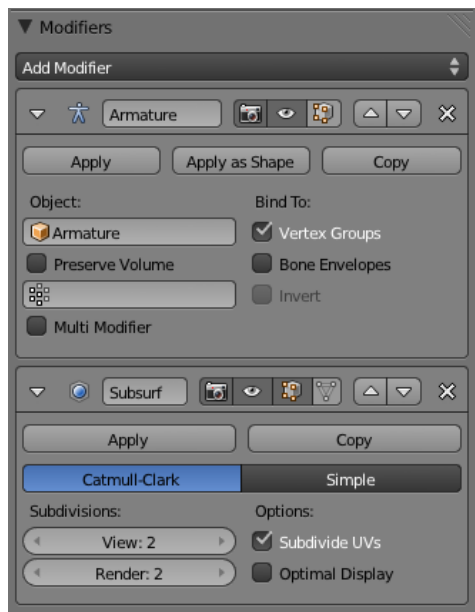
メッシュをスキニングするとき、メッシュオブジェクトは、そのモディファイアのスタックのいちばん下に加えられた Armature modifier を取得します。Gusのメッシュはすでにスタック上に Subdivision Surface

modifier を持っています。メッシュを滑らかに変形させるためには、Armature modifier を Subdivision Surface modifier よりも上の位置に移動させる必要があります。これは Modifier context の Move Modifier buttons をクリックすることで行えます (*The modifier stack in the Modifiers context*)。また、それぞれのボーンは、エンベロープと呼ばれる、影響力のおよぶ領域を持ちます。アーマチュアは、割り当てられた頂点グループとボーンのエンベロープの両方のメッシュを変形させます。これは望ましくない結果につながることもあり、我々の場合には、Armature Modifier の Bone Envelopes を無効にしておくことが重要です (*The modifier stack in the Modifiers context*)。

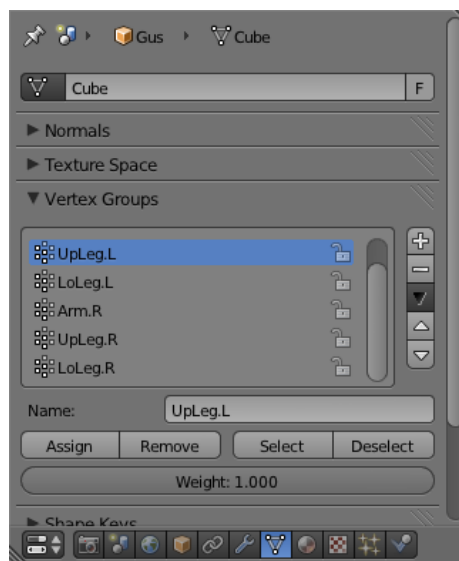


Armature parented.

- さてGusの体を選択し、Edit mode に切り替えます (⇨ Tab)。Object Data context には、Vertex Groups をまとめて管理できる Vertex Groups panel があることもお知らせしておきます (*The vertex groups controls in the Object Data context*)。

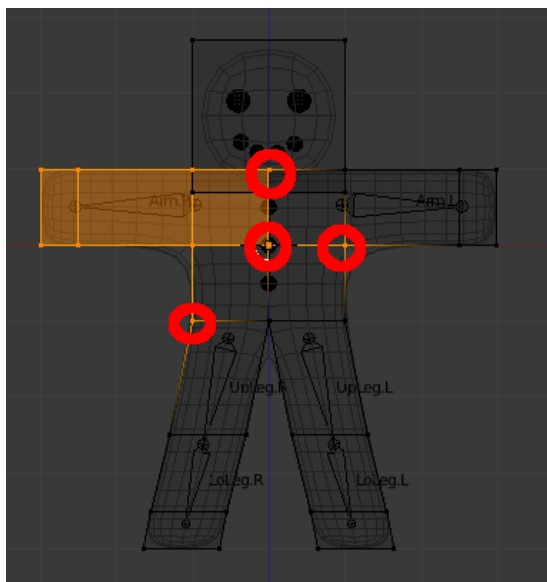


The modifier stack in the Modifiers context.



The vertex groups controls in the Object Data context.

頂点グループのパネルでスクロールをすると、使用可能なすべての頂点のグループを見ることができます – 我々の場合は6個です。しかし、本当に複雑なキャラクターを、手先から足先まで完璧に造ったら、頂点グループは何十個にもなります! (*The vertex groups controls in the Object Data context*)をご覧ください。Select と Deselect のボタンはそのとき選択されているグループのすべての頂点を(非)選択し、どの頂点がどのグループに属しているのかを調べられます。





Gus in Edit mode with all the vertices of group `Arm.R` selected.

右腕のグループを選択し(`Arm.R`)、全ての頂点を選択解除して(必要に応じてA)、Selectを押します。(Gus in Edit mode with all the vertices of group `Arm.R` selected)のように表示されるはずですが。

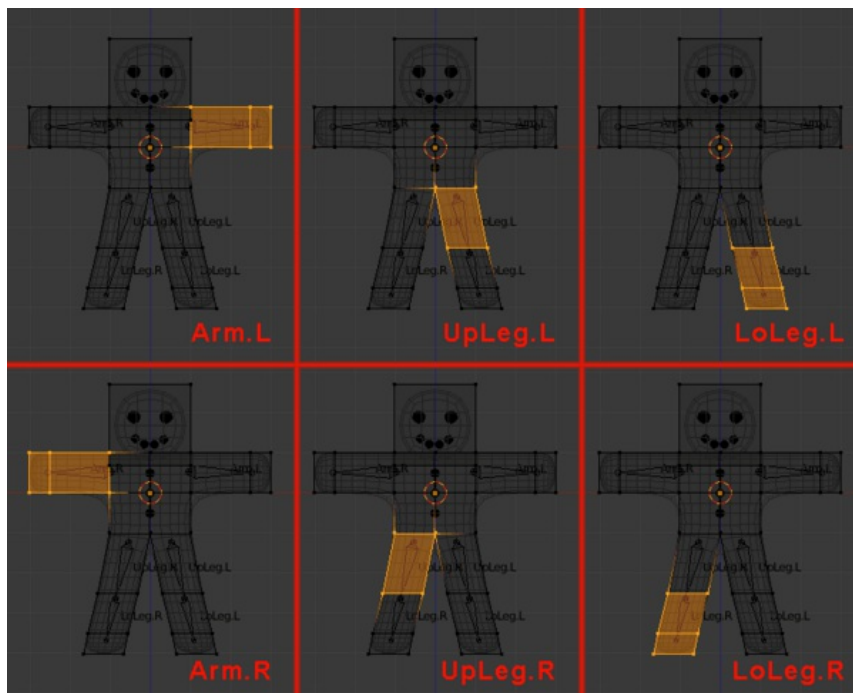
ぴったり正確な位置にボーンを配置したはずなのに、もし同じものが表示されないとしたら、*auto skinning process* がボーンと頂点をマッチングさせるときに過剰に働いてしまったのかもしれませんが。スキニングプロセスが、あなたの期待とまったく同じように骨の頂点にをそろえてくれる可能性はほとんどありません。この場合、あなたは手動でのグループ化の調整を、以下のセクションで説明するように行うことが必要です。

(Gus in Edit mode with all the vertices of group `Arm.R` selected) の赤い丸で印のついた頂点は、含まれるべきではないのに変形グループに属してしまっているものです。

auto skinning process は、それらの頂点がボーンのとでも近くにあると判断すると、それらを変形グループに追加します。そのグループにそれらを入れてたくなくても、Gusの反対側や胸部の頂点のいくつかは、変形のグループに追加されて体のパーツを歪めてしまいます。

そういった頂点を取り除くために、そうではない残しておくべき頂点をすべて選択解除しておき、ボックス選択(B)と、LMB  ではなくMMB  を使って、ボックスを固定して、非選択にするすべての頂点を範囲におさめましょう。

“望ましくない”頂点を選択しおえたら、Remove buttonを押して(*The vertex groups controls in the Object Data context*)、`Arm.R`グループからそれらを除去します。すべてを選択解除してから(A)他のグループのチェックにうつりましょう。他もすべてチェックし、それらが(*The six vertex groups*)のように見えるか確認してください。



The six vertex groups.

頂点グループ

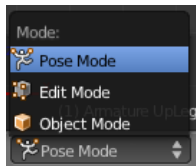
頂点グループへの頂点の割り当てや削除は入念に行いましょう。後で予期しない変形を見かけたとしたら、いくつかの頂点を見落としているか、余分な頂点がグループに置かれている可能性があります。頂点グループの変更はいつでも行えます。

Other details

変形はGusのボディだけに作用し、彼の目、口、ボタンは別個のオブジェクトなので作用されません。これはこの単純なアニメーションにとどまる問題ではなく、もっと複雑なプロジェクトでも考慮する必要があります、たとえば(頂点の)ペアレント化や、ほかにも多種のパーツを接続して単一のメッシュとしてのボディを作るときにも関係してきます (そういったオプションの詳細はすべて [manual](#) に載っています)。

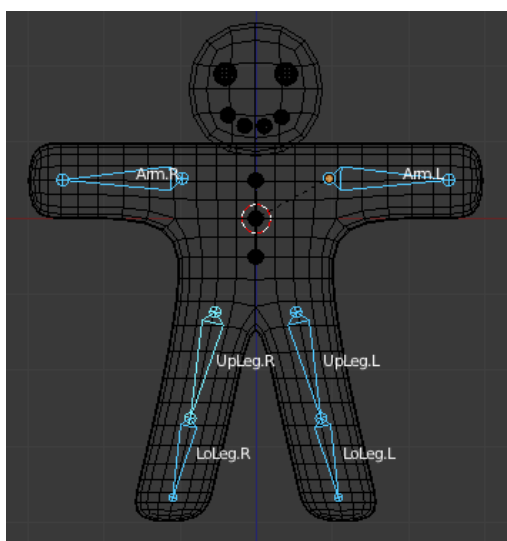
ポーシング

Gusのリグ化とスキン化が済んだら、彼を人形のようにいじって遊べるようになります。ボーンを動かしてその結果を見てみることにしましょう。



Mode menu in the 3D window header.

- アーマチュアだけを選択し、そしてMode menu から Pose Mode を選びます (*Mode menu in the 3D windowheader*) – またはシンプルに `Ctrl+ Tab`と打ちます。このオプションはアーマチュアが選択されている時にだけ利用できます。
- アーマチュア内の選択したボーンは青色に変わります。これでもうあなたは Pose mode に入っています。ここでボーンを選択し、(G)で動かしたり、(R)で回転させると、体はそれに応じて変形します!



You are in Pose mode now!

Original position

Blenderはボーンの元の位置を記憶しています。AltRを押してボーンの回転をクリア、AltGを押してボーン的位置をクリア、というようにアーマチュアの設定を戻せます。また、Object Data context の Rest Position button は、元の位置を一時的に見るために使用するとよいでしょう。

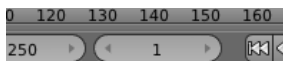
Inverse Kinematics

Inverse Kinematics (IK) とは、チェーンの 端から ボーン位置を定義するもので、“end effector”と呼ばれることもあります。他のすべてのボーンは、アルゴリズムによって位置を推測され、その位置は *IK solver*によって自動的に計算され、チェーンはとぎれることなく保たれます (すなわちIKが数学的にチェーンの位置を導き出してくれるわけです)。IKを使うと手や足の位置決めがはるかに容易で正確になります。

Forward Kinematics

Pose modeでボーンを扱っていると、それらは端に球状関節を有する剛体として振る舞い、堅くて曲げられないことに気づくでしょう。つかむことができるのはチェーンの最初のボーンだけで、他のすべてはそれに従います。チェーンでの後続のボーンはどれも掴んで移動することはできず、回転だけが可能で、選択されたボーンはチェーンでの前のボーンに従って回転し、続くすべてのボーンもこの回転に追従します。

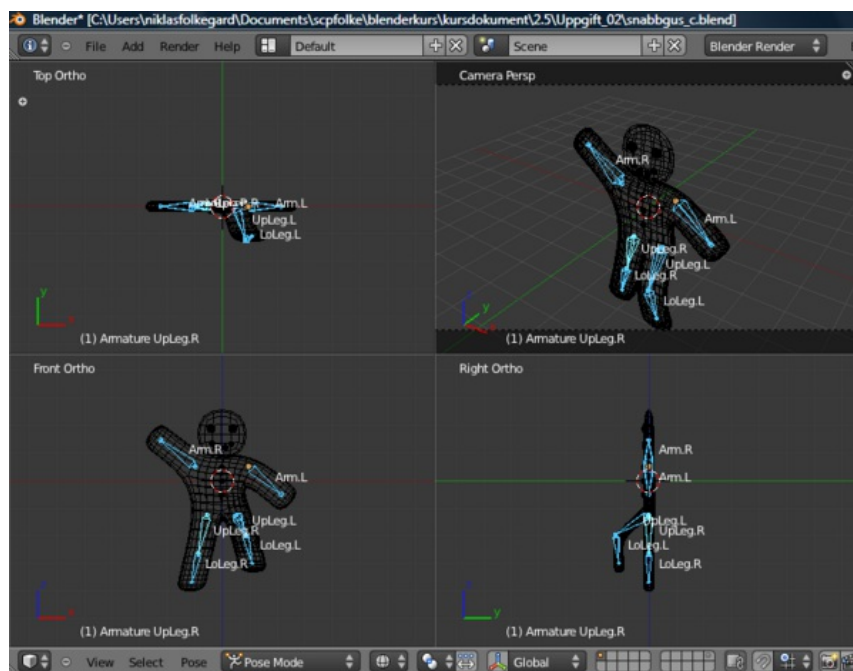
この手続きは、*Forward Kinematics* (FK)と呼ばれ、扱いが容易な反面、チェーンの最後の骨の正確な設置は難しくなります。我々はFKを使用し、足運びを4つのステージに分けそれに対応する4つの異なるポーズ定義して、Gusに歩いてもらいましょう。Blenderはこの作業を流体アニメーションで行います。



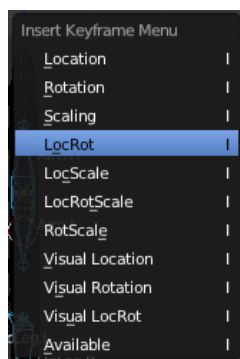
The current frame numeric field in the Timeline window header.

- まず、タイムラインの frame 1 にいることを確認してください。フレーム番号は、Timeline window のヘッダの数値フィールドに表示されます (*The current frame numeric field in the Timeline windowheader*)。それが 1 に設定されていない場合は、今 1 に設定します。
- さて、ボーンをひとつづつ回転させて(R)、UpLeg.Lを上げ、LoLeg.Lをうしろに曲げ、それと同時に Arm.Rを少し上げ Arm.Lを少し下げて、

(Our first pose)と同じようにします。

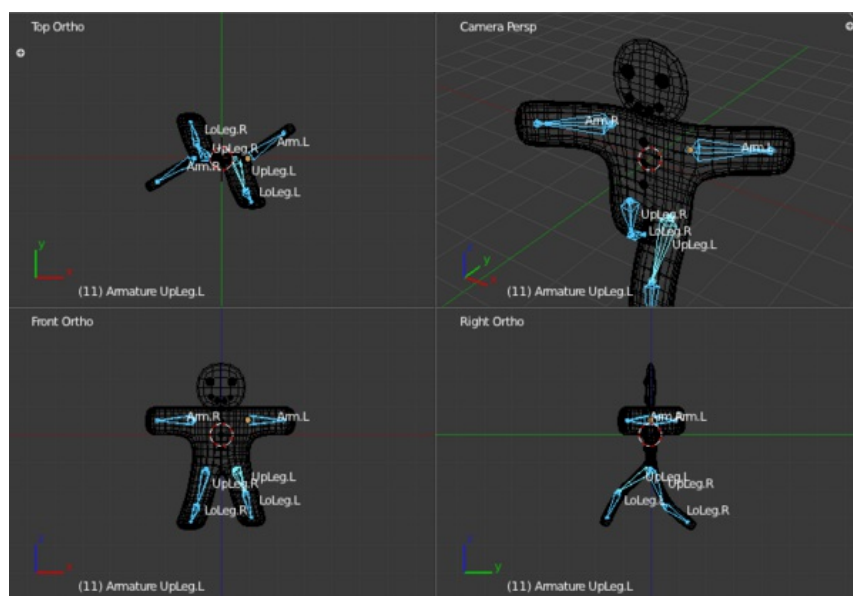


Our first pose.

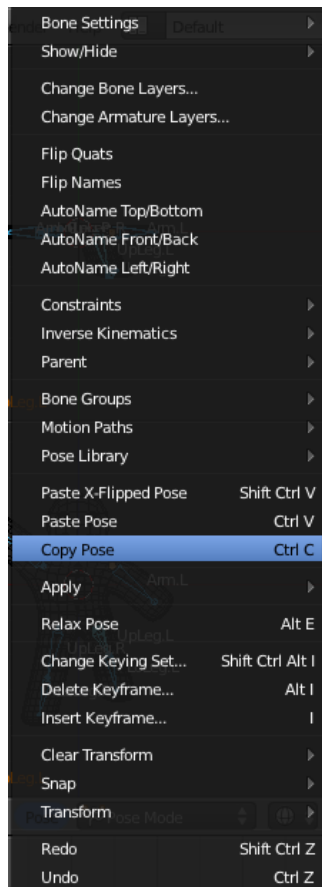


Storing the pose to the frame.

- Aですべてのボーンを選択します。マウスポインタを3D windowにのせて、左押ししましょう。メニューがポップアップします (*Storing the pose to the frame*)。このメニューから LocRot を選択します。これですべてのボーンの位置と向きが取得され、ひとつのポーズとして frame 1 に格納されます。これはGusが足を踏み出す途中を表すポーズで、左足が地面から上がりながら前に動くところです。
- さて、数値フィールドに数値を入力するか ↑ を押して、フレーム11に移動します。そして、(Our second pose)のように、ガスを動かして別のポーズにします。前述の要領で、あらかじめ AltR を使用して両腕の回転をクリアしておきます。トップビューから見ながら、Arm.R をわずかに前方へと回転させ、Arm.L はすこしだけ後方へと回転させます。最後に、左足を前に、右脚は後ろにポーズをつけ、両足とも少し曲げておきます。Gusがその場で歩いています!



Our second pose.



Pose menu.

- もう一度すべてのボーンを選択し、Iを押してポーズをフレーム 11に格納したら、Rotを選びます。
- 我々が今必要なのは、frame 21となる第三のポーズで、右脚を上げたところ、つまりはストライドの真ん中から残り半分です。このポーズは我々がフレーム1で定義したもののミラーです。ですので、すべてのボーンが選択された状態でフレーム1に戻り、3D windowのヘッダのPoseメニューで、Copy Poseという見出しを選択します、(*Pose menu*)参照。もしくは CtrlCを使いましょう。これで、この時点のポーズがバッファにコピーされました。
- frame 21に移動し、Pose menuのPaste X-Flipped Pose optionでポーズをペーストします、(*Pose menu*)参照。もしくは ShiftCtrlVを使いましょう。これはカットしたポーズの添え字が“.L”のボーンと“.R”のボーンの位置を切り替えてペーストするものです。効果的にひっくり返してくれます!

Bone roll

X-flipped poseでペーストする場合、Gusが間違った方向に曲ることがあり、ボーンの回転に問題がおこるかもしれません。Edit Modeですべてのボーンを選択して CtrlN, 3と押し、ボーンの回転を整理しましょう。その後、frame 1と11に戻ってポーズを調整しましょう。frame 1からポーズを再コピーし、frame 21でx-flipped poseを再びペーストしてみましょう。

ポーズはここにありますが、まだ格納されてはいません! すべてのボーンを選択して I » Rotと押す必要があります。

- ここでも同じ手順を適用し、frame 11でポーズをコピーし、それを反転して 31, にペーストします。
- サイクルを完成させるために、frame 1のポーズを回転させずにframe 41にコピーする必要があります。いつものようにコピーして、そして Paste Pose entryを使用しましょう。または CtrlVを使用しましょう。I » Rotでポーズを格納し、手順を終了します。

Checking the animation

アニメーションをプレビューするには、current frameを1にセットして3D windowでAltAと押します。

Gusが歩いてる!

その場でのシングルステップは歩行動作の中核であり、ここまでひとつ定義してきましたが、他にもキャラクターを複雑な道すじに沿って歩くさせるテクニクの数々もあります。ただ、私たちのクイックスタートという目的においては、このくらいのその場でのシングルステップで十分でしょう。

- Properties windowのRender contextで、開始フレーム(Start)を1(デフォルトですでに1になっているはず)を設定し、最後のフレーム(End)を40(これはデフォルトで250に設定されている、*Settings for an animation in the Render context*参照)。frame 41はframe 1と同じなので、完全なサイクルを生成するためにレンダリングする必要があるのは1から40までのフレームだけです。



Settings for an animation in the Render context.

- Output panel のテキストフィールドに `//render/`とタイプしましょう (*Settings for an animation in the Render sub-context*)。
- Format panel でファイルの形式として AVI Raw を選択します (*Settings for an animation in the Render sub-context*)。これは一般的には最良の選択とはいえ、主にファイルサイズの問題があるとはいえ、高速で、任意のマシン上で実行できるので、我々のニーズには適しています。また、よりコンパクトなファイルを生成する AVI Jpeg を選択することもできます。ただし、非可逆JPEG圧縮を使用して動画を生成すると、いくつかの外部プレーヤーでは再生できないこともあります。

最後に、Render panel で Animation button を押してください。あなたがアニメーションで使用する *すべての* レイヤーを表示しておく必要があることに気をつけましょう! 私たちのケースでは、レイヤー 1 と 10 です。

Stopping a Rendering

間違えたとき、たとえばレイヤー10を消し忘れたなんて時は、Escでレンダリングプロセスを停止することができます。

私たちのシーンは非常に単純で、そしてBlenderはたぶん数秒かそこらで40の画像それぞれをレンダリングしてしまうでしょう。表示されたら、それらを見てみましょう。

Stills

もちろん、あなたはいつでも、アニメーションのひとつひとつのフレームを静止画としてレンダリングすることができます。レンダリングしたいフレームを選択して RENDER buttonを押すだけです。

レンダリングが完了したら、あなたはもう `0001_0040.avi` という名前の付いたファイルを取得しているはずですが。カレントディレクトリの `render` subdirectory で `.blend`ファイルなどが入っているところです。ディレクトリは Output panel から変更できます。

top menu の Play Rendered Animation を押すと (または `CtrlF11` を使用すると)、レンダー内で直接このファイルを再生することもできます。アニメーションは自動的に循環します。停止するには `Esc` を押してください。私たちは、ごく基本的な歩行サイクルを作成しました。Blenderにはまだまだたくさんの機能が、whole [manual](#)を読んでいけばすぐに発見していけるでしょう!

ファイル保存

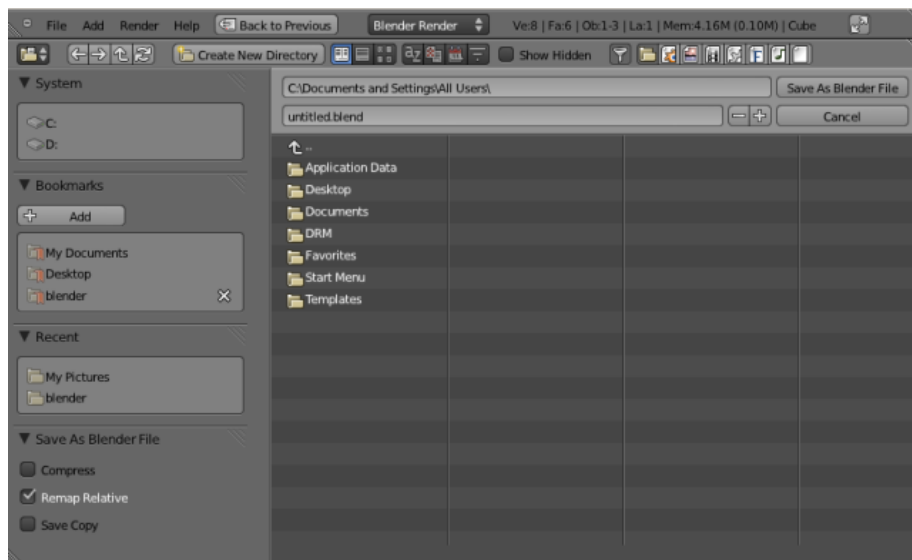
Mode: All modes

Hotkey: F2

Menu: File » Save

説明

ファイル保存は、ファイルを開くのと似ています。F2 キーを押すと、File Browser ウィンドウが開きます。ウィンドウには、サイドパネルのいくつかのオプションを除いて、ファイルを開くときと同様に表示されます。説明とファイルブラウザの機能の使用方法については、前の [ファイルを開く](#) ページを参照。



保存

下のエディットボックスをクリックしてファイル名を入力します。もし、ファイルの末尾に “.blend” がいない場合、自動的に付加されます。その後 ↵ Enter するか、Save File ボタンをクリックを押して保存してください。

同じ名前前のファイルが既に存在する場合、Save Over のポップアップダイアログで上書き保存するかどうか確認する必要があります。

あなたがセットした [save versions](#) の数に応じて、すべての既存のファイルは `.blendn` のファイル拡張子にローテーションします、 n は 1, 2, 3 などです。

すなわち、あなたが `MyWork.blend` で作業していて、それを保存したら、既存の `MyWork.blend` は `MyWork.blend1` に名前を変えられます、そして、新しい `MyWork.blend` が保存されます。この方法は、大量の変更を元に戻す必要がある場合は、ホット バックアップの古い保存された版を開くことができます。

保存オプション

保存オプションは、サイドパネルの下部に表示されます。

ファイル圧縮

File » Compress File を有効にするをデッドスペースを削除し、大きなファイルを圧縮します。

相対リマップ

このオプションは新しい場所でファイルを保存するときに相対パスをリマップします。

コピー保存

このオプションは、現在の作業状態のコピーを保存しますが、保存したファイルをアクティブにしません。

インクリメント保存の Tips

保存ダイアログは、作業の複数のバージョンを作成するのに役立つために小さな機能が含まれています。: + NumPad または - NumPad を押してファイル名の末尾の番号をインクリメント・デクリメントします。単純に現在ロードされているファイルをダイアログを出さずに単純に上書き保存するには、F2 の代わりに Ctrl+W を入力します。

ファイルを開く

Mode: All modes

Hotkey: F1

Menu: File » Open

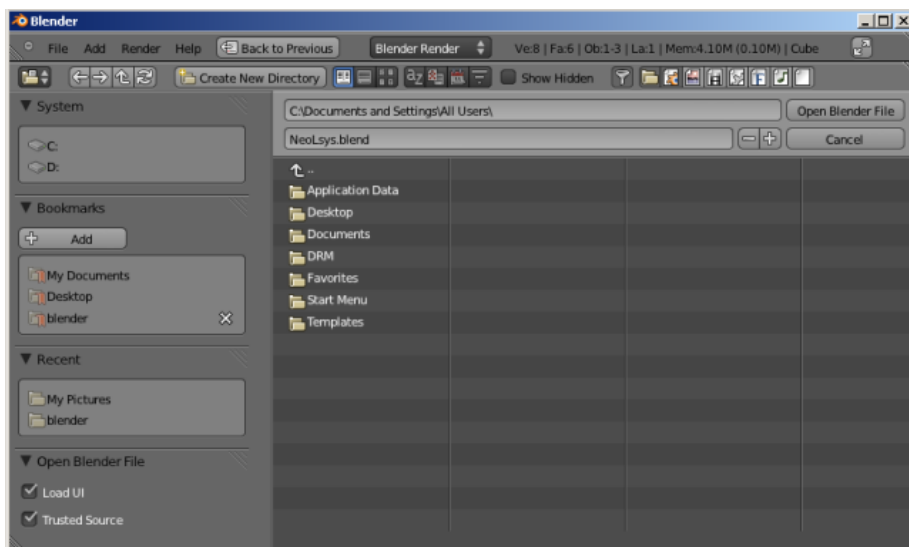
説明

Blender はほぼすべてのもの(オブジェクト、シーン、テクスチャ、およびすべてのユーザインタフェースウィンドウの設定)を保存する .blendファイル形式を使用します。:



Blender は、あなたが何をしているかについてわかっていると予想します！ファイルを読み込む場合に、作業しているシーンに保存されていない変更を、保存するように求められることはありません、ファイルの読み込みダイアログを完了することは偶然実行していない確認であると考えられています。

ファイルを保存することを確認してください。



ファイルブラウザやフォルダナビゲーションの使用

ディスクから Blender ファイルをロードするには、F1 を押します。以下のように File Browser ウィンドウが開きます。



上部のテキストボックスには、現在のディレクトリパスを表示し、下のテキストボックスが選択されたファイル名が含まれています。P(または P ボタン)は、親ディレクトリに移動します。

+ と - ボタンは、ファイル名の末尾に増加したり、数を減らすことによって、番号付きファイルを循環することができます。

フォルダをクリックするとフォルダの中を、ファイルをクリックして Open Blender File ボタンを押すとそれを開こうとする

cancel をクリックすると、ファイルブラウザのウィンドウを閉じてプログラムに戻ります。

サイドパネル

左側のパネルには、ファイルといくつかのオプションを見つけるために様々な方法が表示されます。ファイルをロードするには、LMB  を選択し Enter を押すか、Open File ボタンをクリックしてください。ファイルは、MMB  をクリックすることでもロードすることができます。

システム

システムメニューは、ファイルを見つける事をナビゲートするために利用可能なドライブのリストが含まれています。そのドライブにジャンプするいずれかをクリックしてください。

ブックマーク

ブックマークはファイルブラウザにそれらに移動しなくても、頻繁にアクセスできるようにするフォルダです。ブックマークメニューにディレクトリを追加するには、そのフォルダに移動して Add をクリックします。フォルダを削除するには、横の X ボタンをクリックするだけです。

最近の (recent)

これは、最近アクセスしたフォルダの一覧です。Recent Files とラベルがついているボックスで、ユーザー Preferences の File タブから、どれくらいのフォルダがリストに記載されるか制御することができます。

開くのオプション

各々の .blend ファイルの中に、Blender はユーザ・インタフェース - スクリーン・レイアウト を保存します。デフォルトでは、この保存された UI は、使用している任意のユーザのデフォルト値または現在の画面レイアウトを上書きして、ロードされます。現在のデフォルトを用いて blend ファイルで作業をする場合、ファイルブラウザ (F1) を開いて、フレッシュな Blender を起動します。Load UI ボタンをオフにして、ファイルを開いてください。

ヘッダーパネル

ヘッダには、ナビゲーションファイルのいくつかのツールが含まれています。4つの矢印のアイコンは、次のことができます。

- 前のフォルダに移動
- 次のフォルダに移動
- 親ディレクトリに移動
- 現在のフォルダを更新

Create New Directory アイコンをクリックすると、現在のフォルダ内に新しいフォルダを作成します。

他のアイコンを使用すると、表示するか、どう見えるかファイルを制御することができます。次の操作を実行できます。

- ファイルをショートリストで表示
- ファイルを詳細なリストで表示
- ファイルをサムネイルで表示

ファイルの並べかえ:

- アルファベット順
- ファイルタイプ順
- 最終編集時間順
- ファイルサイズ順

ファイルタイプ表示のフィルタリングコントロール。Enable Filtering アイコンをクリックし、表示されている種類をトグルで切り替え:

- フォルダ
- Blend ファイル
- イメージ
- ムービーファイル
- スクリプト
- フォントファイル
- ミュージックファイル
- テキストファイル

その他のファイルを開くオプション

File メニューから、以下のツールでファイルを開くことができます。

Open Recent

最近使用したファイルを一覧表示します。クリックしてロードします。

Recover Last Session

自動保存された `quit.blend` をロードします。つまり、このオプションは、最後の作業セッションを回復することができます。例えば、アクシデントでBlenderが落ちた時とか...

Recover Auto Save

自動保存したファイルを回復するために開きます。

ファイルの保存

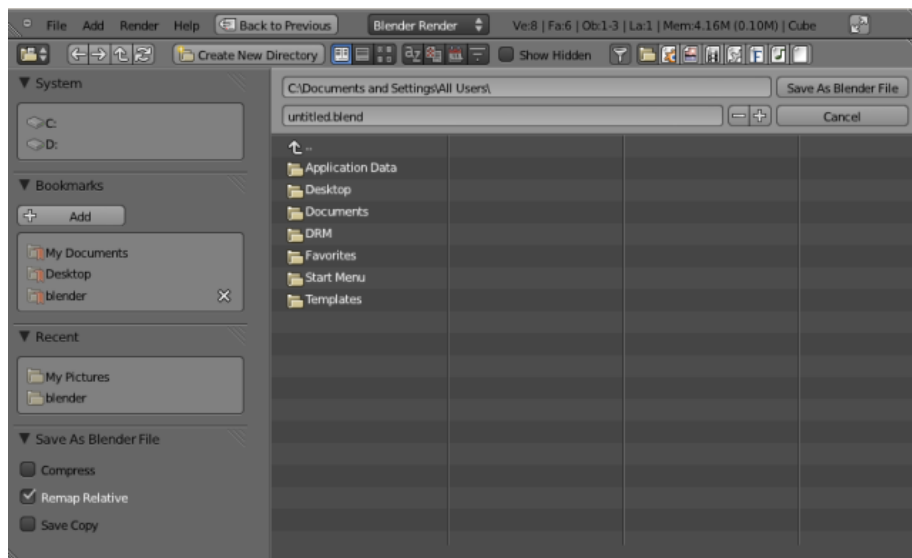
モード: 全てのモード

ホットキー: F2

メニュー: ファイル » 保存 (File » Save)

概要

ファイルの保存はファイルの読み込みに似ています。F2を押すと、ファイルブラウザ (File Browser)ウィンドウが開きます。ウィンドウはファイルを開いたときと同様に現れます、サイドパネルにいくつかのオプションがあることを除いて。ファイルブラウザの機能の概要と使い方は、[ファイルを開く](#)を参照してください。



保存

ファイル名を入力するために下の編集ボックスをクリックしてください。もし、ファイル名が“.blend”で終わらなかった場合、拡張子が自動的に追加されます。ファイルを保存するには、**↵** Enterを押すか、ファイルを保存 (Save File)ボタンをクリックしてください。

既に同名のファイルが存在していた場合、Save Overポップアップダイアログで上書きしたいということを承認しなければいけません。

設定した[バージョンを保存](#) (Save Versions)の数値によって、同名を持つ全ての既存のファイルは、.blend n ファイル拡張子(n は1、2、3などとする)にローテーションされるでしょう。なので、MyWork.blendで作業をしていて、それを保存した場合、既存のMyWork.blendはMyWork.blend1に名前変更され、新しくMyWork.blendが保存されます。このように、もし大規模な元に戻す変更が必要になった場合に開くことが可能な、過去に保存されたバージョンのホットバックアップを持っています。

保存オプション

保存オプションはサイドパネルの下に現れます。

ファイルを圧縮 (Compress File)

巨大なファイルを圧縮するためには、このオプションを有効にしてください。このオプションは無駄な空間を無くします。

相対パスを更新 (Remap Relative)

このオプションは、ファイルを新しいロケーションに保存したときに、相対パスの位置を更新します。

コピーを保存 (Save Copy)

このオプションは実際の作業状態を保存します。しかし、保存されたファイルを有効にはしません。

インクリメント保存のための小ネタ

保存ダイアログは、作品の複数のバージョンを作るために役立つ、小さい特徴を含みます: + NumPadや- NumPadを押すことで、ファイル名の末尾の番号を加算したり減算したりできます。保存ダイアログを飛ばして、現在読み込んでいるファイルを簡易に上書き保存するためには、F2の代わりにCtrlWを押して、プロンプトで承認してください。

Importing Files

(ファイルのインポート)

他形式のファイルのインポートができます。

Collada (.dae)

[こちらをご覧ください。](#)

Motion Capture (.bvh)

Target(ターゲット)

インポート対象としてオブジェクトかアーマチュアを選びます

Scale(拡大縮小)

BVHの拡大率です

Start Frame(開始フレーム)

アニメーションの開始フレームです

Scale FPS(FPS のスケーリング)

フレームレートを BVH から現在のシーンにスケーリングします。行わない場合、各 BVH フレームは直接 Blender フレームにマッピングされます

Loop(ループ)

アニメーション再生をループします

Rotation(回転)

回転を変換します。Quaternion(クォータニオン)か Euler(オイラー)を使います

Forward(前方)

前向きの軸を指定します

Up(上)

上向きの軸を指定します

Scalable Vector Graphics

vector graphic を Curve(曲線)オブジェクトとしてインポートします

Stanford (.ply)

[こちらをご覧ください。\(eng\)](#)

Stl (.stl)

[こちらをご覧ください。\(eng\)](#)

3d studio (.3ds)

[こちらをご覧ください。\(eng\)](#)

Size Constraint(サイズ制限)

サイズ制限に達するまで 10 ずつモデルを拡大縮小します(0 で無効化)

Image Search(画像検索)

サブディレクトリにある関連画像を探します

Apply Transform(トランスフォームを適用)

オブジェクトのトランスフォームが正しくインポートされない場合の回避策です

Forward(前方)

前向きの軸を指定します

Up(上)

上向きの軸を指定します

Autodesk FBX (.fbx)

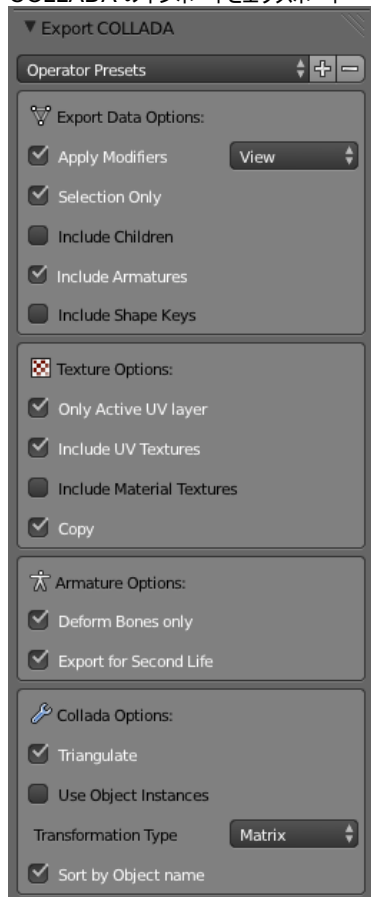
[こちらをご覧ください。\(eng\)](#)

Wavefront (.obj)

[こちらをご覧ください。\(eng\)](#)

X3D Extensible 3D (.x3d)

COLLADA のインポートとエクスポート



Collada モジュールは .dae ファイルのエクスポートやインポート用の柔軟なツールとして実装されました。数多くのツールとの間で Collada ファイルのやり取りを可能にするような、パラメータの提供に気を配りました。しかしながら、Collada モジュールはまだ完成していないことをどうか覚えておいてください。目的の用途にかなう機能がまだサポートされていないことも考えられます。

Collada のエクスポート

プリセットのオペレータ

Second Life ユーザー向けのオペレータプリセットを2つ追加しました (オプションパネルの最上部をご覧ください):

- **Second Life Static**
static mesh のエクスポートに向いています。
- **Second Life Rigged**
SL default character のエクスポートに向いています。

Second Life ユーザー向けの特記事項:

- プリセットを使ってください。他のエクスポート設定はどれも、Second Life 用には動作しません。
- 人物の向きは、人物が正の X 方向を向くようにする必要があります。
- 拡大縮小と回転はエクスポート前に適用しておく必要があります！

エクスポートデータのオプション

モディファイアの適用

アクティブなモディファイアは非破壊なやり方ですべて適用されます。これは、メッシュのコピーに対してモディファイアが適用されるということです。したがってエクスポート前に自分でモディファイアを適用する必要はもうありません。バックグラウンドで自動的に行われます。

プレビュー/レンダーモード

モディファイアにはプレビューモードとレンダーモードを別のメッシュ設定で行えるものがあります。モディファイアの適用時にはどちらのモードもサポートします。

Selection Only (選択部分のみ)

selection only を有効にすると、選択オブジェクトだけがエクスポートされます。無効なら可視、不可視両方のオブジェクトを含む、シーン全体がエクスポートされます。

Include Children (子を含む)

有効なら選択オブジェクトのすべての子要素も (選択されていない場合でも) エクスポートされます。

例

アーマチュアを1つだけ選んで「include children」を有効にしてエクスポートすると、そのアーマチュアに関連づけてリグづけしたメッシュもすべて、エクスポートされます。

Include Armatures(アーマチュアを含む)

有効なら選択オブジェクトに関連したすべてのアーマチュアも(選択されていなくても)エクスポートされます。

例

オブジェクトを選択して「include armatures」を有効にしてエクスポートすると、アーマチュアのデータもエクスポートされます。

Include Shape keys (シェイプキーを含む)

シェイプキー

このオプションにはシェイプキーの適用も含まれます！ したがって現在のシェイプキーの形状がバイクされたメッシュをエクスポート可能になりました。

テクスチャのオプション

Only Active UV layer(アクティブな UV レイヤーのみ)

メッシュが UV レイヤーを複数持っているとき、Blender はデフォルトではすべてのレイヤーをエクスポートします。このオプションを使うとアクティブな UV レイヤーだけをエクスポートできます。

Include Textures (テクスチャを含む)

Blender は2種類のオブジェクトのテクスチャマッピング方式をサポートします。

1. 直接割り当てられている surface texture を使う方法
2. マテリアルベースの画像テクスチャを使う方法

マテリアルベースの画像テクスチャは柔軟性を持つ一方で、surface texture を使う方法ははじめにテクスチャをレンダリングする必要なしに、素早く作れます。これまでは Blender がエクスポートできるのはマテリアルベースのものだけでした。新たなオプションを使うと、レンダリング結果を直接エクスポート可能です。

テクスチャのエクスポートにはマテリアルが必要です

surface texture を使うために、テクスチャ面それぞれに依然としてマテリアルを作る必要があります。そうすればあとはメッシュの正しい面に画像を割り当てだけです。最終的にオブジェクトが希望どおりの外見になれば、そのオブジェクトを「Include UV Textures」オプションでエクスポートします。下の「Copy」オプションも参照してください。

Copy(コピー)

画像をマテリアルベースや surface textures でエクスポートすると、エクスポートファイル内に絶対的なファイル参照が作られます。

しかし「Copy」オプションが有効なら、代わりに画像のコピーを作り、エクスポートファイルに加えてコピーを置きます。この場合ファイル参照は相対的に作られます。

アーマチュアのオプション

Deform Bones Only(ボーンのみ変形)

有効にすると、エクスポートされたアーマチュアから変形しないボーンをすべて取り除きます。アーマチュアが、実際にはキャラクターの骨組みの一部ではない制御用ボーンを含むときに役立ちます。例えばこのオプションを有効にすると Avastar rig をエクスポート可能です。エクスポートされたリグは Second Life と互換性のあるものになります。ただし、この下の他の制約に注意してください。

Export for Second Life(Second Life用エクスポート)

非常に特別なオプションです。有効にするとボーン方向に関する計算が変わります。これはリグづけされたメッシュにだけ関係します。私はこのオプションがやがてもっと有意義なもの(かつ Second Life に依然互換性のあるもの)に置き換えられることを期待しています。ヒント: このオプションはリグづけされたメッシュをエクスポートしたいときにだけ重要です。静的なメッシュには単に何もしません。

Collada のオプション

Triangulate(三角面化)

メッシュはすぐに三角面化できます。三角面化はユーザーインターフェイスにある「選択面を三角面にする」機能をベースにしています。三角面化を完全に制御するために、エクスポート前に三角面化を手動で行うことができます。ただ、このオプションはエクスポートされたデータの三角面化だけを行うものです。元のメッシュには影響しません。

Use Object Instances(オブジェクトのインスタンスを使用)

Blender では同じメッシュを複数のオブジェクトに再利用できます。これを「オブジェクトのインスタンス化」と呼んでいます。このオプションを有効にすると、Blender はオブジェクトのインスタンス化を Collada ファイルに伝播します。

Transformation Type(トランスフォームの種類)

Collada は2種類の変換行列 (Transformation matrix) の仕様をサポートしています。<Matrix> として、または変換分解の集合としてです (Translate/Rotate/Scale 向け)。エクスポート機能はこの設定に厳密に従うわけではなく、可能であればヒントとして使います。これは、エクスポートされたデータタイプには変換行列の独自のエクスポート規則を持つものがあるからです。開発は継続中で、将来的にはあいまいさを抑えた方式を提供できるかもしれません。

Sort by Object Name (オブジェクト名で並べ替え)

データのエクスポート順序は内部のオブジェクトの順序に結び付けられていて、順序を確実に決められません。このオプションは Geometry node と Object node がどちらもアルファベット順にエクスポートされることを保証します。

Collada のインポート

Collad のインポート機能は主に、インポートしたデータによって制御されます。Blender が用意する追加オプションは1つだけで、Import units を制御するためのものです:

- Import Units (単位もインポート)

無効にすると、インポートされたデータは現在使用中の単位系にあわせて拡大縮小されます。Blender unit が1メートルであることを想定しています。有効にすると、Blender 側が Collad ファイルの持つ単位系に合わせて調整されます。

技術的詳細

メッシュ

インポート

サポートされるジオメトリ

- tris (未テスト)
- polylist
- polygons
- ngons
- trifans (未テスト)
- lines

エクスポート

メッシュデータは <polylist>、<lines> および <vertices> としてエクスポート。

光源

インポート

Blender は .dae から光源をインポートする際最善を尽くします。光源用の Blender profile が見つかると、そこにある値がすべて、替わりに使われます。これは Blender のエクスポートした .dae から 100% 再インポートできるということです。<extra> のサポートは Blender 2.57 で追加されました。

エクスポート

光源に関する Blender profile が <extra> tag を通じて追加されました。光源の 減衰曲線を除く、Blender の Lamp の構造全体がこの profile を通じてエクスポートされます。

マテリアルとエフェクト

エクスポート

Blender 2.57 以降では effect のエクスポートに少し変更が入っています。特筆すべきものとして <lambert> は specularity が 0 のときにだけエクスポートされます。

アニメーション

エクスポートとインポート

- オブジェクト (メッシュ、カメラ、光源) のトランスフォームのアニメーションをサポート。現在のところ、オブジェクトのデフォルトオプションであるオイラー回転のみがエクスポートされます。アーマチュアボーンのアニメーションではオイラーおよびクォータニオンの回転をサポート。
- 以下のパラメータのアニメーションの、インポート/エクスポートをサポート:-
 - 光源
 - カメラ
 - マテリアルエフェクト
- スキンの制御用でないアーマチュアのボーンのアニメーション
- スキンを変形するボーンを使ったアーマチュアのアニメーション
- オブジェクトモードでのアーマチュアのアニメーション
- 完全に rigify で作られたアーマチュアのアニメーション。rigify で作られたアーマチュアのアニメーションは、以下の手順でエクスポートします。
 - Bake Action を選びます (3d view でスペースを押し Bake Action を入力)
 - deform bone だけを選択しているなら "Only Selected" をチェックします。dae ファイルが小さくなります。そうでなければ "Only

- Selected" のチェックを外します
- "Clear Constraints" にチェックします
- Bake Action を実行します
- メッシュを選んでボーンを変形します。"Only Selected" オプションを有効にして Collada にエクスポートします (厳密にはメッシュとボーンだけを選択する必要はありません。こうすると dae ファイルが小さくなります)
- [デモ](#)

Nodes

インポート時、<instance_node> 用の親のトランスフォームは正しく子ノードインスタンスに伝播します。Blender のマテリアルは次のマッピングを使ってエクスポートされます:

- phong
- blinn
- lambert

アーマチュアの木構造で葉ノードであるボーンのノード、またはボーンに子要素が2つ以上あるときには、こうした関節ノードのために情報用の blender profile が <extra> tag を使って追加されます。再インポート時に bone->tail 位置を正しく導き出すためです。

重要事項

- オブジェクトとデータブロックの名前は 21 文字 (bytes) に制限されます
- UV レイヤー名は 32 文字 (bytes) に制限されます
- only armature animation on mesh, single skin controller
- モディファイアはまだサポートされていません

<instance_node> を持つ .dae をインポートすると、エクスポート時にこの情報は基本的に失われ、これらのノードは <node> になります。

Exporting Files

(ファイルのエクスポート)

プロジェクトでの作業が、1つのソフトウェアで行えるとは限りません。たいていは、Blender で作業を始めるとしても、他のソフトウェアに切り替えて続きを行いたくなるでしょう。

Blender は他形式への エクスポートを行えるので、こういったことも可能です。Blender 2.65 は現時点で COLLADA を含む 8 つの形式へのエクスポートに対応しています。

Blender ユーザーでない人、例えば Autodesk® 3DS Max® ユーザーともファイルをやりとりできるようになるので、この機能は重要です。

Collada (.dae)

[こちらをご覧ください。](#)

Motion Capture (.bvh)

[インポート](#)にのみ対応。

Stanford (.ply)

[こちらをご覧ください。\(eng\)](#)

Apply Modifiers (モディファイアを適用)

モディファイアを適用してバイクします。

UV's (UV)

ジオメトリのアクティブな UV レイヤーの座標をエクスポートします

Normal (ノーマル)

smooth や hard な辺用のジオメトリの法線データをエクスポートします

Vertex Colors (頂点色)

アクティブな頂点色のレイヤーをエクスポートします

Stl (.stl)

[こちらをご覧ください。\(eng\)](#)

Ascii

ASCIIファイル形式でファイルを保存します。

Autodesk® 3DS Max® (.3ds)

[こちらをご覧ください。\(eng\)](#)

Selection Only (選択物のみ)

選択したオブジェクトだけをエクスポートします

Forward (前方)

前向きの軸を指定します

Up (上)

上向きの軸を指定します

Autodesk® FBX (.fbx)

[こちらをご覧ください。\(eng\)](#)

Wavefront (.obj)

[こちらをご覧ください。\(eng\)](#)

X3D Extensible 3D (.x3d)

対応している形式

画像形式

これはBlenderが内部的に対応している画像ファイル形式の一覧です:

ハイダイナミックレンジ画像

Blenderの画像入出力システムは、普通の32ビット画像(4×8ビット)と、ピクセルあたり128ビットの浮動小数点画像(4×32ビット)を透過的にサポートしています。

HDR画像の読み込みでは、その画像が例えば3×10ビットの時であっても、ピクセルは常に内部でRGBA float値へと変換されます。UV画像エディタ内で画像を表示する間のように、必要に応じて、これを行った後で、速い表示の為に普通の32ビットへと変換されます。

画像がfloatカラーを持っている場合は、Blenderにおける全ての画像機能が標準でそれを使います。これはビデオシーケンスエディター、テクスチャマッピング、背景画像、コンポジターを含みます。

どうやってHDR画像を操作・表示するかのヒントは、カーブUIのページをチェックしてください。

- [OpenEXR \(マルチレイヤー\)](#)
- [DPX、Cineon、Radiance HDR](#)

他の形式

他のサポートしている画像形式がここにあります:

- BMP
- DDS
- SGI IRIS (古い!)
- PNG
- JPEG
- JPEG 2000
- Targa
- Targa Raw (非圧縮Targa)
- TIFF

動画形式

- AVI (Windows)
- AVI JPEG
- AVI Raw
- Frame Server
- H.264
- MPEG
- Ogg Theora
- QuickTime
- Xvid

カラーモード

- BW: 画像は8ビットグレイスケールで保存されます (PNG、JPEG、TGA、TIFのみ)。
- RGB: 画像はRGB(カラー)で保存されます。
- RGBA: 画像はRGBとアルファデータで保存されます (対応している場合)。

色深度

- 8ビットカラーチャンネル
- 12ビットカラーチャンネル
- 16ビットカラーチャンネル
- 32ビットカラーチャンネル

リファレンス

- http://projects.blender.org/scm/viewvc.php/trunk/blender/source/blender/makesrna/intern/rna_scene.c?view=markup&root=bf-blender

OpenEXR

ILM の [OpenEXR](#) は、特にその柔軟性と拡張性を持つ構造により、ソフトウェア業界での HDR 画像ファイルの標準になりました。

OpenEXR ファイルは浮動小数点空間内の値を、正負どちらの数値も格納できます。

RGBA と任意に保存された Z 値を使ったすべての OpenEXR ファイル形式の読み込みとは別に、Blender はふたつの方法で OpenEXR をサポートします:

- [レンダリング結果の出力](#)
- [Multi-layer, Multi-pass, tile-based の各ファイル](#)

Render Output

(レンダリング結果の出力)

以下は OpenEXR のレンダリング結果の出力で利用可能なオプションです (プロパティエディタの Render タブ、Output パネルにあります):

Half (Float(Half))

画像をカスタムの 16ビット/チャンネルの浮動小数点形式で保存します。実際の「ビット深度」を5ビットの指数値と1ビットの符号を使った 10ビットに抑えます。

Zbuf (Zバッファ)

奥行き情報を保存します。カメラからの正確な距離を表します。Blender ではこれも今は "Blender unit" 単位の float で書き込まれます。

Preview (プレビュー)

アニメーション(またはコマンドラインを通じた単一フレーム)のレンダリングで、クイックプレビューやダウンロード用に同じ画像を JPEG でも保存します。

Codec (コーデック)

圧縮形式を選びます。

- PIZ - 可逆ウェーブレット圧縮。ざらついた画像をうまく圧縮します
- ZIP - zlib を使った、標準的な可逆圧縮
- RLE - ランレングス式符号化、可逆。走査線が同じ値を持つときうまく働きます
- PXR24 - Pixar 製非可逆アルゴリズム。32 bits floats を 24 bits floats に変換

Multi-layer, Multi-pass, tile-based files

(マルチレイヤー、マルチパス、タイルベースのファイル)

OpenEXR ファイルはレイヤーとパスを無制限に階層構造で格納し、保持できます。この機能は現在は "Save Buffers" レンダリングオプション用に使われています。このオプションはレンダリング前に最終的なレンダリング結果全体(多くのレイヤーやパスを持ち得ます)のためにファイルを確保せず、中間物であるタイルそれぞれを1枚の OpenEXR ファイルとして Blender の デフォルトの 'temp' ディレクトリに保存します。

レンダリングが完了すれば、レンダリングのデータをすべて解放したあとで、全体をメモリに読み直します。

DPX and Cineon

Cineon は Kodak のフィルム走査用の規格で、10ビット/チャンネルの Log 形式です。DPX は ANSI/SMPTE 工業規格として Cineon から派生したものです。DPX は 16ビットカラー/チャンネルで、Log だけでなくリニアもサポートします。DPX は現在広く採用されている規格で、映画ハードウェア/ソフトウェア産業で使われます。

Cineon と同様 DPX は「目に見える」色だけを(レンダリングや合成の結果として) 0.0 から 1.0 の範囲の値で格納し変換します。アルファは書き込みません。

プログラムコードは CinePaint からありがたくコピーさせていただきました。CinePaint のメインプログラマー Robin Rowe によれば、DPX のコードのデフォルトは Log 式の色空間です。これを無効にする方法を我々はまだ発見できませんが、明らかな損失なしに読み書きがうまくできているようです。

Blender DPX ファイル(Elephants Dream ムービーの全部)は問題なく Quantel IQ HD/2K の仕上げ/グレーディングセットに取り込むことができたため、ひとまずはプロ用途に対応していると考えています。

Radiance HDR

Radiance は照明効果のシミュレーション用ツール群で、Greg Ward によって作られました。Radiance は最初の(そして長い間唯一だった)HDR 画像形式を持っていましたが、この形式は他の多くのソフトウェアがサポートしています。

Radiance (.hdr) ファイルは色の各成分を依然として8ビットで格納しますが、追加の(共有される)8ビット指数値を使って、ピクセルあたり32ビットまで格納できるようにします。このファイルには RGB だけを格納できます。

Overview

(概観)

各 .blend ファイルにはデータベースがひとつあります。データベースにはファイル内にあるすべてのシーン、オブジェクト、メッシュ、テクスチャなどが含まれます。ファイルは複数のシーンを含むことができ、各シーンは複数のオブジェクトを含むことができます。オブジェクトは複数のマテリアルを含むことができ、マテリアルは複数のテクスチャを含むことができます。異なるオブジェクト間でリンクを張ることもできます。

Outliner

(アウトライナー)

アウトライナー (Outliner) ウィンドウを使えば、ファイルの中身を簡単に調べられます。このウィンドウは Blender のデータシステムを表示し、データベースを複数の方法で表示します。また、選択、改名、削除、リンクといったオブジェクトの簡単な操作もできます。ヘッダーにあるボタンで表示内容にフィルターをかけることができます。

[アウトライナーについてもっと詳しく >](#)

Users (Sharing)

(ユーザー (共有))

データブロックの多くはデータブロック間で共有でき、再利用が奨励されます。例えばあるオブジェクトが「光沢」という名前のマテリアルを持っているとします。同じマテリアルを持つオブジェクトを新たに作るには、オブジェクトのマテリアルパネルで追加/NEWボタンをクリックするのではなく、その隣の小さな上下矢印のあるアイコンをクリックして、既存のマテリアルの一覧から「光沢」を選びます。これで、この2つのオブジェクトは同じマテリアルを共有しています。マテリアル名の隣にある数字「2」は、このマテリアルの利用者が2人いる (利用オブジェクトが2つある) ことを示しています。よく使われる例には他に次のようなものがあります:

- テクスチャをマテリアル間で共有する
- オブジェクト間でメッシュを共有する (クローン)
- オブジェクト間で lpo curve を共有する。例えばすべてのライトを一斉に暗くするような場合です

Fake User

(フェイクユーザー)

忘れてはいけないのはデータブロックが *いま開いている* ファイル内のものに一切リンクされていないとき、Blender はこのデータブロックを保存しないということです。他のファイルからリンクして使うライブラリ用に「.blend」ファイルを作っているのなら、いま開いている (ライブラリの) ファイルからデータブロックが誤って削除されないように気を配る必要があるでしょう。具体的には、データブロック名の隣にある F ボタンを押して、データブロックの「Fake User」を作ってください。これでユーザー数がゼロになることはなくなり、データブロックは削除されません (Blender は他ファイルからリンクされている数は管理しません)。

Copying and Linking Objects Between Scenes

(シーン間でのオブジェクトのコピーとリンク)

シーンをまたいでオブジェクトのリンクやコピーをしたいことがあるでしょう。このためには、リンクやコピーを行いたいオブジェクトを選択して、3D描画領域のヘッダーの Object (オブジェクト) » Make Links (リンク作成) および Make Single User (シングルユーザー化) を使います。Make Links (リンク作成) はシーンをまたいでリンクを作ります。純粋なコピーを作るには、まずリンクを作ってから Make Single User を選び、オブジェクトの独立したコピーを現在のシーンに作ります。シーンでの作業の詳細については [こちら](#) をご覧ください。

Appending or Linking Across Files

(ファイルをまたぐ追加やリンク)

File (ファイル) » Append (アペンド) 機能 (いつでも \diamond ShiftF1 でアクセスできます) を使って別の .blend ファイルの内容に簡単にアクセスし、現在のファイルに挿入できます。.blend ファイル間でのオブジェクトのコピーやリンクについて、詳しくは [こちら](#) をご覧ください。

Proxy Objects

(代理オブジェクト) [Proxy objects](#) は、リンクしたデータやその一部をローカルにコピーします。例えば、アニメーターはこの機能を使ってキャラクターの操作ボーンを、実際のリグを複製せずにローカルに「コピー」できます。キャラクターのアニメーション作成において、外部ライブラリから丸ごと読み込んだキャラクターに対して、アニメーターがポーズやアクションの作業をできるようにしたいとき、特に役立ちます。別の例として、車の形状 (メッシュ) の作業をする造形担当と、車のマテリアルの作業をする塗装担当がいるとします。塗装担当は車の形状を変えられませんが、車の色配置を扱う作業を始めることができます。車の形状の更新があれば、塗装担当の代理オブジェクトに自動的に反映されます。

Pack and Unpack Data

(データの梱包/梱包解除)

普段 .blend ファイルの外に保存されるさまざまなデータをすべて、.blend ファイルに組み入れることができます。例えば画像テクスチャとして使っている外部の .jpg ファイルは、File (ファイル) » External Data (外部データ) » Pack into .blend file (.blend ファイルにパック) を使って .blend ファイルの中に置くことができます。.blend ファイルが保存されると、.jpg ファイルのコピーが .blend ファイルの「中に」置かれます。この .blend ファイルは画像テクス

チャといっしょにどこにでもコピーやメール送信できるようになります。

画像テクスチャがパックされたことは、ヘッダーに小さな「クリスマスプレゼントの贈り物」が表示されるのでわかります。

Unpack Data

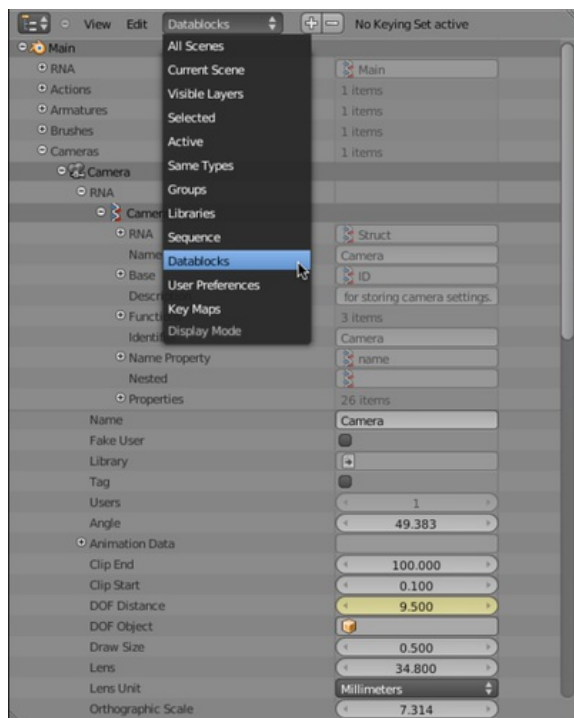
(データの梱包解除)

パックされたファイルを受け取ったら、File(ファイル) » External Data(外部データ) » Unpack into Files(ファイルにパック解除)... を行うことができます。元のディレクトリ構造(original location)を作るか、ファイルを // (.blend ファイルのあるディレクトリ)に置くオプションを選べます。もしテクスチャを編集して、その後に再パックしなおした .blend ファイルを交換する予定なら、そのファイルを送り返して、元の持ち主が(元の場所に)アンパックしたときに彼のテクスチャのコピーも更新されるように、「元の場所」を使ってください。

Datablocks

(データブロック)

どの blender プロジェクトにおいても基本単位になるのがデータブロックです。データブロックの例としてメッシュ、オブジェクト、マテリアル、テクスチャがあります。平面上に単に球が浮いたものであろうと完成した映画であろうと、プロジェクトの定義はプロジェクトが含むデータブロックやそのデータブロックのプロパティ、それにデータブロック同士が相互にリンクする方法によって行われます。プロジェクトをうまく体系化するのに必要な分だけ、データブロックは .blend ファイルに存在できます。



データブロックビュー

データブロックの型:

- RNA
- Filename
- File Has Unsaved Changes
- File is Saved
- Cameras
- Scenes
- Objects
- Materials
- Node Groups
- Meshes
- Lamps
- Libraries
- Screens
- Window Managers
- Images
- Lattices
- Curves
- Metaballs
- Vector Fonts
- Textures
- Brushes
- Worlds
- Groups
- Shape Keys
- Scripts
- Texts
- Speakers
- Sounds
- Armatures
- Actions
- Particles
- Grease Pencil
- Movie Clips
- Masks

Copying and Linking Datablocks

(データブロックのコピーとリンク)

どの型のデータブロックでもコピーやリンクができます。

Scene (シーン) のデータブロックをコピーするには、Info (情報) ウィンドウのヘッダーにあるシーンリストを使ってください。リストはメニューとワークスペースリストの右にあります。現在のシーンのコピーを作るには + ボタンの Add New (新規) を選び、開いたメニューで Full Copy (フルコピー) を選んでください。現在のシーンが新しいシーンに **完全に** コピーされます。

すべてをコピーする代わりに、Link Objects を選んでデータブロックをリンクし、2つのシーンに同じ Object データブロックをリンクして使ったり、Link ObData を選んで、同じ ObData データブロック(メッシュ、曲線、マテリアルなど)を共有するオブジェクトを別に作ったりできます。なお Object データブロックはすべての ObData データブロックの親なので、Link Objects を選ぶと、事実上コピーされるものはほとんどありません。シーン内で起きた編集(オブジェクト位置、メッシュのジオメトリ、...)のほとんどすべてが、この方法でリンクされているもう一方に同じように反映されます。Link ObData に関しては、オブジェクトの "真の" コピーを作りますが、子データブロックのコピーは作りません: オブジェクトの位置、スケール、回転はシーンに固有のものになりますが、形状やマテリアル、テクスチャ、...はそうなりません。(ObData データブロックによる定義通り)。

Copying and Linking Object Datablocks

(オブジェクトデータブロックのコピーとリンク)

Full copy (フルコピー)

◊ ShiftD は選択オブジェクトの通常のコピーを作るのに使います。

オブジェクトと子データブロックの一部は実際に複製されますが、その他の子はリンクされるだけです。複製される属性は User Preferences (ユーザー設定) » Editing (編集) » Duplicate Data: (データの複製) で定義できます。

Linked copy (リンクされたコピー)

AltD リンクされたコピーを作ります。

オブジェクトデータブロックを除くすべてのデータブロックがリンクされます。

Copying and Linking other Datablocks

(他のデータブロックのコピーとリンク)

ObData データブロックがひとつ以上のオブジェクトに使われていると(リンクされていると)、リンクされたオブジェクトの数が記された小さなボタンが設定ウィンドウ(プロパティウィンドウのメッシュ、カーブ、カメラ、...、マテリアルなど)の名前の隣に現れます。クリックすると、現在のオブジェクト用にそのデータブロックのコピーを作ります。

Unlinked Datablocks

(リンクを解除されたデータブロック)

データブロックのリンクを解除 (unlink) することができます。例えば、マテリアルデータブロックがリンクされていたオブジェクトが削除されたときは、そのマテリアルデータブロックのリンクが解除されます。データブロックのリンクが解除されると、デフォルトでは Blender を閉じた時に .blend ファイルから削除されます。リンクを解除されたデータブロックを .blend ファイルに留めたい場合は、プロパティウィンドウのオブジェクト名の隣にある "F" ボタンをクリックします。

Scene Management Structure

(シーンの管理の組み立て)

シーンの管理やライブラリのアペンドリンクは、Blender の [ライブラリやデータシステム](#) に基づいているので、このシステムの基礎に親しんでいなければ、マニュアルページを先に読んでおくのはよい考えです。

Blender はひとつのシーンや画像のような簡単なものから、一本の映画作品全体を作るのにも使えます。映画は通常次の三幕で構成されます:

1. 導入-葛藤
2. 緊張の高まり
3. 山場-解決

各幕にはいくつかのシーンと、話が展開する箇所の設定があります。各シーンは舞台やロケ地のひとつのセットで撮影します。それぞれに小道具や背景が付属します。シーンとは、出演者による(かなうなら説得力のある)演技の連続です。一連の演技や撮影は、通常は2、3秒間続きます。

長回し

1回の撮影が数分に及ぶこともあります。これが「長回し(long take/sequence shot)」で、これだけで完全なシーンになるかもしれません。観客が寝てしまわないようにする高い技術が必要になるでしょう

ひとつの Blender ファイルを体系化して、映画を一本丸ごとおさめられるようにします。ひとつの .blend ファイルには複数のシーンを置くことができます。ひとつのシーンにはレイヤーにわけて整理した、たくさんのオブジェクトがあります。モデリング、小道具の作成、服飾や舞台装飾(マテリアルの割り当て)、演技の定義(アニメーション)、動画のレンダリング、映画の発行といった製作工程の進行にあわせて、その工程を効率よく作業できるように設計された専用の [スクリーンレイアウト](#) を使います。作業設定にあわせてスクリーンレイアウトを調整し、カスタムのレイアウトを作ることができます。

Planning Your Timeline

(タイムラインの設計)

数秒の間にカメラまたは俳優(あるいはその両方が)シーン内を移動することで、シーン内の撮影(Shot)は完了します。Blender のデフォルトの時間計測の単位はフレームで(秒に変えることもできます)、一般的な動画は 25 か 30フレーム毎秒(fps)、映画は 24fps です。したがって5秒間の撮影では最大 $30\text{fps} \times 5\text{秒} = 150\text{フレーム}$ を割り当てます。若干の余裕を持たせるなら、2回目の撮影(Shot 2)は 250フレームから始めることになるでしょう。NTSC 規格用の1シーンが設定された1分の動画は、最終的にはフレーム 1800 枚に至るタイムラインを持つはずですが、2500 フレームにわたって割り付けることができます。こうすれば 700フレームをカットして、遷移時間を除いた最高の 1800フレーム($30\text{fps} \times 60\text{秒} = 1800\text{フレーム}$)を選ぶことができます。

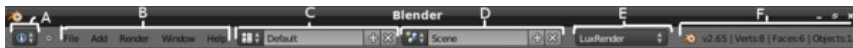
複数台のカメラ

ひとつのシーンを複数台のカメラで別々に撮影し、レンダリング時にどのカメラをアクティブにするか選ぶことができます。[Ctrl + 0] を押して、その時点で使いたいカメラに切り替えてください。

Current Screen Layout and Scene

(現在のスクリーンレイアウトとシーン)

Blender での作業を体系づけるひとつの手段が Scene(シーン)です。シーン間でオブジェクトを共有することができますが、たとえばレンダリング解像度やカメラビューをシーンによって変えることができます。現在のウィンドウレイアウトやシーンは、通常は画面の一番上にある Info(情報)ウィンドウのヘッダーに表示されています。:



Info(情報)ウィンドウのヘッダー。**A)** ウィンドウ種類を示すアイコン、**B)** メニュー、**C)** スクリーンレイアウト、**D)** シーン、**E)** レンダリングオプション
F) 実行中の Blender バージョン(左の Blenderアイコンをクリックするとスプラッシュスクリーンが開きます)。

Loading the UI with “File” → “Open”

(UIをファイルメニューで読み込む)

ファイルを保存すると、Blender は .blend ファイルの中にユーザーインターフェースの配置 - 各スクリーンレイアウトの配置を保存します。デフォルトではこうして保存した UI が読み込まれ、ユーザーのデフォルト設定や現在のスクリーンレイアウトを上書きします。しかし、保存されている UI 設定を無視し、現在の UI 設定を使って作業することができます。Blender を再起動するかメニューからリセットして(ファイル → New または CtrlX)、ファイルブラウザでファイルを開きます(ファイル → Open...または F1)。ファイルブラウザのヘッダーにある Load UI ボタンをオフにしてから、ファイルを開きます。この方法を使うと、新たにファイルを読み込んでも現在のスクリーンレイアウトを引き続き利用できます。

Working with Scenes

(シーンの操作)

シーン名の隣にある上下矢印をクリックして、作業するシーンを選びます。シーンと、そのシーンに含まれるオブジェクトは通常、作業中のプロジェクト固有のものであります。しかし CtrlU を使って、現在の状態をあとで再利用できるように保存することもできます。次に Blender が起動したとき、またはユーザーがファイル → New(新規) (CtrlN) を選んだ時に、保存した状態が復帰します。

Blender はカメラ、照明、立方体をひとつずつ含む、デフォルトのシーンをひとつ備えています。

Adding a Scene

(シーンの追加)

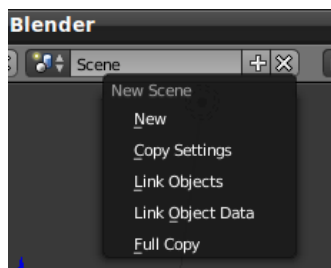
現在のシーンの完全なコピーを作成することも、白紙に戻してやり直すことも、あるいは、現在のシーンへのリンクを持つシーンを作成することもできます。この場合新たなシーンにオブジェクトが現れますが、実際は元のシーンに存在しています。このリンク機能は例えば、元のシーンに舞台を置いて、新たなシーンに俳優や小道具を置くような使い方ができます。

やり直し

新たなシーンで始めたら、忘れないように最初にカメラと照明を追加してください！

シーンはドロップダウンリストにアルファベット順で並びます。別の順序で並べたい場合は、シーン名の先頭を 1- のように順序数にします。シーンへの内部参照は3文字の省略形 "SCE" です。

シーンを追加するには、シーンのリストボタンをクリックして Add New(追加)を選びます。シーンを追加する際に次のオプションを選べます:



シーンの追加時のポップアップメニュー

New(新規)

完全に空のシーンを作ります。

Link Objects(オブジェクトをリンク)

全オブジェクトを新たなシーンにリンクします。オブジェクトのレイヤーや選択フラグはシーンごとに変えることができます。

Link ObData(オブジェクトデータをリンク)

オブジェクトのみを複製します。オブジェクトにリンクされた ObData、たとえば Mesh(メッシュ)や Curve(曲線)は複製されません。

Full Copy(フルコピー)


すべてが複製されます。

通常は、最初のシーン用にデフォルトのフルコピーを作ります。あるいはデフォルトから始めて、あなたを待って行んでいる立方体の編集を開始し、創造

的なことを行えます。

Naming a Scene

(シーンに名前をつける)

シーン名(通常は "Scene.001")を \diamond Shift LMB  クリックするとシーン名を変えることができます。例えば "BoyMeetsGirl" はふつう、三幕の第一幕です。

名前を変えたら、引き続きそのシーンにある小道具やオブジェクトのモデリングを行います。

Linking to a Scene

(シーンへのリンク)

別のシーンにあるどのオブジェクトにも、いつでもリンクできます。リンクしたいオブジェクトのあるシーンを開いて、CtrlL \rightarrow Objects To Scene (オブジェクトをシーンへ) ... を選び、オブジェクトを表示させたいシーンを選ぶだけです。これで元のオブジェクトにリンクされます。特定シーンでこのオブジェクトをシングルユーザー(独立した、リンクされていないもの)にしたいときはそのシーンに行き、オブジェクトを選んで U を押します。解放したいデータブロック(オブジェクト、マテリアル、テクスチャ)を選ぶポップアップが表示されます。

Removing a scene from the file

(ファイルからシーンを取り除く)

現在のシーンはシーン名の隣にある X を押すと削除できます。

アウトライナー・ウィンドウ (Outliner window)

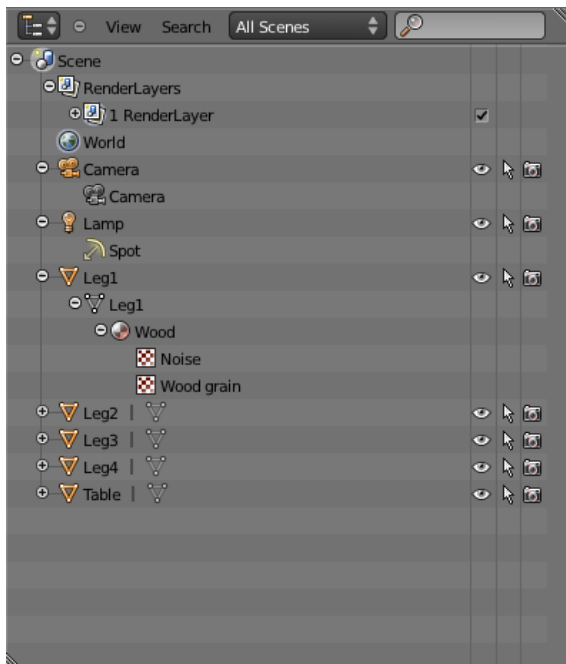
説明

アウトライナー (Outliner) のウィンドウは複雑なシーンを簡単に操作するために用いられます。アウトライナーは複雑な 3D 空間の情報を二次元上に表示させます。シーン中で見つけたい物があるときに利用してください。

例えば、あなたがオブジェクトを移動させている最中にくしゃみをしてしまったと想定してください。マウスが机から飛んでいき (お気の毒にも！) 移動中だったオブジェクトはスクリーン上からどこか彼方へと行ってしまいました。そのオブジェクトを探すにはアウトライナーを利用してください。そのオブジェクトをアウトライナー上で選択し、カーソルを 3D ウィンドウへ戻してカーソルの元へオブジェクトを移動させてください (⇧ ShiftS → Selection to Cursor)。

別の実用的な例は、関連する [データ・ブロック \(datablocks \)](#) の変更の影響を見定めることです。あなたがテーブルの天板となるオブジェクトに注目しており、その `Wood` マテリアルのため出来があまり好ましくなく、もっとマホガニーらしく見せたい、と思索していると想定してください。同じマテリアルは多くのメッシュに利用されているため、あなたがマテリアルの設定を変更することでどれだけの物が変わるのか明確には分かりません。アウトライナーを利用すれば、マテリアルとあなたのシーン中でそのマテリアルにリンクしている全てのメッシュを把握できます。

アウトライナーの操作系

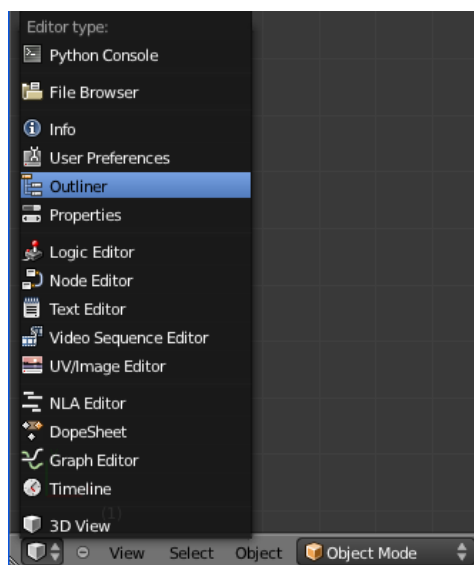


アウトライナーウィンドウ



アウトライナーは互いに関連付けられた物を管理するリストのような物です。アウトライナーでは以下のことが可能です:

- シーン中のデータの閲覧
- シーン中のオブジェクトの選択および選択解除
- シーン中のオブジェクトの表示および非表示
- 3D ビュー内でオブジェクトを選択不可能 (unselectable) にする事による、選択 (Selection) の有効化および無効化
- あるオブジェクトをレンダリング結果に出力するか否かの選択
- シーンからのオブジェクトの削除
- データのリンクを切る (データブロックの名前の側で X キーを押すのと同じ)
- どのレイヤーをレンダリング結果に出力するかの簡単な選択
- どのパスでレンダリング結果を出力するかの簡単な選択 (例えば、Specular パスだけを選択できます)

アウトライナー・ウィンドウを選択



ウィンドウ・タイプを アウトライナー に変更



最初にウィンドウを選択してヘッダの最も左端にある Window type ボタンを LMB  でクリックし、Outliner を選択して LMB  でクリックしてください。

アウトライナーの利用

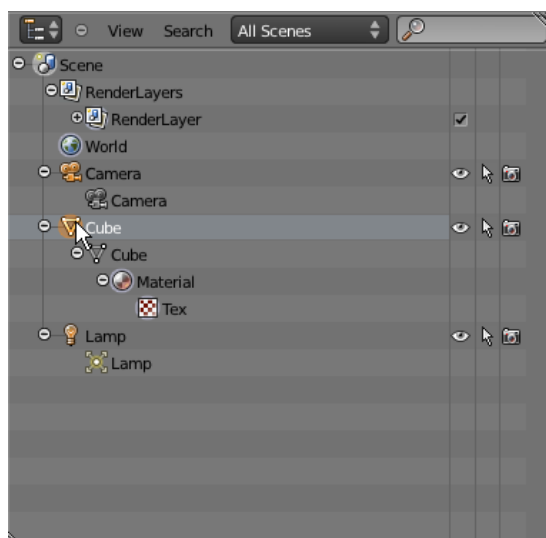
アウトライナーの各行はデータブロックを表示しています。それぞれの名前の左側に位置するプラスの印をクリックしてそのデータブロックが格納している他のデータブロックを確認できます。もしその行が開かれていた場合、アイコンの左隣にあった印はマイナスの印になっています。このマイナスの印をクリックする事でそのデータブロックに付属しているオブジェクトは収納されます。


アウトライナーでデータブロックを選択できますが、必ずしもシーン中のデータブロックが選択されるわけではありません。シーン中のデータブロックを選択するためには、それをアクティブ化する必要があります。

選択とアクティブ化


1つの選択は準選択 (Pre-select) を必要としません。単純に名前とアイコンが表示されている領域の *外* を LMB  でクリック (もしくは RMB  - コンテキスト・メニュー (Contextual menu) から参照) するだけです。

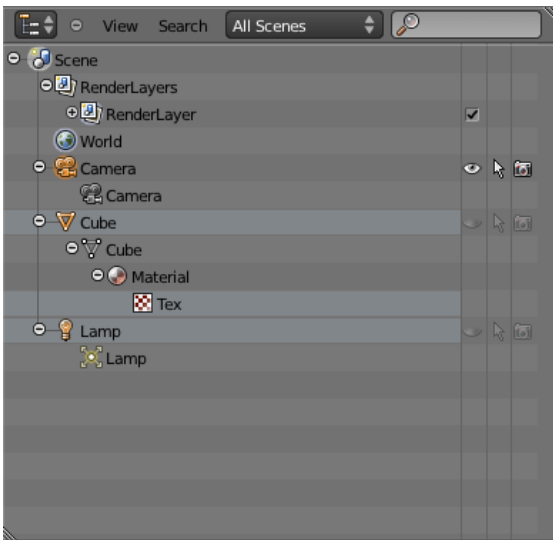
リスト内のオブジェクトをこの方法で選択した場合、そのオブジェクトが選択されて他の 3D ビューではアクティブなオブジェクトになります。3D ビューでオブジェクトを探す際にこの機能を利用する場合、アウトライナーからオブジェクトを選択し、. NumPad キーを押してズームします。テンキーが無い場合は \uparrow ShiftS \rightarrow Cursor \rightarrow Selection そして C キーを押してオブジェクトをあなたのカーソル上に移動させます。



エディット・モードをアクティブにするため、立方体のメッシュ上で LMB  をクリック

1つのデータブロックのアクティブ化







データブロックのアイコンを LMB  でクリックしてアクティブ化して下さい。データブロックをアクティブ化すると自動的に関連するモードに切り替わります。例えば、立方体のメッシュデータをアクティブ化した場合はその立方体が選択されて エディット・モード へ遷移し、立方体のオブジェクトデータをアクティブ化した場合はその立方体が選択されて オブジェクト・モード へ遷移します (右図参照)。




1つのデータブロックの準選択 (Pre-select) を切り替える

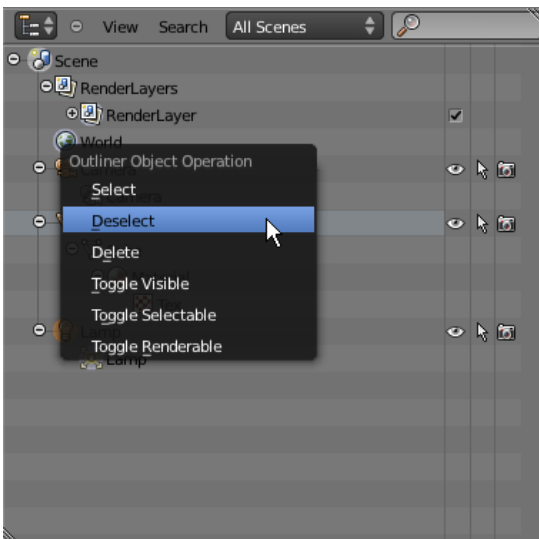
データブロックのグループの準選択の切り替え

データブロックのかたまりを一度に選択あるいは選択解除したい場合に有効です。これを利用するためにあなたはリンクによって選択範囲を準備する必要があります。:

- RMB  もしくは LMB  をクリック
- ⇧ Shift RMB  もしくは ⇧ Shift LMB 
- RMB  を押してドラッグもしくは LMB  を押してドラッグ


名称あるいはアイコンの外側の領域全てに対してです。これらの中で準選択された領域は明るい色で表示されています。

名称とアイコンの領域上で RMB  をクリックする事で表示されるコンテキスト・メニュー (Context menu) で選択あるいは選択解除ができます。



コンテキスト・メニュー (Context menu) 上の Cube オブジェクト

コンテキスト・メニュー (Context menu)

アイコンあるいは名称上で RMB  をクリックする事でコンテキスト・メニューを表示させてください。準選択されたデータブロックによって、以下の全てのもしくは一部の選択肢を選択できます:

- 選択 (Select)
- 選択解除 (Deselect)
- 削除 (Delete)
- リンク解除 (Unlink) - その“所有者”からデータブロックのリンクを解除します (例えばメッシュにおけるマテリアル等)。
- ローカル上に作成 (Make Local) - 選択されたデータブロックの複製物を“ローカル”上に作成します。

Note: ある種のデータブロックでは上記のコンテキスト・メニューが1つも表示されない場合があります !

データブロックの削除

選択されたデータブロックを削除するには X キーを利用してください。

一段階展開する

ツリー・リストを一段階展開するには + NumPad キーを利用してください。

一段階格納する

ツリー・リストを一段階格納するには - NumPad キーを利用してください。

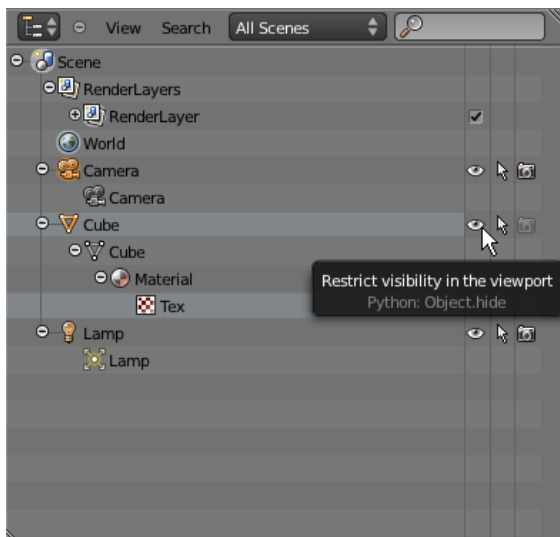
全て展開・格納する

ツリー・リストを全て展開(格納)するには A キーを利用してください。

オブジェクト・レベルの制限の切り替え

以下の3つのオプションは アウトライナー の右側からオブジェクトに対してのみ利用できます:

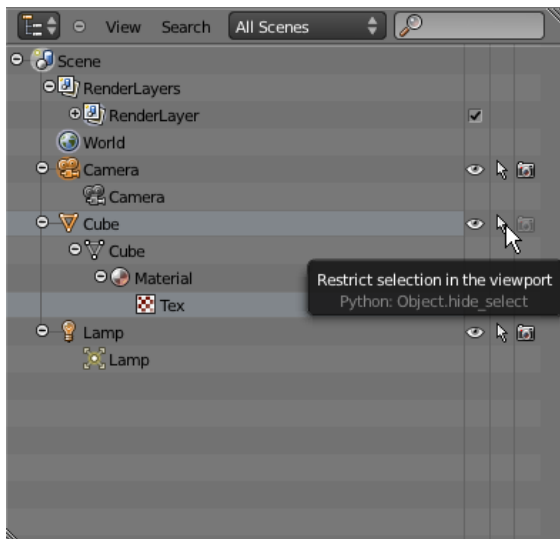
可視性 (Visibility)



可視性 (Visibility) の制限

可視性は アウトライナー においてオブジェクトの右側に有る“目“のアイコンを押すことで切り替えられます。複雑なシーンで、別のレイヤーへ移動させたくないオブジェクトに対して有効です。これは見えるレイヤーでのみ有効です - 見えないレイヤー上のオブジェクトは アウトライナー の設定に関係しない。この設定は V キーを押すことで、アウトライナー で準選択 (Pre-selected) されたオブジェクトに対して切り替えることができます。

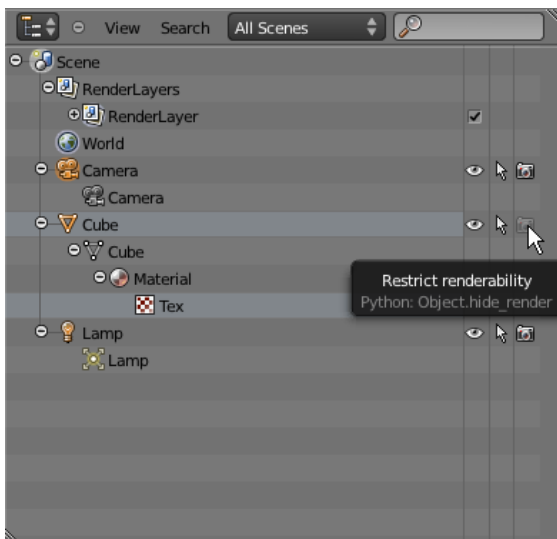
選択可能性 (Selectability)



Restrict selection

“矢印カーソル”のアイコンをクリックするだけで選択可能性は切り替えられます。これはシーン中に意図せずして作業中に選択したくない物が有る場合に有効です。アウトライナー で準選択されたオブジェクトに対して S キーを押すことでこの設定を切り替えることができます。

レンダリング (Rendering)



レンダリング出力の切り替え

“カメラ”のアイコンをクリックする事で切り替えることができます。シーン中のオブジェクトとしては見えたままですが、レンダーにはそのオブジェクトは無視されます。アウトライナー で準選択されたオブジェクトに対して R キーを押すことでこの設定を切り替えることができます。

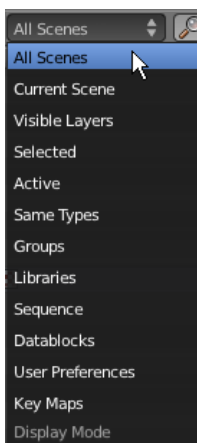
検索 (Searching)

データブロックのファイルを検索することができます。アウトライナー のヘッダ上の Search メニューを利用するか、以下に示すホットキーの1つを利用することで検索が行えます。

- F - 検索 (Find) .
- CtrlF - 大文字・小文字を区別して検索 (Find (case sensitive)) .
- AltF - 完全一致で検索 (Find complete) .
- CtrlAltF - 完全一致かつ大文字・小文字を区別して検索 (Find complete (case sensitive)) .
- ⇧ ShiftF - 再検索 (Find again) .

検索条件に適合したデータブロックは自動的に選択されます。

表示のフィルタリング (Filtering)



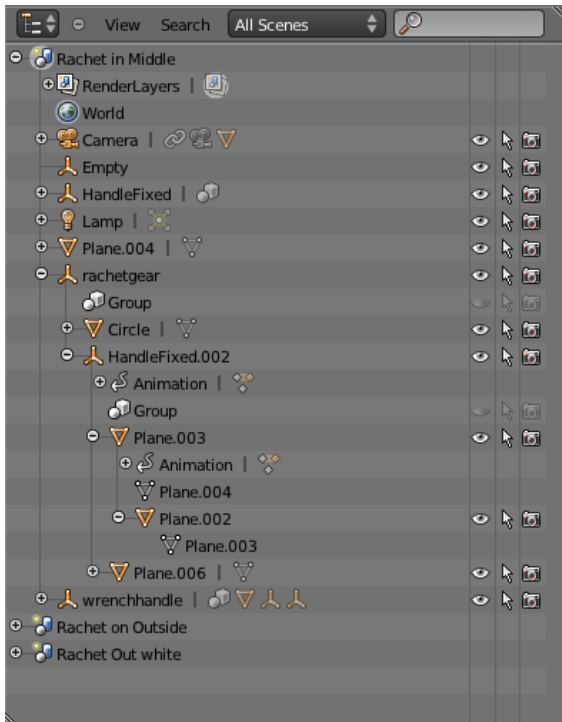
アウトライナー・ウィンドウのドロップダウンメニュー

アウトライナーのヘッダにはアウトラインに何を表示させるかを決定させる文字入力フィールドがあります。標準ではアウトライナーは全てのシーン (All Scenes) を表示しています。あなたは現在のシーンだけ、選択されたオブジェクトだけ、現在選択されたレイヤー上のオブジェクトだけ等々を選択して表示する事ができます。この選択は、あなたが簡単かつ迅速にオブジェクトを見つけられるようにリストを狭める助けとなります。

- 全てのシーン (All Scenes) - アウトライナーにおいて (全てのシーン、全てのレイヤー等で) 表示できる 全てを表示します。
- 現在のシーン (Current Scene) - 現在のシーンで表示できる全てを表示します。3D ウィンドウで可視なレイヤー上にオブジェクトを造るには [layers buttons](#) を利用してください。
- 可視なレイヤーのみ (Visible Layers) - 現在のシーンで可視の (現在選択された) レイヤー上の全てを表示します。
- 選択された物 (Selected) - 3D ウィンドウで現在選択されているオブジェクトだけをリスト上に表示します。⇧ Shift RMB をクリックする事で複数のオブジェクトを選択する事が出来ます。
- アクティブな物 (Active) - アクティブな (往々にして最後に選択された) オブジェクトだけをリスト上に表示します。
- タイプが同じ物 (Same Types) - 現在のシーンにおいて 3D ウィンドウ上で選択されたオブジェクトとタイプが同じ物だけをリスト上に表示します。
- グループ (Groups) - [グループ \(Groups \)](#) とそのメンバーであるオブジェクトのみがリスト上に表示されます。
- Libraries - TODO
- Sequence - TODO

- Data Blocks - TODO
- User Preferences - TODO
- Key Maps - TODO

例



リストモードの アウトライナー ウィンドウ

この図で示されたアウトラインの例では以下の3つのシーンを持つ .blend ファイルを表示しています。即ち “Ratchet in Middle”、“Ratchet on Outside”および “Ratchet Out White”です。各シーンの名称の左隣に有る小さなプラスの印を押すことで、アウトラインは一段階展開されます。図では “Ratchet in Middle” のシーンに対してそれがなされています。ご覧の通り、このシーンは “ワールド (World)” マテリアルの設定、1つの “カメラ (Camera)”、1つの “エンプティ (Empty)”、“HandleFixed”オブジェクト他、追加された全てのオブジェクトを持っています。

“ratchetgear”オブジェクトの隣の小さなプラスの印を押すことで、“Animation”要素に記述されたモーションを持っていることが分かります。他にも “Circle”メッシュを基礎としている事、“Plane.003”の親である “HandleFixed.002”の親である事等が分かります。

ちょっとしたテクニックとして: もしあなたがここに示されたデータブロックを選択すると、それらは 3D ウィンドウでも同様に可能な限り選択されます。マウスカーソルを 3D ウィンドウ上の任意の場所に移動させて * NumPad を押すことで選択されたオブジェクトを中心にしてビューが調整されます。さらに便利な手法として、X を押すことでそのオブジェクトを削除できるように、ほかの全てのホットキーが現在選択されているオブジェクトに対して機能します。

Linked Libraries Overview

(リンクライブラリの概要)

Blender は他の .blend ファイルに「手をつっ込む」ことができ、そこからお望みのものを何でも持ち込めます。この方法で、あなたの Blender データを再利用できます。たとえば、素敵なマテリアルを使った .blend ファイルをライブラリとして持っているものとします。現在開いている .blend ファイルの中にそのマテリアルを追加することができます。これで、設定を手動でいちから再作成する手間を省けます。

General Procedure

(おおまかな手順)

モード: 全モード

ホットキー: CtrlAltO

メニュー: File(ファイル) » Link(リンク)

モード: 全モード

ホットキー: ⇧ ShiftF1

メニュー: File(ファイル) » Append(アペンド)

Blender のメインメニューは Info(情報) ウィンドウにあります(デフォルトでは画面の最上部にヘッダーがあります)。必要な操作は、このメニューの File(ファイル) から Link(リンク) または Append(アペンド) を選ぶか、アクティブなウィンドウ内でショートカットキー(それぞれ CtrlAltO、⇧ ShiftF1)を押すだけです。アクティブウィンドウが File Browser(ファイルブラウザ) の選択ウィンドウに変わります(ウィンドウ種類のアイコンがフォルダになります)。このウィンドウを使ってハードディスクやネットワークドライブにあるフォルダやサブフォルダを見て回り、再利用したいオブジェクトを含む .blend ファイルを見つけてください。.blend ファイルをクリックすると Blender はそのファイルに含まれるデータブロックタイプ(Scenes、Objects、Materials、Textures、Meshes など)を一覧に表示します。クリックすればそのタイプの具体的なインスタンスが表示されます。

Folder and File Organization






(フォルダとファイルの構成)

/lib や /library という名前で作るフォルダをお勧めします。このライブラリの中に、materials、textures、meshes など、あとでアクセスし再利用したいものの種類ごとにフォルダを作ります。ライブラリが大きくなれば、各フォルダ内にサブフォルダを作りましょう。meshes フォルダに作るサブフォルダには、たとえば 人々、宇宙船、家具、建物 などがあるでしょう。次に、椅子メッシュを含む .blend ファイルがあるとすれば、このファイルを 家具 フォルダ内にコピーしてできあがりです。

Appending library objects into your current project

(ライブラリオブジェクトを現在のプロジェクトにアペンドする)

以下の手順で、オブジェクトおよびそこにリンクされているメッシュ、マテリアル、テクスチャのようなすべてのデータを、現在開いている .blend ファイルに追加できます。

1. File(ファイル) » Append(アペンド) を選びます。
2. 追加したいオブジェクトを持つファイルを見つけて選びます(多くの場合「library」ファイル)。
3. そのファイルの Object 欄を開きます。
4. 一覧からオブジェクトをひとつ LMB  で選ぶか、⇧ Shift LMB  で複数選択するか、または LMB  ドラッグでオブジェクトを範囲選択します。LMB  の代わりに RMB  も使えます。
5. 上の手順を、アペンドしたいオブジェクトの種類ごとに繰り返します。Parents と Armatures(すべての Modifier オブジェクト)は分けて選択する必要があります。
6. 左側のLink/Append from Library パネルにある希望のオプションをセットします(Relative Path、Active Layer、...)。
7. Link/Append from Library をクリックするか、↵ Enter を押すか、またはアペンドしたいデータを直接ダブルクリックします。


もちろん、オブジェクト以外にもたくさん、アペンドやリンクできるものがあります: すべての ObData -カメラ、曲線、グループ、光源、マテリアル、メッシュなど-それに一つのシーン全体ですら可能です。... オブジェクトの追加と、メッシュなどのオブジェクトデータの追加には、大きな違いがあることに注意してください。メッシュ データブロックをアペンドすると、持ち込めるのはメッシュの個別インスタンス(particular instance)に関するデータだけで、あなたが見ることのできるメッシュの実オブジェクトインスタンス(actual object instance)ではありません。

あなたのファイル内に、オブジェクトのローカルな独立したコピーを作りたいときは Append(アペンド) を使ってください。元のファイルに動的なリンクを作りたいときは Link(リンク) を使います。リンクすると、元のファイルを誰かが変更すると、現在のファイルは次回開いたときに更新されます。

Link/Append from Library(ライブラリからリンク/アペンド) をクリックすると、いま開いている .blend ファイルにオブジェクトをアペンドまたはリンクします。

その他の読み込みオプションには次のものがあります(File Browser(ファイルブラウザ) の左側にあります):

Select(選択)

読み込まれたオブジェクトはアクティブや選択状態にされません。単に .blend ファイルにポチャんと落とされます。読み込み直後のオブジェクトに何か拡大縮小や移動などの操作をしたいことはよくあります。このボタンを有効にすると、読み込んだオブジェクトは魔法で RMB  クリックしたかのように選択状態になります。このボタンはオブジェクトを見つけ、選択する手間を省きます。

Active Layer(アクティブなレイヤー)

Blender は大きなシーンを分けるための 20 枚のレイヤーを持っていて、各オブジェクトは1つ以上のレイヤーに所属します。デフォルトでは、オブジェクトは元のファイルで所属していたのと同じレイヤーに、直接読み込まれます。現在作業中のアクティブレイヤーだけに読み込むには、このボタ

ンを有効にします。

Relative Path(相対パス)

ファイルを .blend ファイルからの相対指定にします。

Instance Groups(インスタンスグループ)

DupliGroupとして 各グループのインスタンスを作ります。

読み込んだものの探し方

読み込んだオブジェクトが見えないときは、Select(選択)を使ってみてください。Select(選択)を使うなら、カーソルをそのオブジェクトに合わせるスナップツールがあることを思い出してください(⇧ ShiftS4(Cursor to Selected(カーソル→選択物)))。その後ビューをそのオブジェクトの中央に合わせます(⇧ ShiftC(View » Align View » Center Cursor and View All(3Dカーソルのリセットと全表示)))。なお、選択していないレイヤーにオブジェクトがある場合はそのオブジェクトは見えないので、このツールは動作しません。

Reusing Objects (Meshes, Curves, Cameras, Lights, ...)

(オブジェクトの再利用(メッシュ、曲線、カメラ、光源、...))

ある .blend ファイルで作った車輪を、現在のプロジェクトで再利用したくなるとします。車輪はメッシュで再現されており、おそらくタイヤとリムできているでしょう。たぶんあなたはこのメッシュに手頃な名前をつけています、たとえば...えーっと何だろう「wheel」? 車輪は着色されていて、つまりマテリアルがいくつか(ゴムやクロム)割り当てられています。

Objectのデータブロックでそのファイルを見つけて「wheel」を選ぶと、現在開いているファイルに持ち込むことができます。コピーを持ち込むことも、単にリンクすることもできます。

リンク

オブジェクトにリンクして、あとで元のファイルの同オブジェクトを編集すると、現在開いているファイルでは次回ファイルを開いたときに(編集後の)今のオブジェクトの姿が表示されます。

他のアーティストがパブリックドメインで公開しているモデルや、友人たちが .blend ファイルを投稿したり電子メールで送ったりしあって共有しているモデル、あるいはあなたの過去のプロジェクトを、あなたの PC やサーバーの Download ディレクトリに保持しておく、車輪を新たに作り直す手間を避けられるでしょう。

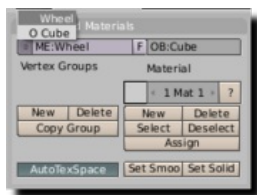
リンクしたオブジェクトは、選択すると水色で縁取りされます。選択状態のオブジェクトの縁取りは、通常はピンクです。

なお、リンクしたオブジェクトは動かせません! オブジェクトは元のファイルでの位置と同じ位置にあります。移動/拡大縮小/回転を行うには、[Proxy\(代理オブジェクト\)](#)に変えてください。

アペンドリンクしたメッシュデータの使用

メッシュデータのようなリソースをアペンドかリンクしたとき、3D ビューポートにすぐに表示されないかもしれません。これは、データは Blender に読み込まれたものの、オブジェクトに割り当てられていないためで、割り当てによって見えるようになります。Outliner ウィンドウで Display Mode を Datablocks 等に切り替えることで、目に見えないメッシュがファイル内に存在することが確認できます。

(訳注: 原文にある、オブジェクト間の繋がりを示す OOPS Schematic ビューは、バージョン 2.5 以降の Blender にはありません)



新たに読み込んだ wheel メッシュをオブジェクトに割り当てるには、いま見えているいずれかのオブジェクトを選ぶか、新たにオブジェクト(cube等)を作ってから、プロパティエディタの Object Data コンテキストを開き、メッシュのドロップダウンで wheel メッシュを選びます。この時点でオブジェクトへの割り当てが完了し、見えるようになるはずですが。

メッシュにアペンドリンクせずにオブジェクトを Blender に読み込めば、メッシュとオブジェクトを関連付けることなしに(もう関連付けられているので)、すぐに 3D ビューポートに表示されます。

Reusing Material/Texture Settings

(マテリアル/テクスチャ設定の再利用)





Image Browser でのマテリアルのプレビュー

(訳注:これは Blender 2.4 の説明です。確認する限りでは、Blender 2.6/2.7 のリンク/アペンド時のファイルブラウザでは、プレビュー表示はできないようです...)

ガラスやクロムのようなマテリアルには、ぴったり取得するのが非常に難しいものがあります。[Blender Foundation](#) がリリースしたものの中には、たとえば公式サイトから無料でダウンロードして利用できる [Materials CD](#) があります。この CD にある .blend ファイルを使うと、ガラスやクロム、木やバナナなどの一般的なマテリアルを取り込むことができます。マテリアルの再作成のためだけに小さなボタンやスライダーを操作する必要がないことを考えると、この機能は大幅な時間の節約になります。バナナのマテリアルは、カラーランプやプロシージャルテクスチャを使った簡単なプロシージャルマテリアルのよくて例で、見た目がとてもリアルです。これと呼び出してみましょう。ファイルを見つけた際に、Material を選択すると、ブラウザにマテリアルの球状サンプルが表示され、名前とともに、選んだテクスチャの確認をやすくします

Reusing Node Layouts

(ノードレイアウトの再利用)

ヌードル(ノードレイアウト)を再利用するには、元のファイルを開いて再利用したいノードのセット用にグループを作ります。現在開いているファイルにこのノードグループを持ち込みたいときは、File » Append(アペンド) または File » Link(リンク) を選んでそのファイルを探します。ファイル内に潜り込めば、NodeTree が見つかります。LMB  でそのファイルにあるノードグループの一覧が開きます。目的のノードグループを LMB  してください。

[\[Verse\(英文、ページがなくなっています\)\]](#)

Verse はオープンソースの素晴らしいコラボレーションツールで、Blender と統合されています。複数人での Blender ファイルでの作業やリンク、それにオブジェクトや編集の共有がリアルタイムでできるようにします。

Proxy Objects

(代理オブジェクト)

Proxy とは正規の代理人、あるいは代役です。Blender では、リンクされた複製(上述)を作ると、このオブジェクトは編集できません。そこにあるのは本体へのリンクだけです。本体は現在開いていない別のファイル内にあるので、リンクには何も追加できず、変更もできません。

チームで作業をしていると、もっと柔軟さがほしくなるでしょう。たとえば車をモデリングしているとして、車の形状(メッシュ)を一人が作り、有用なカラースキーム(マテリアル)を別の人が作っているとします。この場合塗装担当者には代理オブジェクトを渡して、マテリアル設定の変更ができるようにしたいでしょう。もっと一般的な例では、あるキャラクターのアニメーション製作を担当するチームがいます。このチームはポーズの定義はできますが、キャラクターの色やアーマチュアの変更はできません。リグ担当者が定義したものだけを使います。

代理(proxy)オブジェクトの重要な性質は、ローカルにデータ編集ができる一方で、特定のデータに制限をかけたままにできる点です。制限つきで定義されたデータは常にライブラリから復元されます(通常はファイルの読み込み、操作取り消し、再実行時)。この制限は参照されるライブラリ自身で定義されます。つまり、ローカルで変更可能なものを定義できるのは、ライブラリファイルだけであるということです。

ポーズに関しては、ボーンレイヤーを Protected(保護)に指定することで、この制御ができます。

モード: オブジェクトモード

ホットキー: CtrlAltP

自分で代理オブジェクトを作るには、上記手順により元のオブジェクトにリンクを作ってください。リンクしたコピーを選択した状態で CtrlAltP を押して、Make Proxy(プロキシ作成) ダイアログを確定します。オブジェクトの名前は元の名前に接尾子「_proxy」をつけたものになります。これで代理オブジェクトを動かしたり編集したりできます。選択すると、ローカルのオブジェクトのように見えます(オレンジ色で縁取られます)。

こうして制限されていないデータを編集できます。ほとんどのオブジェクトでは、制限されていないデータに位置や回転が含まれます。Ipo カーブを使ってオブジェクト位置をアニメートさせることもできます。メッシュオブジェクトについては、メッシュの形状に制限があるためシェイプキーを定義できません。ファイルを開き直すと、Blender は元の制限つきデータに施された変更をすべて、あなたのファイルに反映させますが、あなたの行った変更はリセットしません(所有者が行わない限り)。

Armatures and Multiple instances

(アーマチュアと複数インスタンス)

この機能は開発作業中です。Blender 2.43 と CVS (2007年4月29日現在)では、代理オブジェクトは *グループのすべてのインスタンス* を制御します。グループインスタンスごとに代理オブジェクトを持つことはまだできません。特に、グループインスタンスごとにひとつ代理アーマチュアを持つことはまだできません。部分的に有効な救済策は、複数のインスタンス複製については、ファイルをリンクする代わりにアペンドして使うことです。ファイルのアペンドは元のファイルが更新されても更新されません。

POSIX 準拠のファイルシステムを使っていれば、[Linked Lib Animation Madness\(英文、リンク切れ\)](#) に書かれている手軽なハックを使うと、1グループ1代理オブジェクトの制限を回避できます。

はじめに

Blender を利用する際は、4つの次元で形成される世界を作ることになります:

1. 左右に伸びる次元、一般に“x”軸と呼ばれます。
2. 前後に伸びる次元、一般に“y”軸と呼ばれます。
3. 上下に伸びる次元、一般に“z”軸と呼ばれます。
4. アニメーションで動作するオブジェクト、マテリアル、フレーム内で定義された動きによって形成される時間の次元。

ここで問題となるのは、あなたの目の前に2次元で表現を行うコンピュータのスクリーンが有る事です！カーソルは上下もしくは左右にのみ移動可能です。時間をさかのぼることや、読んで字のごとくにスクリーン内のオブジェクトを掴んでどこかへ移動させる事もできません。(鏡面を通り抜けて男が出現するホラー映画が有りますが、それはまた別の話です。)

代わりに、Blender に指示を与えることでそれが可能になります。この章では Blender のインターフェースを用いてどのようにあなたの仮想空間を操作するのかということについて触れます。Have a nice trip !

はじめに

Blender は二次元の表現のみ可能なモニタスクリーンを利用して三次元空間での作業を可能にします。三次元空間で作業するようにするために、あなたはシーンを見たい方向から見られるように視点を変更する事ができます。この操作は全ての3D Viewport で可能です。私達は 3D View のウィンドウについて説明しますが、その他のほとんどのウィンドウでも同様の機能を利用できます。For example, it is possible to translate and zoom a Buttons window and its panels.



マウスのボタンとテンキー

もしあなたが2つ以下だけのボタンを持つマウスやテンキーを持たないキーボードを利用している場合、このマニュアルの [Keyboard and Mouse](#) のページを参照し、そのような機器で Blender をどのようにして使うかを調べてください。

ビューを回転させる

モード: 全てのモード

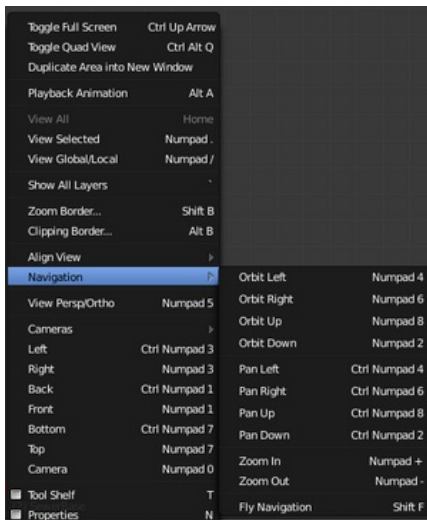
ホットキー: MMB / 2 NumPad / 4 NumPad / 6 NumPad / 8 NumPad / CtrlAlt Wheel

メニュー: View → Navigation

説明

Blender では4つの異なる方向からの視点を利用できます。即ち側面、正(背)面、天(地)面そしてカメラのビューです。Blender は Z 軸が上方へ伸びた直交座標系 (デカルト座標系) を利用し、側面が X 軸に、手前に向かって正面が Y 軸に、天面が Z 軸に対応しています。カメラのビューは現在のシーンをカメラの視点から見た光景に対応しています。

補足



3D Viewportにおける View のメニュー

View メニュー(3D viewport のViewメニュー)から選択する、あるいは3 NumPad で右側面を、1 NumPad で正面を、7 NumPad で天面をという具合にホットキーを利用する事で、3D Viewport で見る方向を選択する事ができます。このテンキーのショートカットは Ctrl を押しながら利用することでそれぞれ逆側からの視点を参照できます。最後に、0 NumPad を押すとカメラからの視点を参照できます。

ホットキーについて

ホットキーが、フォーカスされているアクティブなウィンドウに対して影響を及ぼすという事を記憶しておいてください。ホットキーを利用する前にカーソルが、あなたが作業を行いたい領域内に存在するか確認してください！

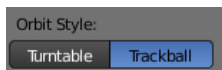
これら4つの標準で定義されている方向とは別に、視点は任意の方向に変えることができます。Viewport の領域内で MMB を押しながらドラッグしてください。ウィンドウの中心辺りでドラッグを始めて上下左右にカーソルを動かすと、View がウィンドウの中心を基準に回転します。代わりに Alt を押したまま LMB を押しながらドラッグすることでも Viewport の領域の視点を変えることができます。

視点の角度を一定の度合いで段階的に変化させるためには、8 NumPad および 2 NumPad キーを利用してください (これは鉛直方向の MMB を押しながらのドラッグに対応しています)。あるいは4 NumPad および 6 NumPad キーもしくは CtrlAlt Wheel を同時に利用することで View ウィンドウで何を選択していても、シーンの Z 軸まわりの回転で視点を変更できます。


トラックボールとターンテーブル

標準では、View をこの章で既に述べた方法で変化させる、他にも“トラックボール”を利用して手で回転させるようにシーン全体を回転させる事ができます。一部のユーザーにとっては直観的だと感じる事ができる操作です。この方法で 3D View のウィンドウを回転させることに違和感を感じる場合は“ターンテーブル”を利用する方法に切り替える事もできます。

ターンテーブルは2つの軸に対する回転を実現できるレコードプレイヤーのような形をした物で、3D空間における“上”や“下”といった物の定義を理解しやすくなります。ターンテーブルの欠点は、オブジェクトの操作の柔軟性がいくばくか失われる事です。しかしながら、方向感覚をつかみにくい時には、“上”や“下”を認識する助けになります。当然ながらあなたはあなた自身が取り組む仕事に応じて操作の方法をいつでも切り替えることが可能です。



設定画面

ここまで述べた回転操作の"方法"を変更するためには [User Preferences ウィンドウ](#) を参照してください。その画面上の Input ボタンを押し、入力方法に関連するボタン類が集まったページを表示させてください。そこに回転方法 (Orbit style) を選択する部分があります。User Preferences の Interface タブ内にも3Dウィンドウにおける視点の操作方法を設定する2つのチェックボックスがあります。Interface タブ内の Auto Perspective が有効な場合は MMB  を押し、View を操作する際に自動的にパースペクティブを切り替えます。Rotate Around Selection が有効な場合は現在選択している集合の中心を基準に View を回転させます (3D View 上で A キーを使えばすべての選択を無効にできます)。どんな場合でも最後に選択された集合が利用されます。



View のパン(移動)

モード: 全てのモード

ホットキー: ◊ Shift MMB  / Ctrl2 NumPad / Ctrl4 NumPad / Ctrl6 NumPad / Ctrl8 NumPad / ◊ ShiftAlt LMB 

メニュー: View → Navigation

説明

View をパンするためには、3D Viewport 内で ◊ Shift キーを押しながら MMB  を押してドラッグします。視点の角度を一定の度合いで段階的に変化させるには Ctrl8 NumPad、Ctrl2 NumPad、Ctrl4 NumPad、そして Ctrl6 NumPad というホットキーを回転させる時に利用してください (note: Ctrl キーを ◊ Shift キーに置き換えて利用することもできます。)。マウスの中ボタンを利用しない場合は ◊ Shift Alt を押しながら LMB  を押してドラッグします。


View のズーム(拡大縮小)


モード: 全てのモード

ホットキー: Ctrl MMB  / Wheel  / + NumPad / - NumPad

メニュー: View → Navigation

説明

あなたは Ctrl キーを押しながら MMB  によるドラッグを行うことでズームインおよびズームアウトができます。ホットキーは + NumPad 及び - NumPad です。3D View のヘッダの View → Navigation サブメニュー内からもこの機能を利用することができます (3D viewport の View メニューを参照してください)。

あなたがホイールマウスを利用している場合は、ホイール Wheel  を回転させることで + NumPad キー及び - NumPad キーの操作を行うことができます。また Buttons ウィンドウは多くのパネルを持っているため、ホイールの回転によって水平方向のウィンドウ内の移動が可能になります。これによって Buttons ウィンドウの範囲を狭めたり小さくしたりしても必要なパネルの所へ移動する事ができます。さらに、Buttons ウィンドウを鉛直方向に表示することも可能で、この場合はウィンドウ内のパネル類は鉛直方向に表示されるようになります。

迷子になった時

3D 空間内で自分の位置が分からなくなったときは、一般的では有りませんが2つのホットキーが助けになります。⌘ Home キーを押すことで全てのオブジェクトが見えるようになります (View → View All メニューも同様の機能を持っています)、. NumPad キーを押すことで選択されているオブジェクトに視点をあわせるようになります (View → View Selected メニューも同様の機能を持っています)。

はじめに

Blender の使用する3次元空間で作業をするためには、観測者の目の位置(観点、viewpoint)やシーンへの視線方向を変えることができません。ここでは 3Dビュー ウィンドウについて述べますが、他のウィンドウのほとんども同じ機能を持っています。例えば ボタンウィンドウやそのパネルも移動やズームできます。

💡 マウスボタンとテンキー

ボタンが3つ未満のマウスやテンキーのないキーボードをお使いなら、Blender でこうした機器を使う方法がマニュアルの [キーボードとマウス](#) のページにあります。

透視投影 (Perspective) と 平行投影 (Orthographic)

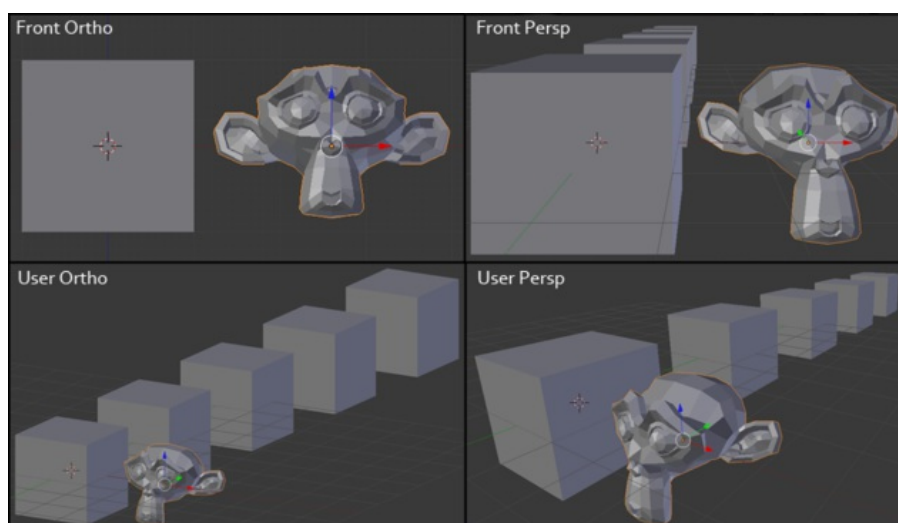
モード: 全モード

ホットキー: 5 NumPad

メニュー: View(ビュー) » Perspective(透視投影) / View(ビュー) » Orthographic(平行投影)

解説

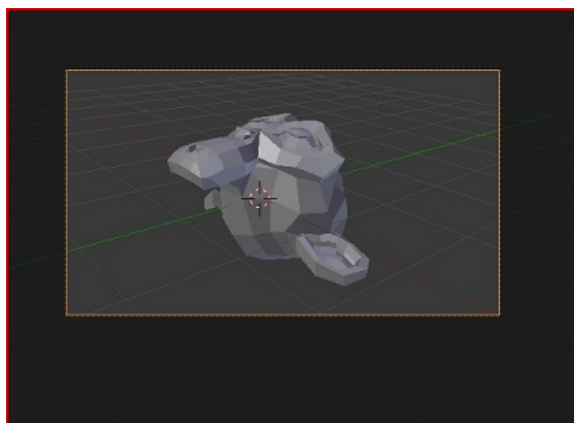
3Dビューポートごとに、2種類の投影方式のいずれかを選ぶことができます(下図「平行投影(左)と透視投影(右)」)。



平行投影(左)と透視投影(右)

透視投影は我々が見慣れているもので、遠くにあるものほど小さく映ります。平行投影では、ものの距離が変わっても大きさは同じままで、一見すると少し変です。ちょうど、シーンを無限遠から眺めているようなものです。それにもかかわらず、平行投影はシーンの「実用的な」視界を作り、比率の反映や見積もりをしやすくするので大変役立ちます(Blenderや他のアプリケーションのほとんどで、平行投影がデフォルトです)。

オプション



カメラビューのデモ

3D ビューの投影方式を変えるには、View(ビュー) » Orthographic(平行投影)またはView(ビュー) » Perspective(透視投影)メニューを選びます。ショートカット5 NumPad この2つのモードの切り替えができます。3Dビューの投影方式の変更はシーンのレンダリング方法には影響しません。レンダリングはデフォルトでは透視投影で行われます。平行投影でレンダリングする必要がある場合は、カメラを選択し、オブジェクトデータコンテキストを開いてレンズ(Lens)パネルにある平行投影 ボタンを押します。

ビュー » カメラメニューは 3Dビューをカメラモードにします(0 NumPad)。シーンは、後にレンダリングされるのと同じように表示されます(図「カメラビューのデモ」をご覧ください)。レンダリングした画像には、オレンジの点線の内側にあるものがすべて入ります。このビューでズームインやズームアウトを行えます。

すが、観点 (viewpoint) を変えるには、カメラを移動させるか回転させる必要があります。

大きなシーンでは、シーンにあるオブジェクトがすべてカメラビューに表示されないかもしれません。考えられる原因の一つはカメラの[クリッピング距離 \(clipping distance\)](#)が近すぎることです。カメラはクリッピング範囲内のオブジェクトだけを表示します。

[レンダリングの透視図法についてもっと詳しく»](#)

[カメラビューについてもっと詳しく»](#)

[カメラのクリッピングについてもっと詳しく»](#)

技術的な詳細

透視投影の定義



透視投影ビューは、シーンを3次元に置き、観察者を点Oに置くことで幾何学的見地から作られます。2次元の透視投影を行うには、まず点Oの前に視線方向に直交するように、2次元のシーンを描く平面 (例: 紙) を置きます。3Dシーンの各点Pに対してOとPを通る直線POが描かれます。この直線POと平面の交点Sが、透視投影の行われる点です。シーン内のすべての点Pを投影したものが、透視投影ビューです。

平行投影の定義

平行投影ビューには、視線の方向はありますが観点Oはありません。視線方向に平行になるように、点Pを通る直線が引かれます。この直線と平面の交点Sが、点Pの平行投影点になります。シーン内のすべての点Pを投影したものが、平行投影ビューです。

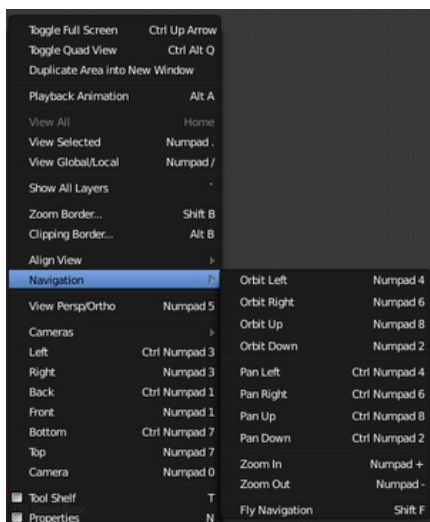
ビューの回転

モード: 全モード

ホットキー: MMB  / 2 NumPad / 4 NumPad / 6 NumPad / 8 NumPad / CtrlAlt Wheel 

メニュー: View(ビュー) » Navigation(ナビゲーション)

解説







3D ビューポートのビューメニュー

Blenderにはデフォルトのビュー方向が4つあります。側面図 (Side)、正面図 (Front)、上面図 (Top) とカメラビュー (Camera) です。Blender は Z 軸が上を向く「直交デカルト座標系」を使います。「側面図」は X軸上で負の方向を見る視点に一致します。「正面図」は Y軸上にあり、「上面図」は Z軸上にあります。カメラビューはカメラの視点から見えるままのシーン表示します。

オプション

3D ビューポートの視線方向は ビューメニューか、次のホットキーで選べます: 3 NumPadで「側面図」、1 NumPadで「正面図」、7 NumPadで「上面図」。ショートカットを使うときにCtrl を押し続けていると、逆側のビューを選べます。最後に、0 NumPad は「カメラ」の観点 (viewpoint) に入ることができます。

ビューは4つのデフォルトから外れた方向にも、好きな角度に回転できます。ビューポート内を MMB ドラッグします。ウィンドウの真ん中から始めたとして、上下や左右に動かすと、ビューはウィンドウの真ん中を中心にして回転します。ユーザー設定で3ボタンマウスを再現 (Emulate 3 button mouse) オプションを選んでいる場合は代わりに、Altを押しながらビューポート内を LMB ドラッグします。

視線を一定の刻み幅で回すには、8 NumPad と 2 NumPad を使ってシーンをグローバルの X軸周りに回すか (縦方向への MMB ドラッグと同じです。観点 (viewpoint) がどこにあっても使えます)、もしくは 4 NumPad や 6 NumPad (または CtrlAlt Wheel ) を使って、シーンを グローバルの Z軸周りに回します。

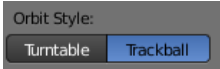
ホットキー

ほとんどのホットキーはアクティブなウィンドウ (フォーカスを持つウィンドウ) に作用することに注意してください。作業したい場所にマウスカーソルがあることを確認してからホットキーを使ってください。

トラックボール/ターンテーブル

上述のようなビューの回転は、デフォルトでは「ターンテーブル」スタイルで行われます。これを直感的だと感じる人もいれば、そう思わない人もいます。もしこのウィンドウの回し方が難しいと思ったら、「トラックボール」スタイルに変えられます。トラックボールスタイルを使うと、トラックボールに手を置いて回しているかのようにシーンを回します。

ターンテーブルスタイルは、回転軸が2つあるレコードプレイヤーに似ており、その世界で何が「上」で何が「下」なのかをうまく定義しているように見えます。ターンテーブルスタイルの欠点は、オブジェクトを扱う際の柔軟性が幾分失われることです。しかしながら、方向感覚を失ったときに役立つ「上下」の感覚を得られます。もちろん、作業内容によっていつでもスタイルを切り替えることができます。



ビューの回転

回転のスタイルを変えるには [ユーザー設定ウィンドウ](#) を使います。入力 (Input) タブに、回転スタイル (Orbit Style) のオプションが見つかるでしょう。ユーザー設定のインターフェイス (Interface) タブには2つのチェックボックスがあり、3D ウィンドウでの表示を決めることができます。自動パース (Auto Perspective) は MMB でビューを回転させるときは常に、自動的に透視投影に切り替えます。選択範囲を中心に回転 (Rotate Around Selection) は現在の選択を中心にしてビューを回転します。何も選択されていないときは (例えば A で選択を解除したときには)、直前に選択していた範囲が使われます。

ビューのパン

モード: 全モード

ホットキー: Shift MMB / Ctrl2 NumPad / Ctrl4 NumPad / Ctrl6 NumPad / Ctrl8 NumPad / ShiftAlt LMB

メニュー: View (ビュー) » Navigation (ナビゲーション)

解説

ビューをパンするには、 Shift を押しながら 3D ビューポート内を MMB でドラッグします。刻み幅をつけてパンするには、回転に合ったホットキー Ctrl8 NumPad、Ctrl2 NumPad、Ctrl4 NumPad、Ctrl6 NumPad を使います (註記: Ctrl の代わりに Shift も使えます)。中ボタンのないマウスでは、 Shift Alt を押しながら LMB でドラッグします。

ビューのズーム

モード: 全モード

ホットキー: Ctrl MMB / Wheel / + NumPad / - NumPad

メニュー: View (ビュー)

解説

Ctrl を押しながら MMB ドラッグすると、ズームイン/アウトできます。ホットキーは + NumPad と - NumPad です。View (ビュー) » Navigation (ナビゲーション) サブメニューにもこれらの項目があります。詳細は上の 3D ビューポートの ビュー メニューの画像を参照してください。

マウスホイールがあれば、3D ビューポートで + NumPad や - NumPad で行う操作を、Wheel を回して行えます。ボタン ウィンドウをズームするには、Ctrl MMB を押して、マウスを上下に動かします。

迷子になったら...

珍しいことではありませんが、3D 空間で迷子になったら、2つのホットキーが助けてくれるでしょう。⌘ Home は、すべてのオブジェクトが見えるように視点を調整してくれます (View (ビュー) » View All (すべてを見る) メニュー)。透視投影モードにいれば、. NumPad は、現在選択しているオブジェクトにあわせて視点をズームします (View (ビュー) » View Selected (選択物を表示) メニュー)。

ボーダーにズーム (Zoom Border)

ボーダーにズームツールは、ズームインしたい矩形範囲を指定すると、3D ビューがその範囲でいっぱいになるまでズームインします。

この機能は ビューメニューかショートカット ShiftB で実行し、ズームインしたい範囲をドラッグします。

ドリー (Dolly the View)

モード: 全モード

ホットキー: Ctrl Shift MMB

解説

何かを近くで見たいとき、ほとんどの場合はビューをズームさせれば十分でしょう。ただ、ある点を境に、それ以上近くにズームできなくなります。

この理由は、Blender が回転とズームに使われる観点 (view-point) を記憶しているためです。ほとんどの場合これはうまく動作しますが、観点を別の場所に移動したくなる場合もあります。ドリーはこのための機能で、ビューをある場所から別の場所まで運ぶことができます。

(訳注: ドリー (dolly, dolly camera) はレール上の台車にカメラを搭載したもので、被写体の動きに合わせて動くことができます)

Ctrl+ Shift を押しながら MMB をドラッグすると、ドリーを前後に動かします。

視点を揃える

視点を揃える (Align View)

ビューの位置や方向をさまざまな方法で揃えるためのオプションです。この機能は ビューのメニューにあります。

選択物に視点を合わせる (Align View to Selected) メニュー

選択したオブジェクトのローカルの特定の軸にビューを合わせます。編集モードでは、選択面の法線にビューを合わせます。

上 ⇨ Shift7 NumPad
 下 ⇨ ShiftCtrl7 NumPad
 前 ⇨ Shift1 NumPad
 後 ⇨ ShiftCtrl1 NumPad
 右 ⇨ Shift3 NumPad
 左 ⇨ ShiftCtrl3 NumPad

3Dカーソルのリセットと全表示 (Center Cursor and View All) (⇨ ShiftC)

カーソルを原点に戻し、シーンにあるすべてがビューに入るようにズームイン/アウトします。

アクティブカメラを選択に合わせる (Align Active Camera to View), CtrlAlt0 NumPad

現在の観点 (viewpoint) にアクティブなカメラを合わせます。

選択部分を表示 (View selected), . NumPad

現在選択しているオブジェクトをビューポートの中央に配置し、画面いっぱいになるまでズームします。

3Dカーソルを視点の中心に (Center view to cursor), Ctrl. NumPad

3Dカーソルを視点の中心にします。

選択部分を表示 (View Selected)

上の説明をご覧ください。

全てを表示 (View All) \ Home

シーンにあるすべてのオブジェクトをフレームにおさめます。

ローカルビューとグローバルビュー

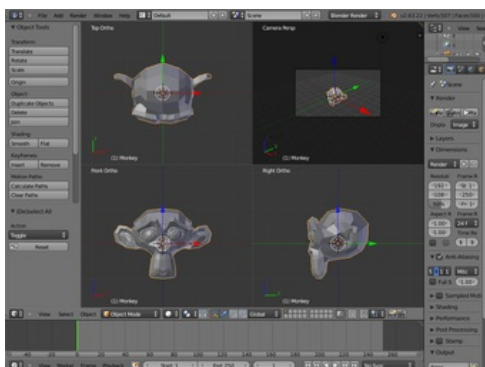
ビューメニューからオプションを選ぶか、/ NumPad を使ってローカルと グローバルの切り替えができます。ローカルビューは選択オブジェクトを隔離して、ビューポートにそれ以外のものを表示しないようにします。これは他のものに隠されているオブジェクトや、重いジオメトリを持つオブジェクトでの作業に役立ちます。/ NumPad を押すとグローバルビューに戻ります。

四分割表示 (Quad View)

モード: 全モード

ホットキー: CtrlAltQ

メニュー: ビュー » 四分割表示 (Toggle Quad View)



四分割表示

四分割表示は 3D ウィンドウを、上面図/平行投影、正面図/平行投影、右側面図/平行投影、カメラ/透視投影の、4つのビューに分割します。このビューを使うと、モデルを複数の観点からすぐに見ることができます。この配置ではズームやパンをビューのそれぞれに行えますが、ビューの回転は行えません。これは、ウィンドウを自分で分割して並べるのとは違うことに注意してください。四分割表示では、4つのビューは1つの 3Dウィンドウの一部として固定されています。ビューをそれぞれ回転させたいときは、3D ウィンドウを分割する必要があるでしょう。

[ウィンドウの分割についてもっと詳しく»](#)

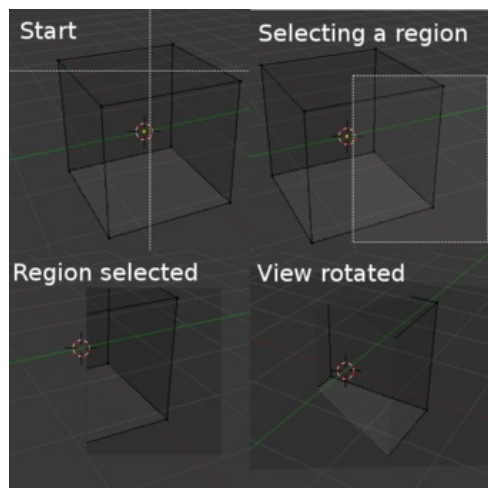
矩形でクリッピング (View Clipping Border)

モード: 全モード

ホットキー: AltB

メニュー: View(ビュー) » Set Clipping Border(矩形でクリッピング)

解説



範囲/ボリュームクリッピング

複雑なモデルやシーンでの作業を支援するために、ビューをクリップして作業対象を視覚的に分離することができます。



クリップを使うと、指定したボリューム内にあるものしか見えなくなります。ペイント、スカルプト、選択、トランスフォームのスナップなどのツールも、クリップ領域の外側のジオメトリを無視します。

AltBで開始し、3D ビュー内にマウスで四角形を描きます。できたクリッピングボリュームは次のいずれかになります：

- 平行投影ビューなら、無限の長さの直方体。
- 透視投影ビューなら、無限の高さの四角錐(ピラミッド型)。

クリップを削除するには AltB を再度押します。

例

画像「範囲/ボリュームクリッピング」は立方体にクリップを使った例です。AltB でクリップを始めると、十字の点線カーソルが現れます(図の左上)。四角形の範囲を LMB  ドラッグして決めます(図の右上)。これで範囲が定義され、3D 空間でのその範囲が(ピラミッド型をしています)クリップされます。立方体の一部が見えなくなる(クリップされている)ことに注意してください。MMB  を使ってビューを回転すると、ご覧のように、ピラミッドの内側にあるものだけが見えています。編集ツールはどれも、ピラミッド型のクリップの内部にだけ普段通りに機能します。

濃い灰色の部分はクリッピングボリューム自身です。新たにクリップを開始するとAltB 現在のクリップは消え、3D 空間にあるものがすべて再び見えるようになります。

視点の操作

モード: 全モード

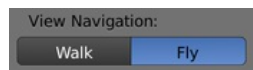
ホットキー: ⇧ ShiftF

解説

ビューで見たい場所を決めるとき、通常は上記の方法をとります。

しかし特に巨大なモデル、自然環境や建築物のようなものでは、単にモデルのあちこちを見て回りたくなることがあります。透視投影(perspective)モードではモデルを見るにも限界があります。例えばズーム後の大きなパンは、非常にやっかいな難しいものになったり、カメラを近づけることができないように見えたりします。試しに、ビューで非常に遠くにあるオブジェクトに、従来の方法(上述)でカメラを寄せてみてください。

[Walk mode\(歩行モード\)](#) や [Fly mode\(飛行モード\)](#) を使えば、こうした制限なしに、カメラをあちこちに移動、パン、ティルト、ドリーします。



[User Preferences\(ユーザー設定\)ウィンドウ](#) で、ビューナビゲーション処理を起動した際の、デフォルトのナビゲーションモードを選んでください。もしくは、View(ビュー) » Navigation(視点の操作)からそれぞれのモードを呼び出せます。

View Shading

(ビューの陰影処理)

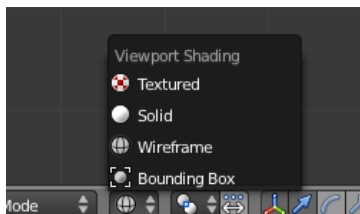
モード: 全モード

ホットキー: Z / ⇧ ShiftZ / AltZ / ⇧ ShiftAltZ / D

Description

(解説)

コンピュータの速度、シーンの複雑さ、作業内容に応じて、複数の描画モードを切り替えることができます (訳注: モード種類はバージョンによって多少変わります):



3Dビューの描画モードボタン

レンダーード(Rendered)

レンダリングプレビューを表示します。

テクスチャ(Texture)

OpenGL ライティングを使って、UV画像テクスチャを貼ったモデルを表示します。プロシージャル(procedural)や UVマップされていないテクスチャは表示されません。

Shaded

(訳注: Blender 2.6 にはありません) 各頂点のすべてのテクスチャとライティングを概算し、次々に混ぜ合わせます。レンダーエンジンを使ってテクスチャを確認するよりかなり正確性に欠けますが、高速です。なお、シーンに照明がないと何もかも黒いまです。

ソリッド(Solid)

面を組み込みの OpenGL ライティングでベタ塗りする、デフォルトの描画モードです。この描画モードはシーンの光源に左右されません。設定はユーザー設定 ウィンドウの システム タブにある ソリッド OpenGL ライト(Solid OpenGL lights) で変更できます。

[システム設定についてもっと詳しく»](#)

ワイヤーフレーム(Wireframe)

オブジェクトが線だけで描かれ、形状がわかりやすくなります (つまりメッシュやサーフェスの辺です)。

バウンディングボックス(Bounding Box)

オブジェクトはまったく描かれませんが、代わりに、オブジェクトのサイズや形状に応じた直方体だけを表示します。

描画モードの切り替えには次のいずれかを使います:

- 3D ビューのヘッダーにある 描画モード のドロップダウンリスト (図「3Dビューの描画モードボタン」)。
- D を押して 描画モード メニューをポップアップさせます。
- Z-ベースのショートカット。下記参照:

描画モードとZベースのショートカット。

- Z 描画モード ワイヤーフレーム(Wireframe) とソリッド(Solid) を切り替えます。
- ⇧ ShiftZ 描画モード ワイヤーフレーム(Wireframe) と Shaded を切り替えます。
- AltZ 描画モード ソリッド(Solid) と テクスチャ(Texture) を切り替えます。
- ⇧ ShiftAltZ テクスチャ(textured) 描画モードに切り替えます。

View Properties Panel

(ビューのプロパティパネル)

モード: 全モード

パネル: Properties (プロパティ)

メニュー: View (ビュー) » View Properties (プロパティ) ...

Description

(解説)

上述のヘッダーにあるコントロールに加えて、View (ビュー) のプロパティ パネルで 3D ビューに関するその他の設定を行えます。View (ビュー) » Properties (プロパティ) ... メニューから表示できます。

View

(ビュー)

レンズ(Lens)

3Dビューのカメラの焦点距離を、[レンダリングカメラ](#) の設定とは別の、ミリメートル単位で調整します。

オブジェクトに注視(Lock to Object)

オブジェクト欄にオブジェクト名を入力すると、そのオブジェクトにビューを固定します。つまり、オブジェクトが常にビューの中心になります(唯一の例外はアクティブカメラビュー 0 NumPad です)。注視するオブジェクトがアーマチャなら、さらに ボーン(Bone) 欄に名前を入れて、そのボーンを注視できます。

カーソルを注視(Lock to Cursor)

ビューの中心を 3Dカーソルの位置に固定します。

カメラをビューにロック(Lock Camera to View)

カメラビューにいるとき、カメラビューにとどまったまま 3D 空間にあるカメラを動かすのに使ってください。

クリップ開始(Clip Start) とクリップ終了(Clip End)

ビューポートに表示されるものの最小/最大距離を調整します。

註記



クリップ範囲を広くすると遠近両方のオブジェクトを見ることができますが、深度の精度が下がります。

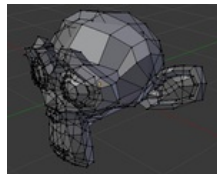
これを回避するには...

- 大きな画面で作業するときはクリップ開始値を増やします。
- オブジェクトを遠くから見ないときはクリップ終了値を減らします。

透視投影が無効の場合はクリップ終了値だけが使われます。非常に高い値にすると描画の乱れ(artifacts)が起こる可能性があります。

これは blender 特有の問題ではなく、OpenGL/DirectX グラフィックスを使うアプリケーションはすべて、同じ制限を持ちます。

例:



クリップによる画像乱れのないモデル。 クリップによる画像乱れのあるモデル。 編集モードで画像乱れのあるメッシュ。

ローカルカメラ(Local Camera)

このビューで使われているアクティブなカメラ

3D カーソルの位置(3D Cursor Location)

3D カーソルの位置を精密に定義できます。

Item

(アイテム)

このセクションには、現在選択しているオブジェクトを表示します。

Display

(表示)

レンダリングのみ(Only Render)

レンダリングされるものだけを表示します。

選択時にアウトライン表示(Outline Selected)

無効にすると、ソリッド(Solid)/Shaded/テクスチャ(Textured) 描画タイプで選択オブジェクトの周囲に描かれるピンクのアウトラインが表示されなくなります。

全てのオブジェクトの原点(All Object Origins)

有効にすると、オブジェクトの中心を示す点が、オブジェクトの非選択時を含めて常に表示されます(デフォルトでは、ソリッド/shaded/テクスチャシェイディングでは、非選択のもの中心はジオメトリに隠されるかもしれません...)

関係の線(Relationship Lines)

親子関係(parenting)、拘束(constraining)、フック(hooking)を示す点線の表示/非表示を切り替えます。

All Edges

オブジェクト コンテキストでワイヤーオーバーレイ(wire overlay)が有効のとき、すべてのワイヤーフレームをビューポートに表示します。

グリッドフロア(Grid Floor)

無効にすると、平行投影の上面図/正面図/側面図以外のビューでグリッドを表示しません。

X軸, Y軸, Z軸

平行投影の上面図/正面図/側面図以外のビューで、どの軸を表示するか決めます。

ライン(Lines)

平行投影の上面図/正面図/側面図以外のビューで、グリッドを構成する線の数を調整します(両方向)。

拡大縮小(Scale)

グリッドフロアのサイズを調整します。

細分化(Subdivisions)

ズームインしたときに各マス目の内側に現れる線の数を調整します。したがって平行投影の上面図/正面図/側面図専用の設定です。

シェーディング(Shading)

3Dビューにあるオブジェクトのシェーディング方式を調整します。

テクスチャソリッド(Textured Solid)

テクスチャを割り当てた面をソリッドビューで表示します。

四分分割表示(Toggle Quad View)

4つ描画域のある3Dビューの切り替えです。[フレームの配置についてもっと詳しく](#)》

Background Image

(下絵)

モード: 全モード

パネル: Background Image(下絵)

メニュー: View(ビュー) » Properties(プロパティ)...

3Dビューに下絵を置くと非常に役立つ状況が多くあります。モデリングは当然のこと、ペインティングや(モデルに直接テクスチャを描くときに、各面の参照画像を設定できます)アニメーション(動画を下絵にするとき)などです。

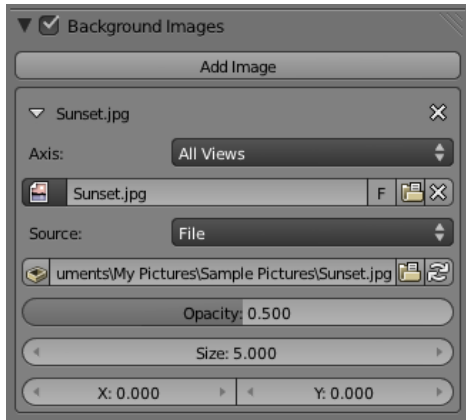


下絵には特筆すべき点がいくつかあります。:

- ウィンドウに対して指定します(つまり3Dビューごとに別の下絵を指定できます。例: 上面図/正面図/側面図の下絵を対応するビューに置く、など)
- **平行投影の上面図、側面図、正面図(およびそれぞれの反対側のビュー)にしか使えません!** これら6つのビューを切り替えても、画像は同じままです。
- 画像サイズはウィンドウのズーム率に結びついています(例えばズームインすると大きくなります)。
- 動画やアニメーションシーケンスを使えます。

Settings

(設定)



下絵パネル

この機能はビューのプロパティパネル(N)にある下絵(Background Image)メニューから利用できます。最上段にあるチェックボックスで下絵機能をオン/オフ切り替えます。デフォルトでは1枚の画像用のスペースしかありません。白い三角形をLMB すると設定の一覧が現れます。

有効化すると、下絵に使う画像を既存のデータブロックから選ぶか、新たに画像を読み込むことができます。座標軸(Axis)メニューは下絵を表示するビューを決めます。画像を追加(Add Image)ボタンをLMB して、さらに画像を追加することができます。画像を読み込むと、次の設定が利用できるようになります。

ソース(Source)

使うファイルの種類を選びます。選んだ種類によって、オプションがその下に現れます:

ファイル

画像ファイルを使います

ソースファイル(Source File)

現在のデータブロックにリンクされた実際のファイルを表示します。サポートされる形式には bmp、gif、jpg、png、tga、tif があります。

シーケンス(Sequence)

番号が振られた一連の画像ファイル

フレーム(Frames)

シーケンス内で使う画像ファイルの数

開始 (Start)

開始フレーム番号

オフセット(Offset)

シーケンス内で使うフレーム番号のオフセット

Fields

描画フレームごとのフィールド(fields)数

自動更新 (Auto Refresh)

フレームの変更時、画像を自動更新します

繰り返し (Cyclic)

シーケンス内の画像を繰り返します

動画 (Movie)

動画ファイルを使います:

動画の長さ matches (Match Movie Length)

フレーム数を動画に合わせます

生成 (Generated)

Blender 内で生成した画像を使います:

幅 (Width), 高さ (Height)

画像の幅と高さをピクセル単位で指定します。

ブランク (Blank)

空の画像を生成します。

UVグリッド (UV Grid)

UVマッピングのテスト用のグリッドを作ります。

カラーグリッド (Color Grid)

UVマッピングのテスト用のカラーグリッドを作ります。

不透明度 (Opacity)

下絵の不透明度を調整するスライダーです (**0.0** (完全な不透明) から **1.0** (完全な透明) の範囲内)。

サイズ (Size)

3D ビューでの画像のサイズを調整します (単位は Blender Unit です)

X オフセット, Y オフセット

ビュー内の下絵の横方向/縦方向のオフセット (デフォルトは原点上に置かれます。単位は Blender Unit です)。



低解像度の代用品を使いましょう

PC の処理効率向上のため、下絵を使うときは低解像度の代用品を使う必要があるかもしれません。モニタが解像度が 800×600 なら、フルスクリーン、ズームなしの下絵は 800×600 あれば十分です。もし下絵が 2048×2048 なら、コンピュータはピクセルの放棄に精を出します。2k×2k 画像を使うのではなく、(Blender や Gimp を使って) 例えば 512×512 などに縮小してみてください。画質や正確さを失うことなしに、処理効率は16倍良くなるでしょう。そして、モデルが洗練されるに従って、解像度を増やすことができます。

Shortcuts

(ショートカット)

Camera View

(カメラビュー)

モード: 全モード

ホットキー: 0 NumPad

メニュー: View(ビュー) » Camera(カメラ) » Active Camera(アクティブカメラ)

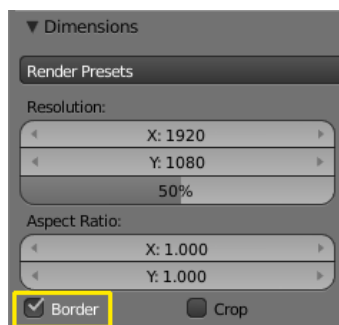
カメラビューで仮想的な写真を構成して、シーンがレンダリング時にどのように見えるのかプレビューすることができます。0 NumPad を押すと現在アクティブなカメラから見えるシーンを表示します。このビューでは レンダリングボーダー (Render Border) を設定して、レンダリングされるカメラビューの範囲を決めることができます。



カメラビューでは最終的なレンダリング画像のプレビューができます。左: 透視投影ビュー、中央: カメラビュー、右: レンダリング画像

Render Border

(レンダーボーダー)



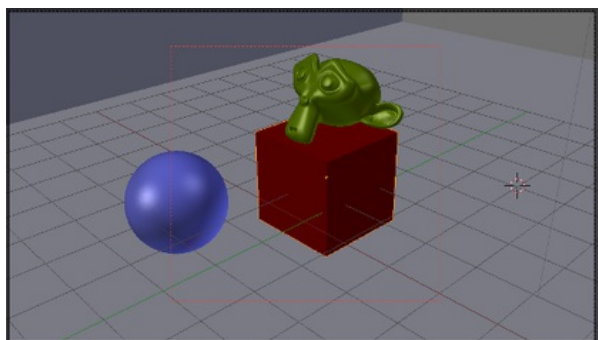
レンダー境界の切り替え

カメラビューでは、⇧ ShiftB を押して レンダーボーダー (Render Border) を定義できます。カメラビューの中に、オレンジの点線で四角形を描けるようになります。レンダリングは、シーンの、レンダーボーダー内に見えている部分に限定されます。これはレンダリング時間を抑えるのに非常に役立ちます。レンダーボーダーは、レンダー コンテキストで 寸法 (Dimensions) パネルにある ボーダー (Border) オプションを使うか、⇧ ShiftB でカメラビューより大きな レンダーボーダー を定義して、無効にできます。

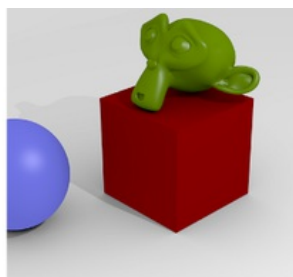
アンチエイリアスとブラーオプションとボーダー

レンダーボーダーが有効なときは、Full Sampling Anti-Aliasing が無効になり、Sampled Motion Blur が有効になります。

[アンチエイリアスについてもっと詳しく](#) » [モーションブラーについてもっと詳しく](#) »



レンダーボーダーとそのレンダリング結果



[レンダリングオプションについてもっと詳しく](#) »

[カメラについてもっと詳しく](#) »

Layers

(レイヤー)

モード: オブジェクトモード

パネル: 関係 (オブジェクト コンテキスト)

ホットキー: M

メニュー: Object(オブジェクト) » Move to Layer(レイヤー移動)...

たいてい、3Dシーンが複雑になるほど、わかりづらさは飛躍的に増えます。アーティストが照明効果をオブジェクトごとに細かく調整しなければならない場合や、あるオブジェクト用の照明が近くのオブジェクトに影響しないようにしたい場合もあります。このために、もしくは下記の他の目的のために、オブジェクトを1つ以上の「レイヤー」に置くことができます。オブジェクトレイヤーを使うと、次のようなことができます:

- 3Dビューヘッダーでレイヤーを選んで、3Dビューに特定レイヤーのオブジェクトだけを選択して表示します。再表示が速くなり、仮想世界が片付いて作業しやすくなります。
- 所属するレイヤー(複数の場合もあります)のオブジェクトだけを照らす照明を作って、[オブジェクトを照らすライトの選択 \(eng\)](#)ができます。
- どのフォース(forces)がどの[パーティクル](#)に働くのかを決めます。パーティクルは同じレイヤーにあるフォースにだけ影響を受け、影響を与えるからです。
- [レンダリングレイヤー \(eng\)](#) を使って、レンダリングするレイヤー(すなわちレンダリングするオブジェクト)と、コンポジティングで利用できるプロパティチャンネルを決めます。

アーマチュアも、多様なボーン、コントローラー、ソルバー、カスタムシェイプなどによって非常に複雑になります。アーマチュアの配置はふつう密集するので、散らかりやすいものです。このため Blender はアーマチュア専用のレイヤーを用意しています。アーマチュアレイヤーはオブジェクトのレイヤーと非常に似ており、アーマチュア(リグ)をレイヤーをまたぐように分けたり、作業したいレイヤーだけを表示したりできます。

[アーマチュアレイヤーについてもっと詳しく](#) »

Working with Layers

(レイヤーでの作業)



2D のグラフィックスアプリケーションのレイヤーをご存知かもしれません。3D のレイヤーはそれとは違って描画順に影響せず、シーンの整理が主目的です。

レンダリングするとき、Blender は選択されたレイヤーだけを描画します。*非選択*のレイヤーにしか照明がなければ、環境光で照らされたオブジェクトを除けば、レンダリングした画像には何も表示されません。

[グループと親子関係](#) は、関連のあるオブジェクトの集合を論理的にグループ化するもう一つの方法です。さらなる情報については、関連する章をご覧ください。

Viewing layers

(レイヤーの表示)

Blender は 20 のレイヤーを備えています。ヘッダーにある、ラベルのない小さなボタンで表示非表示が切り替えられます(右図「3D ビューポートのレイヤーボタン」)。レイヤーをひとつ選ぶには、対応するボタンを LMB  クリックします。複数を選ぶには ◊ Shift LMB  を使います。すでに選ばれているレイヤーにこの操作を行うと選択を解除します。



3D ビューポートのレイヤーボタン

キーボードでレイヤーを選ぶには、1 から 0 (テンキーではないほう) でレイヤー 1 から 10 まで(上段のボタン)を、Alt1 から Alt0 でレイヤー 11 から 20 まで(下段)を選びます。複数選択用の ◊ Shift キーはこのショートカットキーに対しても働きます。

Locking to the scene

(シーンへのロック)

デフォルトで、レイヤーボタンのすぐ右にあるロックボタンは有効です。これは、ビューに表示されているレイヤーへの変更が、シーンにロックされている他のすべての 3D ビューに反映されるということです。さらなる情報は [3Dビューのナビゲートオプションのページ](#) をご覧ください。

Multiple Layers

(複数レイヤー)

オブジェクトは複数レイヤーに存在できます。例えば、同じレイヤーのオブジェクトだけを照らす照明を、レイヤー 1,2,3 に置きます。あるオブジェクトがレイヤー 3 と 4 に属するとき、このオブジェクトには照明が当たりますが、4 と 5 に属する別のオブジェクトには当たりません。特に照明やパーティクルには、レイヤー特有の効果が働く場所がたくさんあります。

Moving objects between layers

(レイヤー間の移動)



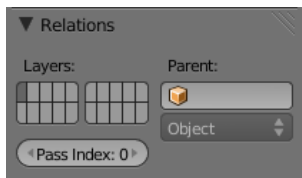
レイヤー選択

選択オブジェクトを別のレイヤーに移動するには、Mを押してポップアップダイアログから希望のレイヤーを選びます。オブジェクトは同時に複数のレイヤーに属することもできます。複数レイヤーに置くには **⇧** Shift クリックします。



オブジェクトコンテキスト選択

選択オブジェクトのレイヤーを見たり変えたりするために、オブジェクトコンテキストの関係 (Relations) パネルも使えます。



オブジェクトコンテキストの関係パネルにあるレイヤー

関係パネルにレイヤーボタンがあるのが見えるはずですが。前述のように **⇧** Shift LMB  クリックを使うとオブジェクトを複数のレイヤーに表示できます。

Example of object layer arrangement

(オブジェクトのレイヤー配置の例)

おすすめは、シーンの重要な部分をレイヤー上段に、使用しないものやそれほど頻繁に変更しないものをレイヤー下段に置くことです(または上段の代替として使えます)。例えば、2人の役者を含むステージセットでは、次のようなレイヤー分けができるかもしれません:

1. 主役の俳優
2. 脇役
3. サポートスタッフ(エキストラ)
4. パーティクルと効果(渦巻き、風)
5. メインの舞台
6. メインの背景幕とパネル
7. メインの小道具(机、椅子)
8. 細かな小道具、fillers、装飾、アクセサリ
9. カメラ、照明
10. 主役のアーマチュア
11. 脇役のアーマチュア
12. スタッフのアーマチュア
13. 替えの衣装
14. メッシュ用の仕掛け
15. 構成や寸法の違う舞台
16. 使ったかもしれない背景幕
17. シーンをふさぐ可能性のあるその他の大道具
18. 小道具用の仕掛け
19. 追加の照明

ローカルビューとグローバルビュー (Local or Global View)

モード: 全モード

ホットキー: Numpad/

メニュー: ビュー (View) » ローカルビュー (Local View) または ビュー (View) » グローバルビュー (Global View)

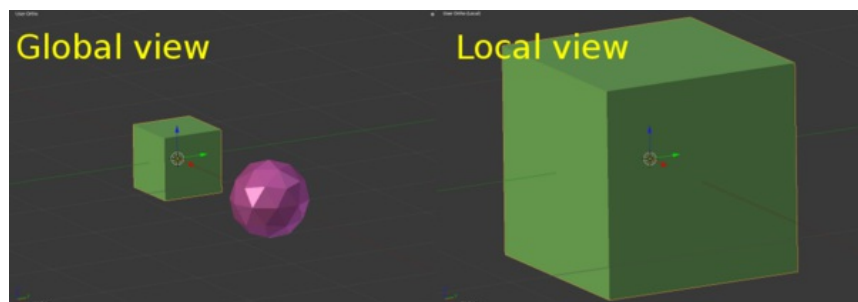
解説

ローカルビューでは、選択されているオブジェクトだけが表示され、入り組んだシーンで編集をしやすくなります。ローカルビューに入るには、初めにオブジェクトを選んでから、ビュー » ローカルビュー メニューを選びます。グローバルビューに戻るには ビュー » グローバルビュー メニューを使ってください。Numpad/ は2つのビューを切り替えます。

ローカルビューでは 3Dビュー ヘッダーのレイヤーおよびロックボタンは見えなくなることに注意してください。

例

グローバルビューでは、すべてのオブジェクトが見えます。緑の立方体を選択して Numpad/ でローカルビューに切り替えると3Dビューの中央に立方体が配置されます。シーンに何千ものオブジェクトがある場合は、この機能を使えば選択オブジェクトだけを表示するので、対話的な操作をスピードアップできるかもしれません。



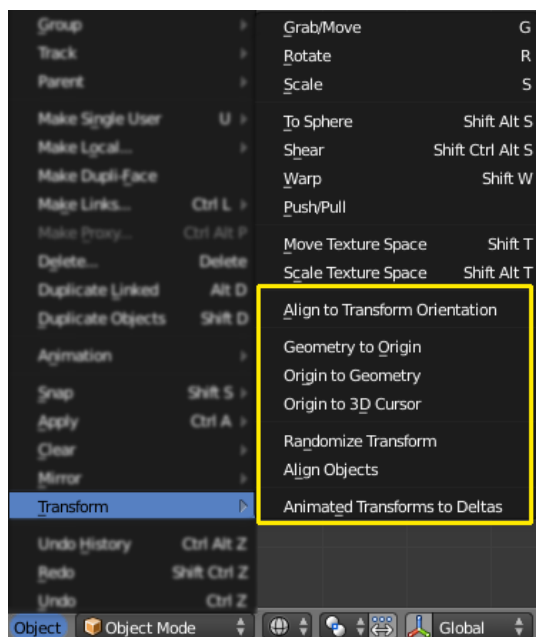
Global and Local view

Transformations

(トランスフォーム)

モード: オブジェクトおよび編集モード

メニュー: Object(オブジェクト)/Mesh(メッシュ) » Transform(トランスフォーム)



オブジェクトモードのトランスフォームメニュー。黄枠の部分はオブジェクトモードでのみ使えます

トランスフォームには、選択したオブジェクトやメッシュに実行し位置や特性を変えるような多くの操作が含まれます。基本的なトランスフォームには選択要素の移動、回転、拡大縮小があります。高度なトランスフォームにはミラー、球体化、せん断、引く/押す、湾曲があります。以下のリンクには、各操作のさらに詳しい説明があります。

Basic transformations

(基本トランスフォーム)

- [移動\(Grab/Move\)](#): 選択要素を移動します。
- [回転\(Rotate\)](#): 選択要素を回転します。
- [拡大縮小\(Scale\)](#): 選択要素のサイズを変えます。

Advanced Transformations

(高度なトランスフォーム)

- [ミラー\(Mirror\)](#): 選択部分をミラー反転します。
- [球へ変形\(To Sphere\)](#): 選択部分を球の形に近づけます。
- [せん断\(Shear\)](#): 選択要素をせん断します。せん断は、平行な選択要素を互い違いに移動させます。
- [湾曲\(Warp\)](#): 選択部分を湾曲させます。
- [法線に沿って伸縮/膨張\(Shrink/Fatten\)](#): 各頂点をその法線に沿って動かします(メッシュの編集モードのみ)。
- [押す/引く\(Push/Pull\)](#): 選択要素を押すか、引きます(誰かが選択要素の端を持って、広げたり押し縮めているところを想像してみてください)
- [テクスチャ空間の移動\(Move Texture Space\)](#) [\(eng\)](#): テクスチャ空間がテクスチャの配置を決めます。これが動かされると [テクスチャマッピング](#) で役立つかもしれません。
- [テクスチャ空間の拡大縮小\(Scale Texture Space\)](#) [\(eng\)](#): 上と同じです。[テクスチャマッピング](#) で役立ちます。
- [座標軸に沿う\(Align to Transform Orientation\)](#): 現在の座標軸にオブジェクトを沿わせます。
- [ジオメトリ → 原点\(Geometry to Origin\)](#): オブジェクトのジオメトリを原点に移動します。
- [原点 → ジオメトリ\(Origin to Geometry\)](#): オブジェクトの原点をジオメトリ位置に移動します。
- [原点 → 3Dカーソル\(Origin to 3D cursor\)](#): オブジェクトの原点を 3Dカーソル位置に移動します。
- [ランダムな変形\(Randomize Transform\)](#): 選択要素にランダムな移動、回転、拡大縮小を行います。
- [オブジェクトの整列](#) [\(eng\)](#): 特定の軸にオブジェクトを揃えます。
- [アニメーションしたトランスフォームをデルタへ](#) [\(eng\)](#): トランスフォーム値のアニメーションをデルタトランスフォーム値に変換します。キーフレームを持つオブジェクトを複製して、移動、回転、拡大縮小のようなオフセットをもたせることができます

Transform Control

(トランスフォームコントロール)

上の各ページに書かれている専用操作に加えて、通常のコントロールにもトランスフォームの効果を変えるのに使えるものがたくさんあります。例えば精密操作のためのキーボード入力やトランスフォームのリセット、軸ロックなどがあります。

[トランスフォーム操作についてもっと詳しく](#) »

Grab/Move

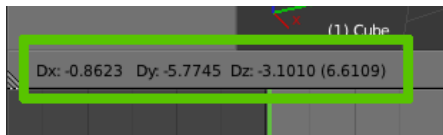
(移動)

モード: 3Dビューでは オブジェクトモード、編集モード、ポーズモード;
UV/画像エディタツール、シーケンスエディタ、ドープシート、グラフエディタ では他の特別な移動操作

ホットキー: Gまたは軸制約との組み合わせ

メニュー: コンテキスト依存。現在のオブジェクトのメニュー → Transform(トランスフォーム) → Grab/Move(移動)

このオプションは、アクティブな 3Dビューポートの 3D 空間にあるオブジェクト(オブジェクトモード時)やオブジェクトの構築に使われる要素の移動を行います。ノードエディタ、グラフエディタ、UV エディタ、シーケンサーなど、他の様々なエディタ環境でも似た機能が使えます。オプションの詳細については、それぞれの関連節で扱います。



移動の表示

移動中は、座標軸ごとの変化量が 3Dビューウィンドウの左下隅に表示されます。

3D View

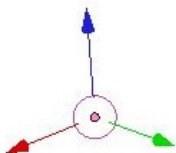
(3D ビュー)

3Dビューでは 2 種類の移動オプションがあります:


- ショートカットやその組み合わせを使う方法。
- マニピュレータを使う方法。3Dビューのヘッダーから選んだ場合。

Transform Widget

(マニピュレータ)



マニピュレータ

デフォルトでは マニピュレータが表示されています。3D ビューにあるマニピュレータ自身を LMB  でドラッグ移動すると移動操作を行えます。

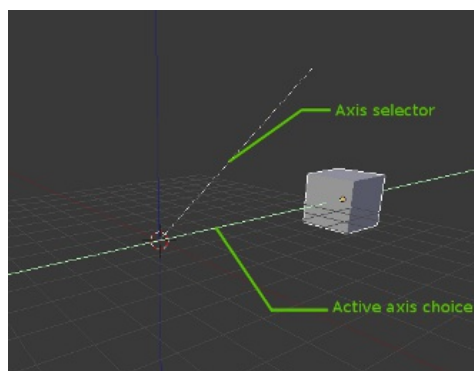
Shortcuts in the 3D View

(3Dビューでのショートカット)

3D空間でものを動かす最速の方法の一つは G です。このホットキーを押すと「移動」トランスフォームモードに入り、選択オブジェクトやデータをマウスポインタに追従させて自由に動かせます。このショートカットと特定の軸を表すショートカットを組み合わせると、トランスフォームを完全に制御できます。



LMB 

移動を確定し、オブジェクトやデータを画面上の現在の位置に置きます。



アクション時の軸制約

MMB

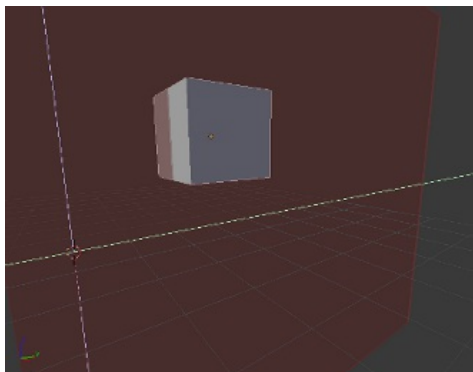
3Dビューのマウスポインタの位置によって、X、Y、Z軸方面への移動を自動的に制限します。G キーを押した後に MMB  が押されると、3Dビュー空間の3軸を含む、移動制限のオプションが表示されます。制限軸は MMB  が離された場所に関する軸になります。操作中はいつでも、キーボードの X、Y、Z を押して軸を変更できます。

RMB または Esc

移動をキャンセルし、オブジェクトやデータを元の位置に戻します。

Alt + g

これまでにそのオブジェクトに行われたすべてのトランスフォームを消去します。オブジェクトモードでのみ動作します。



Shift+X が作動中

◇ Shift と XYZ

補完的な軸によるトランスフォーム制約です。このオプションを使うと、選んだ軸以外の軸でのみ移動を行えるようになります。右図をご覧ください。

Controlling Grab/Move Precision

(移動の精密な制御)

上述の軸の制限オプションに加えて、Blender にはトランスフォームの移動量を小さくしたり、移動の刻み幅を決めたりする機能があります。

◇ Shift

トランスフォーム動作を遅くします。G を押した後、移動モードに入った状態で ◇ Shift キーを押すと移動量が抑えられ、細かな位置を決められます。

Ctrl

オブジェクトの移動中、事前に決めておいたスナップに基いて [スナップ](#) します。スナップ機能そのものが無効であっても動作します。

Ctrl+◇ Shift

直感的にわかるように、Ctrl と ◇ Shift 操作の組み合わせです。オブジェクトをスナップさせながら精密に移動させられます。

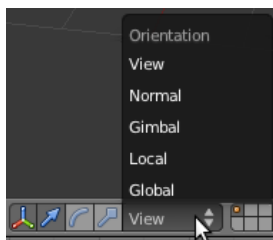
X/y/z + <数値>

この操作は指定軸にトランスフォームを制限すると同時に、その軸に沿う移動量を数値で指定します。入力された数値は 3Dビューウィンドウの左下隅に表示されます。入力中にバックスペースを押すと数値指定オプションを取り消しますが、オブジェクトの軸制限は同じままで、数字を入力し直すことができます。x/y/z キーを押すといつでも、制限軸を変えることができます。特定位置に移動させたり、オブジェクト間の距離を増やしたりするのに使えます。

Orientations

(座標系)

すべてのトランスフォームで5種類の座標系を使えます。



座標系の選択メニュー

- グローバル(Global、デフォルト)
- ローカル(Local)

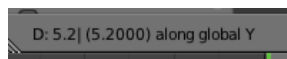
- 法線 (Normal)
- ジンバル (Gimbal)
- ビュー (View)

[座標系についてもっと詳しく >>](#)

座標系に従ってトランスフォームは行われます。座標系はマニピュレータの選択ウィジェットの横にある、ポップアップメニューから選べます。

G キーに続けて Xx/Yy/Zzのいずれかを入力すると、そのままそのローカル軸に沿った移動が可能になります。もちろん、さらに数字を続けて移動量を指定することもできます。

上の操作に似ていますが、G キーに続けて \diamond Shiftと Xx/Yy/Zzのいずれかを入力すると、そのままローカル指定軸以外の2軸による平面に移動が制限されます。



数値入力の表示

Other Editor Windows

(その他のエディタウィンドウ)

UV/画像エディタツール、シーケンスエディタ、ドープシート、グラフエディタといった他のエディタウィンドウでは、移動操作は文脈に応じたオブジェクトや要素を動かすのに使われます。ただし3Dビューとは違って(このページが**3D インタラクション**セクションにあるにも関わらず)通常は **X** と **Y** の2つの軸しか表示されませんが、これらのオブジェクトや要素は 3D インターフェースに現れます。Blender は単に、3番目になり得る軸の動きを拘束するだけです。3Dビューで使ったショートカットの大半は、これらのエディタウィンドウでも使えます。回転や拡大縮小など他のすべてのトランスフォームでも同じです。

Python Scripting

(Python スクリプトの記述)

Blender では Python スクリプトを記述してオブジェクトや要素を特定位置に移動することもできます。これには Python の対話コンソールを使うか、テキストエディタウィンドウで Python スクリプトを実行します。

現在のオブジェクトの位置ベクトル

```
bpy.context.scene.objects.active.location
```

シーン内のアクティブオブジェクトの位置ベクトルを返します。位置ベクトルに別の値を割り当ててオブジェクトの位置を変えることができます。

アクティブオブジェクトの移動オペレータと構文

```
bpy.ops.transform.translate(value=(<DX>, <DY>, <DZ>), constraint_axis=(<bool>, <bool>, <bool>),
constraint_orientation='<ORIENTATION NAME>', mirror=<bool>, proportional='<ENABLE?DISABLE>',
proportional_edit_falloff='<FALLOFF TYPE>', proportional_size=<INT>, snap=<bool>, snap_target='<SNAP TARGET>',
snap_point=<x,y,z>, snap_align=<bool>, snap_normal=<x,y,z>, texture_space=<bool>, release_confirm=<bool>)
```

Hints

(ヒント)

- オブジェクトモードでの移動は、編集モードですべての頂点/辺/面を選択してオブジェクトを移動するのとはまったく違います。こうしてしまうとオブジェクトに用いるトランスフォームの中心が混乱するでしょう。
- グローバル座標系以外でショートカット G → x/y/z を使うと、その座標系の X軸ではなく、グローバル座標系の X軸だけに移動が制限されます。

Rotate

(回転)

モード: オブジェクトモードおよび編集モード

ホットキー: R

メニュー: Object (オブジェクト)/Mesh (メッシュ)/Curve (曲線)/Surface (曲面) » Transform (トランスフォーム) » 回転

解説

回転 (rotation) はまたの名をスピン (spin)、ねじり (twist)、公転 (orbit)、ピボットターン (pivot)、旋回 (revolve)、転がり (roll) とも言い、1つ以上の軸や要素のピボットを中心に、要素 (頂点、辺、面、オブジェクトなど) の方向を変える動作を含みます。要素を回転させるには次のような複数の方法があります:

1. キーボードショートカット (R)
2. マニピュレータ
3. プロパティメニュー (N)

基本的な回転の使い方と一般的なオプションを以下に述べます。さらなる情報、例えば精密制御 (Precision)、軸ロック (Axis Locking)、数値入力 (Numeric Input)、スナップ (Snapping) と多様なピボット (Pivot Point) などのオプションの詳細情報は、トランスフォームコントロールと座標系のページにあります。

[トランスフォームコントロールについてもっと詳しく »](#)



[座標系についてもっと詳しく »](#)

Usage

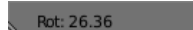
(使い方)

Rotation using the keyboard shortcut

(キーボードショートカットを使った回転)

1. RMB  で回転したい要素を選択します。
2. R を1度だけ押して回転モードに入ります。
3. マウスを動かして要素を回転します。マウスが要素の中心に近づくほど回転効果が大きく現れます。
4. LMB  クリックで変更を確定します。

回転量は 3Dビューの左下隅に表示されます。



回転値

Constraining the rotation axis (axis locking)

(回転の軸を制限する (軸ロック))

[軸ロック](#) を使って、回転を特定軸に制限することができます。回転を制限するには、次のショートカットを使います:

- R, X: **X軸** に対してのみ回転
- R, Y: **Y軸** に対してのみ回転
- R, Z: **Z軸** に対してのみ回転

軸ロックは回転モードに入ってから MMB  を押しでも有効にできます。マウスを希望する方向に動かしてください。例:

- R, マウスをX方向に動かして MMB  : **X軸** に対してのみ回転します

[軸ロックについてもっと詳しく »](#)

Fine Tuning The Rotation

(回転の精密な調整)

回転の [精密制御](#) は \diamond Shift と Ctrl キーを使って行い、刻み幅をつけた回転ができます。回転モードに入れば、回転角度を [数値](#) で入力することもできます。

- 回転中に Ctrl を押し続けると、選択要素の回転の刻み幅が5度ずつになります。
- 回転中に \diamond Shift を押し続けると、選択要素の回転が0.01度ずつになります。
- 回転中に \diamond ShiftCtrl を押し続けると選択要素の回転が1度ずつになります。
- Rを押して数値を入力し、 \leftarrow Enter で確定します。
- トラックボール回転は R, R で有効になります。



座標系依存の回転

デフォルトでは、すべての回転はグローバル座標系で行われます。回転の軸は「軸キー」を2度押しすと変更できます。例えば、R、X、Xと押せば

デフォルトではローカル座標系で回転が行われるようになります。

[精密制御についてもっと詳しく](#)»

[数値を使ったトランスフォームについてもっと詳しく](#)»

[座標系についてもっと詳しく](#)»

Rotation with the 3D Transform Manipulator

(マニピュレータを使った回転)



回転マニピュレータ

3Dビューのヘッダーで、マニピュレータが有効である(赤、緑、青の3つ組が選択されている)ことを確認してください。マニピュレータの種類を回転にしてください(下図でハイライトされている円弧のアイコン)

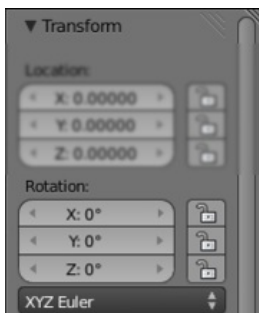


1. RMB で要素を選択します。
2. マニピュレータの色のついた軸のいずれかを LMB を使ってドラッグし、オブジェクトをその軸に沿って回転させます。マニピュレータといっしょに Shift、 Ctrl を使ったり、数値入力をしたりしてさらに細かく操作することもできます。
3. LMB を離すか、Space か Enter を押すと変更が確定します。また、RMB か Esc で変更をキャンセルできます。

[マニピュレータについてもっと詳しく](#)»

Rotation with the Properties Panel

(プロパティパネルを使った回転)



回転トランスフォームのプロパティパネル

回転値は、プロパティパネル (N) でトランスフォームの角度スライダーの値を変更して指定することもできます。特定軸に沿った回転は、南京錠のアイコンで有効化/無効化できます。回転モード(オイラー角、軸角度、クォータニオン)も、このパネルのドロップダウンリストから選べます。

[パネルについてもっと詳しく](#)»

[回転のモードについてもっと詳しく](#)»

[回転モードに関する詳細\(英文\)](#)»

拡大縮小

モード: オブジェクトモードおよび編集モード

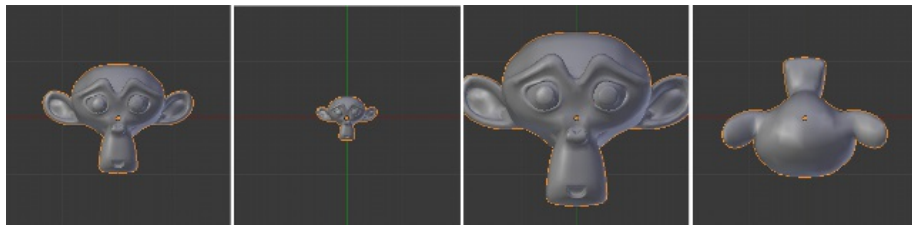
ホットキー: S

メニュー: オブジェクト/メッシュ/曲線/サーフェス » トランスフォーム » 拡大縮小

解説

Sを押すと拡大縮小トランスフォームモードに入ります。このモードではマウスポインタの位置に応じて選択要素が内側に縮小、もしくは外側に拡大されます。マウスポインタがピボットより離れるほど要素のサイズは大きくなり、近づくほど小さくなります。マウスポインタがピボットの元の側から反対側に入っても拡大縮小は負の方向に続けられ、要素を反転させます。

[ピボットについてもっと詳しく»](#)



基本的な拡大縮小の使い方です。左から、元のオブジェクト、縮小したオブジェクト、拡大したオブジェクト、反転したオブジェクト

要素の拡大縮小には次のような方法があります:

1. キーボードショートカット (S)
2. マニピュレータ
3. プロパティメニュー(N)

基本的な拡大縮小の使い方と一般的なオプションを以下に述べます。さらなる情報、例えば精密制御 (Precision)、軸ロック (Axis Locking)、数値入力 (Numeric Input)、スナップ (Snapping) と多様なピボット (Pivot Point) などのオプションの詳細情報は、トランスフォームコントロールと座標系のページにあります。

[トランスフォームコントロールについてもっと詳しく»](#)

[座標系についてもっと詳しく»](#)

使い方

キーボードショートカットを使った拡大縮小

1. RMB で拡大したい要素を選択します。
2. S を1度押して拡大縮小モードに入ります。
3. マウスを動かして要素を拡大縮小します。
4. LMB クリックして変更を確定します。

拡大縮小の量は 3Dビューの左下隅に表示されます。

Scale X: 1.0701 Y: 1.0701 Z: 1.0701

拡大縮小の値

拡大縮小の軸を制限する (軸ロック)

軸ロックを使って、拡大縮小を特定軸に制限することができます。拡大縮小を制限するには、次のショートカットを使います:

- S, X: **X軸** に対してのみ拡大縮小
- S, Y: **Y軸** に対してのみ拡大縮小
- S, Z: **Z軸** に対してのみ拡大縮小

軸ロックは拡大縮小モードに入ってから MMB を押しても有効にできます。マウスを希望する方向に動かしてください。例:

- S, マウスをX方向に動かして MMB : **X軸** に対してのみ拡大縮小します

[軸ロックについてもっと詳しく»](#)

拡大縮小の精密な調整

拡大縮小の**精密制御**が可能です。⇧ Shift と Ctrl キーで拡大縮小を不連続な量に制限できます。拡大縮小モードに入れば、拡大量を Blender 単位 (BU) の **数値** で入力することもできます。

- Ctrl を押し続けると、選択要素の拡大縮小の刻み幅が 0.1 BU ずつになります。
- 拡大縮小中に ⇧ Shift を押し続けると、選択要素を非常に細かな刻み幅で拡大縮小できます。
- 拡大縮小中に ⇧ ShiftCtrl を押し続けると選択要素の拡大縮小が 0.01 BU ずつになります。
- S を押して数値を入力し、↵ Enter で確定します。

💡 座標系に影響される拡大縮小

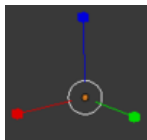
デフォルトでは、すべての拡大縮小はグローバル座標系で行われます。拡大縮小の軸は「軸キー」を2度押すと変更できます。例えば、S、X、Xと押せばデフォルトではローカル座標系で拡大縮小が行われるようになります。

[精密制御についてもっと詳しく»](#)

[数値を使ったトランスフォームについてもっと詳しく»](#)

[座標系についてもっと詳しく»](#)

マニピュレータを使った拡大縮小



拡大縮小マニ
ピュレータ

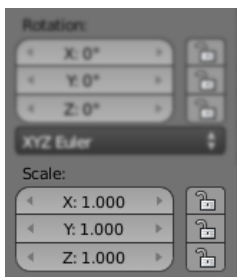
3Dビューのヘッダーで、マニピュレータが有効であることを確認してください(赤、緑、青の3つ組が選択されているかどうか)マニピュレータの種類を拡大縮小にしてください(下図でハイライトされている箱型のアイコン)



1. RMB で要素を選択します。
2. マニピュレータの色のついた軸のいずれかを LMB を使ってドラッグし、オブジェクトをその軸に沿って拡大縮小させます。マニピュレータといっしょに Shift、 Ctrl を使ったり、数値入力をしたりしてさらに細かく操作することもできます。
3. LMB を離すか、Space か Enter を押すと変更が確定します。また、RMB か Esc で変更をキャンセルできます。

[マニピュレータについてもっと詳しく»](#)

プロパティパネルを使った拡大縮小



拡大縮小トランスフォーム
のプロパティパネル

拡大縮小値は、プロパティパネル (N) でトランスフォームのスライダーの値を変更して指定することもできます。特定軸に沿った拡大縮小は、南京錠のアイコンで有効化/無効化できます。

[パネルについてもっと詳しく»](#)

[拡大縮小のモードについてもっと詳しく»](#)

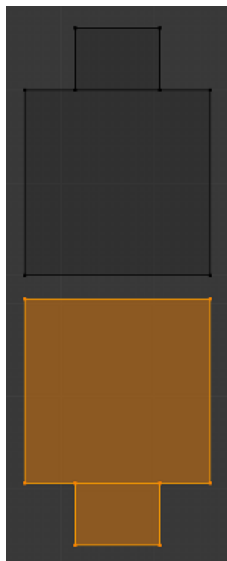
ミラー反転

モード: オブジェクトモードおよび編集モード

ホットキー: CtrlM

メニュー: オブジェクト/メッシュ » ミラー

解説



選択部分のミラー反転

オブジェクトやメッシュの選択要素をミラー反転すると、選択要素の反転版を作ります。選択要素の鏡映版の位置はピボットで決まります。ミラー反転の一般的な用途は、オブジェクトの半分をモデリングして複製し、それからミラー反転して逆転版を作り、モデルを完成させるというものです。ミラー反転した複製はミラーモディファイアで作る方法もあることに注意してください。

[ピボットについてもっと詳しく»](#)

[ミラーモディファイアについてもっと詳しく»](#)

使い方

特定のグローバル軸を中心に、選択要素をミラー反転するには:

CtrlMの後にX、Y、Zのいずれかを押します。

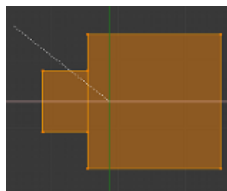
画像 選択部分のミラー反転 は、メッシュ要素の複製の後でこの操作を行った結果を示しています。

メッシュモードでは、ミラー反転操作の後に軸を表すキーを押すと、現在選択している座標系を基準にミラー反転できます。例えば座標系が法線である場合は:


CtrlMの後にXを押し、Xを再度押すと

選択要素を法線座標系のX軸に対してミラー反転します。

[座標系についてもっと詳しく»](#)



インタラクティブにミラー反転する

もしくは、MMB  を押し続けながらマウスを動かして、インタラクティブにオブジェクトをミラー反転することができます。

To Sphere

(球に変形)

モード: 編集モード

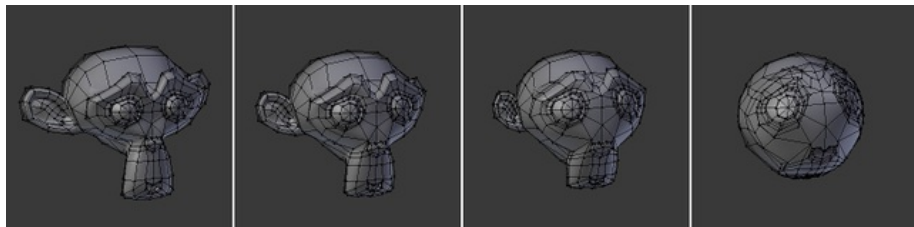
ホットキー: \diamond ShiftAltS

メニュー: Mesh(メッシュ) » Transform(トランスフォーム) » To Sphere(球へ変形)

Description

(解説)

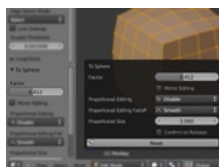
球へ変形(To Sphere) トランスフォームは選択要素を球状に変えます。下の画像「球面度が増していく Suzanne」は、ご覧のように 球へ変形 トランスフォームを Suzanne のメッシュに用いた結果です。



球面度が増していく Suzanne。上の画像は Suzanne のメッシュにそれぞれ強さ 0、0.25 (25%)、0.5 (50%)、1 (100%) の球へ変形を用いたものです。

Usage

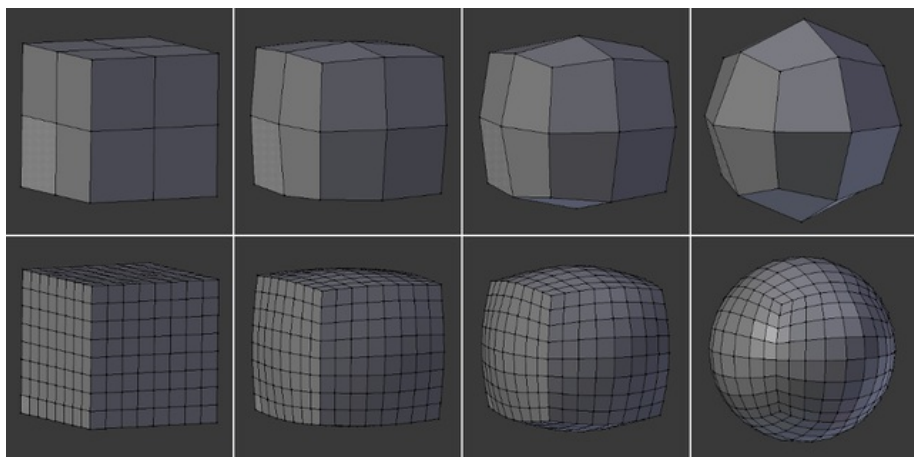
(使い方)



球へ変形機能

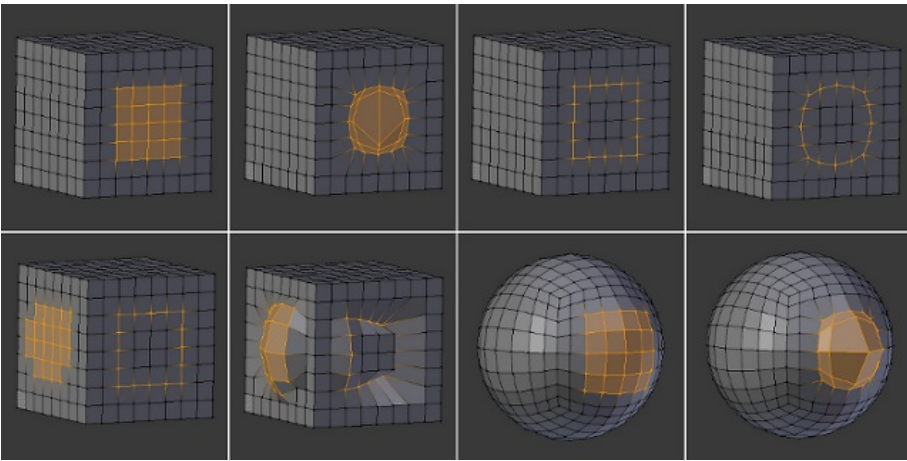
操作対象の要素を選択し、球へ変形トランスフォーム機能を実行します。球へ変形オプションはMesh(メッシュ) » Transform(トランスフォーム) » To Sphere(球へ変形) メニューか \diamond ShiftAltS を押して実行します。選択部分の球面度はマウスを動かして対話的に決めるか、0 から 1 の範囲で数字を入力して決めます。 \diamond Enter を押すとトランスフォームが確定します。確定したトランスフォームは F6 を押すか ツールシェルフ (T) で 強さ(Factor) スライダーを動かしてさらに編集できます。球へ変形 トランスフォーム確定後に別の操作をすると、このスライダーは現れません。

球へ変形トランスフォームの結果は選択しているメッシュ要素(頂点、面など)の数に左右されることに注意してください。下の画像では、対象となるメッシュ要素の数が多いほど結果が滑らかで球に近づくことがわかります。



細分化数を変えたキューブに球へ変形を用いたところ。球へ変形はキューブ全体に、左から順に強さ 0、0.25 (25%)、0.5 (50%)、1 (100%) で用いています。

球へ変形トランスフォームの結果は選択した要素の数や並び方によって変化します(下図参照)。



選択要素を変えながら球へ変形を用いたところ。

Shear

(せん断)

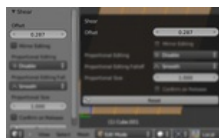
モード: オブジェクトモードおよび編集モード

ホットキー: ◊ ShiftCtrlAltS

メニュー: Object (オブジェクト)/Mesh (メッシュ)/Curve (曲線)/Surface (曲面) » Transform (トランスフォーム) » Shear (せん断)

Description

(解説)



せん断のオフセット機能

せん断 (Shearing) は平行な二つの面を互い違いに動かす移動方法です。このトランスフォームを行う間、選択要素は現在のビューの水平方向の軸に沿って移動します。軸の位置はピボットで決まります。この軸より「上」にあるものがすべて、マウスポイントと同じ方向に(しかも水平方向の軸と常に平行に)移動します。水平方向の軸より「下」にあるものはすべて、反対の方向に移動します。

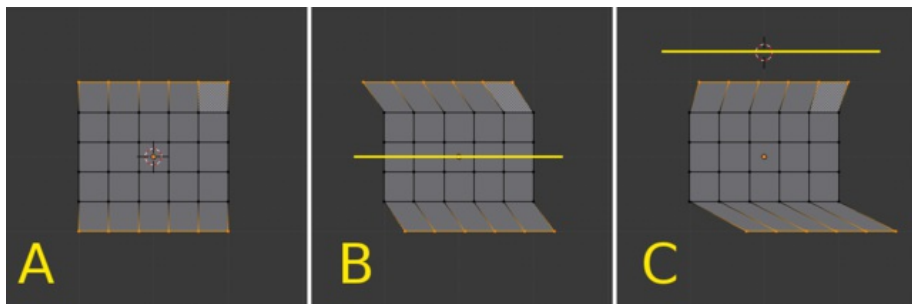
[ピボットについてもっと詳しく](#) »

Usage

(使い方)

操作対象の要素を選択して せん断 トランスフォーム機能を実行します。せん断 オプションは Object (オブジェクト)/Mesh (メッシュ)/Curve (曲線)/Surface (曲面) » Transform (トランスフォーム) » Shear (せん断) メニューか ◊ ShiftCtrlAltS を押して実行します。選択要素の移動量はマウスを動かして対話的に決めるか、数値を入力して決めます。↵ Enter を押すとトランスフォームを確定します。確定したトランスフォームは F6 を押すか、ツールシェルフ (T) でオフセットのスライダーを動かすと、さらに編集可能です。せん断 トランスフォームの確定後、他の操作を行うとこのスライダーは現れません。

せん断 トランスフォームの結果は選択要素の数や種類 (オブジェクト、頂点、面など) にも左右されることに注意してください。複数の違う要素に せん断 を使う下の例をご覧ください。



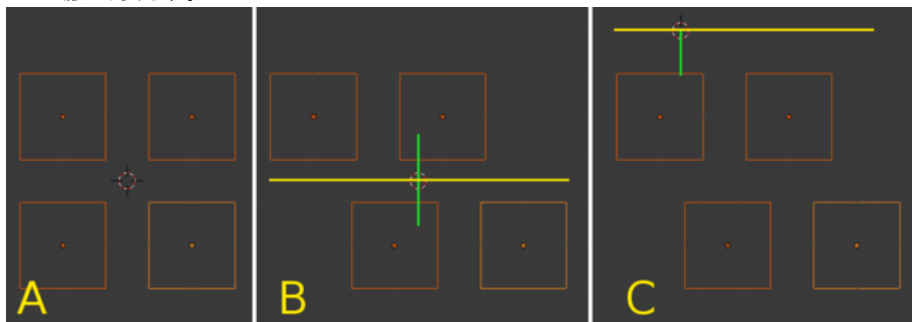
ピボットを変えてせん断トランスフォームを行った結果。さらなる情報が下のテキストにあります。

上の3枚の画像は、ピボットを変えながら選択頂点にせん断を行った結果を示しています。B の画像ではピボットが中点 (黄色の線) に設定されており、せん断中にマウスを左に動かしました。C の画像ではピボットが 3Dカーソルに設定されており、3Dカーソルはメッシュより上に置かれています (これも黄色の線です)。せん断中にマウスを左に動かすと、選択頂点は水平方向の軸より下にあるため右に動きます。



せん断トランスフォームの大きさ

せん断 選択要素に用られたトランスフォームの変形度合いは、水平方向の軸からの距離に完全に比例します。つまり、軸から遠いほど動きが激しくなります。



ピボットを変えて複数オブジェクトにせん断トランスフォームを行った結果。さらなる情報が下のテキストにあります。

上の3枚の画像は、ピボットを変えながら複数の選択オブジェクトにせん断を行った結果を示しています。B の画像では、ピボットが中点 (黄色の線) に設定されており、せん断中にマウスを左に動かしました。C の画像ではピボットが 3Dカーソルに設定されており、3Dカーソルはメッシュより上に置かれ

ています(これもまた黄色の線です)。せん断中にマウスを左に動かすと、選択されたすべてのオブジェクトは水平方向の軸より下にあるため右に動きます。ここでも、選択要素に用られたトランスフォームの結果は、水平方向の軸からの距離に完全に比例します。この場合は、下にあるオブジェクトほど上にあるものより大きく動きます。

Warp

(湾曲)

モード: オブジェクトモードおよび編集モード

ホットキー: ⇧ ShiftW

メニュー: Object (オブジェクト)/Mesh (メッシュ)/Curve (曲線)/Surface (サーフェス) » Transform (トランスフォーム) » Warp (湾曲)



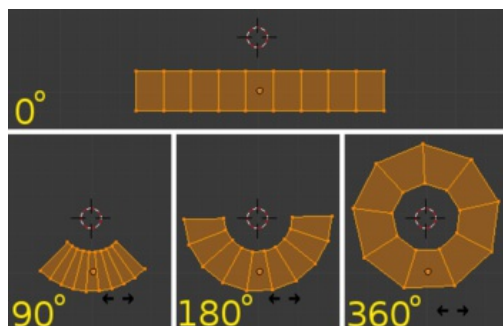
湾曲ツールのオプション

編集モードでは、湾曲トランスフォームは選択要素を 3Dカーソルの周囲に指定角度まで巻きつけます。このトランスフォームの結果は常に、3D カーソルの位置に左右されることに注意してください。ピボットは関係しません。湾曲トランスフォームの結果はビューにも左右されます。

オブジェクトモードでは、湾曲トランスフォームは選択要素を惑星の公転のように 3Dカーソルの周囲に移動させます。編集モードと同じく、ピボットは関係せず結果はビューに左右されます。

Usage

(使い方)



この例では平面が、表示角度まで 3Dカーソルに巻きつけられています

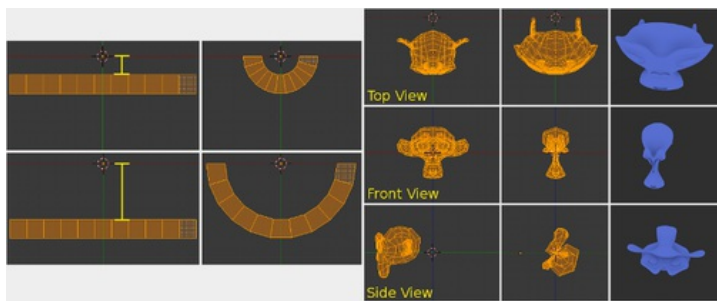
操作対象要素を選び、湾曲トランスフォーム機能を実行します。湾曲オプションは Object (オブジェクト)/Mesh (メッシュ)/Curve (曲線)/Surface (サーフェス) » Transform (トランスフォーム) » Warp (湾曲) メニューを選ぶか、⇧ ShiftW を押して実行します。選択要素が巻きつけられる量は、マウスを動かして対話的に決めるか、数字を入力して決めます。⇧ Enter を押すとトランスフォームを確定します。確定されたトランスフォームは F6 を押すかツールシェルフ(T)で角度のスライダーを動かすと、さらに編集できます。湾曲トランスフォームの確定後に他の操作を行うと、このスライダーは現れません。

Cursor position and view

(カーソル位置とビュー)

3Dカーソルの位置は湾曲トランスフォームの結果を変えるのに使えます。この節の例をご覧の通り、湾曲の半径は選択要素とカーソルの距離に左右されます。距離が広いほど半径が広がります。

湾曲トランスフォームの結果はまた、現在のビューにも左右されます。この節にある例では 180 度の湾曲トランスフォームをビューを変えながら同じ Suzanne のメッシュに使った要素を示しています。比較用にレンダリング画像もあります。



左側は湾曲トランスフォームがカーソル位置に影響されることを示しています。右の画像は現在のビューの影響を示しています。

テキストの湾曲

テキストを湾曲させたいなら、テキストオブジェクトはAltC を押して Mesh from Curve/Meta/Surf/Text (曲線/メタ/曲面/テキストからメッシュ) オプションを選び、メッシュに変換する必要があります。

Example

(例)



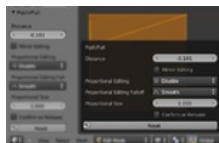
ロゴの周りに巻きつけたテキスト。Blender ロゴとテキストは別のオブジェクトとして作られています。テキストはメッシュに変換され、Blender ロゴの周りに「湾曲」しました。

押す/引く(Push/Pull)

モード: オブジェクトモードおよび編集モード

メニュー: オブジェクト/メッシュ » トランスフォーム » 押す/引く

解説



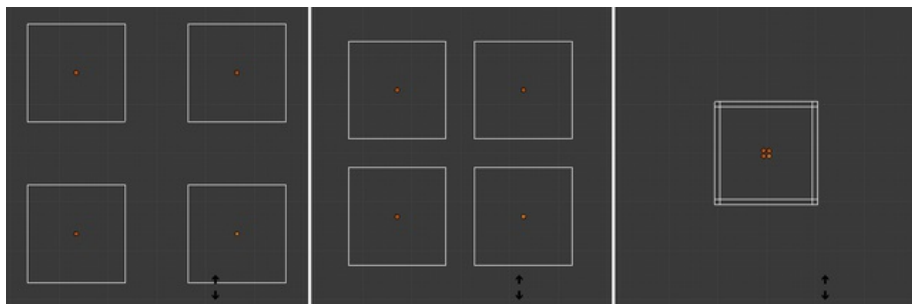
押す/引く距離

押す/引くは選択要素(オブジェクト、頂点、辺、面)をお互いに近づける(押す)か遠ざける(引く)かします。具体的には、各要素は中心から同じ距離だけ近づくか、遠ざかるかします。この距離はマウスの上への移動(=押す)、下への移動(=引く)、数値入力、もしくはスライダーコントロールから指定します。

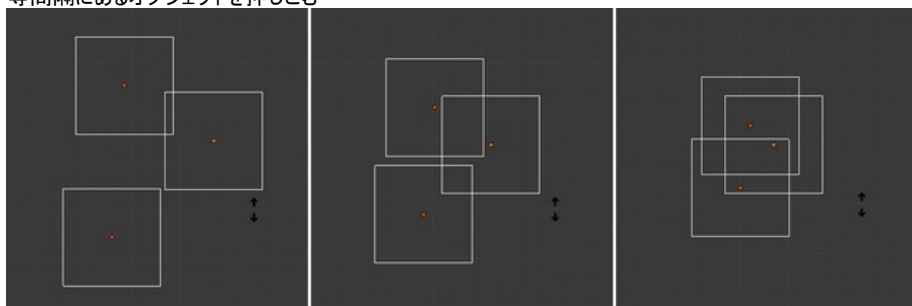
使い方

操作したい要素を選んで、押す/引く変形機能を実行します。押す/引くの実行は オブジェクト/メッシュ » トランスフォーム » 押す/引く メニューまたは \diamond ShiftAを押して検索メニューから Push か Pull を検索して行います。移動量の指定はマウスを動かしながらインタラクティブに行うか、数値入力で行います。 \downarrow Enter を押すとトランスフォームを確定します。確定されたトランスフォームは F6 かツールシェルフ (T) の距離のスライダーを動かすことでさらに編集できます。距離のスライダーは 押す/引く変形の確定後、他の操作をすると表示されません。

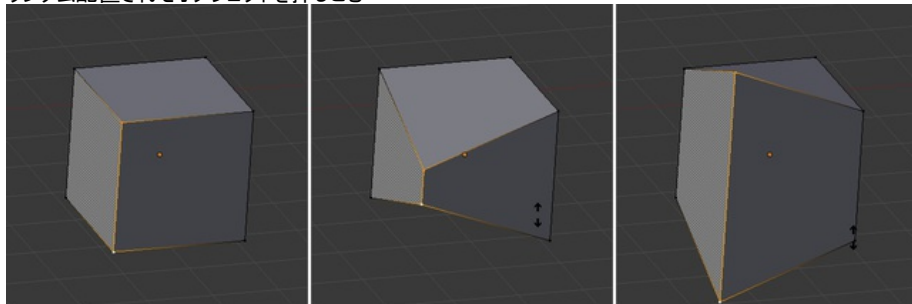
押す/引くの結果は選択要素(オブジェクト、頂点、面など)の数や種類によっても変わることにご注意ください。以下は様々な要素への 押す/引く の実行結果です。



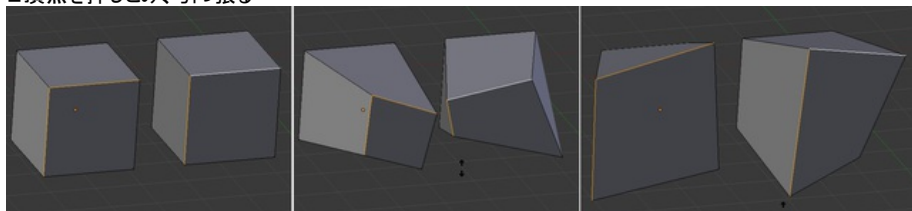
等間隔にあるオブジェクトを押しこむ



ランダム配置されたオブジェクトを押しこむ



2頂点を押しこみ、引っ張る



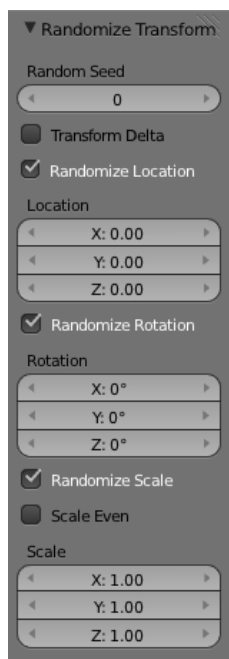
別々のメッシュにある辺を押し込み、引っ張る

Randomize Transform

(ランダムなトランスフォーム)

モード: Object モード

メニュー: Object(オブジェクト) » Transform(トランスフォーム) » Random Transform(ランダムトランスフォーム)



ランダムトランスフォームのオプション

ランダムトランスフォームツールを使うと、単独か複数のオブジェクトにランダムな移動値、回転値、拡大縮小値を適用できます。複数オブジェクトに適用すると各オブジェクトにシード値が与えられるため、それぞれの結果が異なります。

Options

(オプション)

ランダムシード(Random Seed)

ランダムトランスフォームのオフセットを表します。シードが変わると新たな結果が作られます。

トランスフォームデルタ(Transform Delta)

標準のトランスフォームでなくデルタ変換の値をランダム化します。[デルタ変換\(Delta Transforms\)](#) をご覧ください。

位置をランダムに(Randomize Location)

位置をランダムにします。

位置(Location)

各軸上をオブジェクトが移動できる最大距離です。

回転をランダムに(Randomize Rotation)

回転値をランダムにします。

回転(Rotation)

軸ごとの最大回転角度です。

スケールをランダムに(Randomize Scale)

拡大縮小値をランダムにします。

等しくスケーリング(Scale Even)

すべての軸で同じスケールを使います。

拡大縮小(Scale)

軸ごとの最大拡大縮小値です。

トランスフォームコントロール

トランスフォームコントロールは利用可能な[トランスフォーム](#)の効果を修正・調整する際に利用します。

以下に示すページでは、利用可能なコントロールのオプションについて詳細に記述しています。

トランスフォームの量

- [トランスフォームの精度](#)
- [数値的トランスフォーム](#)
- [トランスフォームプロパティ](#)
- [トランスフォームのリセット](#)
- [比例する影響の編集](#)

トランスフォームの方向

- [マニピュレータ](#)
- [トランスフォームの方向](#)
- [軸固定](#)

トランスフォームの基準点

- [ピボット](#)
- [アクティブオブジェクト](#)
- [個別の原点](#)
- [3D カーソル](#)
- [中点](#)
- [バウンディングボックスの中心](#)

トランスフォームのスナップ

- [スナップ](#)
- [メッシュに対するスナップ](#)

Precision

(精密操作)

モード: オブジェクトモードおよび編集モード

ホットキー: Ctrl または ⇧ Shift、同時使用可

Description

(解説)

トランスフォーム操作(移動/回転/拡大縮小)中に Ctrl か ⇧ Shift を押し続けると、トランスフォームを固定値で行うか、非常に小さな値で行うか、その両方を行うことができます。トランスフォームの大きさは(トランスフォーム中に) 3Dビューのヘッダーの左下隅に表示されます。Ctrl や ⇧ Shift を離すと、動きは通常の操作モードのものに戻ります。

[ウィンドウヘッダーについて詳しく»](#)

Usage

(使い方)

With hotkeys

(ホットキーを使う方法)

G、R、S のいずれかを押して、Ctrl、⇧ Shift、Ctrl⇧ Shift のいずれかを押し続けます。

With the Transform Manipulator

(マニピュレータを使う方法)

Ctrl、⇧ Shift、Ctrl⇧ Shift のいずれかを押し続けながらマニピュレータの適切なハンドルをクリックします。それからマウスを望みの方向に動かします。逆の動作も有効です。マニピュレータのハンドルをクリックしてから、精密操作のショートカットキーを押します。

[マニピュレータについてもっと詳しく»](#)



他の操作との組み合わせ

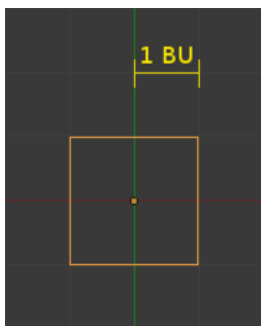
このページにある精密操作はすべて、[軸ロック](#)と組み合わせることができ、どの[ピボット](#)でも使えます。

Holding CTRL

(CTRL を押し続ける操作)

Grab/move transformations

(移動)



1 Blender単位 (Blender Unit) - デフォルトの拡大率で表示

デフォルトの拡大率で移動操作を行うとき、Ctrl を押し続けると選択要素を 1 Blender 単位 (Blender Unit、1 BU) ずつ動かします (明るい灰色の線同士の距離です)。次の灰色の線のセットが見えるまで画面を拡大すると、Ctrl 移動の刻み幅は 1/10 BU ずつになります。さらに次の灰色の線のセットが見えるまで拡大すると、刻み幅は 1/100 BU になります。拡大率が限度に来るまで、同様に繰り返されます。逆に、画面の縮小では 10 BU、100 BU といった刻み幅になります。

[ズームについてもっと詳しく»](#)

Rotation transformations

(回転)

Ctrl を押し続けると回転が 5 度ずつになります。

Scale transformations

(拡大縮小)

Ctrl を押し続けるとサイズ変更が 0.1 BU ごとになります。

スナップのモード

[要素にスナップ](#) オプションを有効にしていると、Ctrl を押し続けた時、選択要素が最も近い要素にスナップすることに注意してください。

[スナップについてもっと詳しく](#) »

Holding SHIFT

(SHIFT を押し続ける操作)

トランスフォーム中に ◊ Shift を押し続けると、固定の刻み幅を使わない、非常に精密な操作ができます。マウスで画面に対して大きな動きをしても、選択要素は少ししか動きません。

Holding CTRL and SHIFT

(CTRL と SHIFT を押し続ける)

Grab/move transformations

(移動)

デフォルトの画面の拡大率では、移動操作で Ctrl◊ Shift を押し続けると選択要素の移動が 1/10 Blender 単位刻みになります。また、画面の拡大率によらず、Ctrl◊ Shift を押し続けるとトランスフォームの刻み幅は常に、Ctrl だけを押し続けているときの 1/10 になります。

Rotation transformations

(回転)

Ctrl◊ Shift を押し続けると1度ずつ回転します。

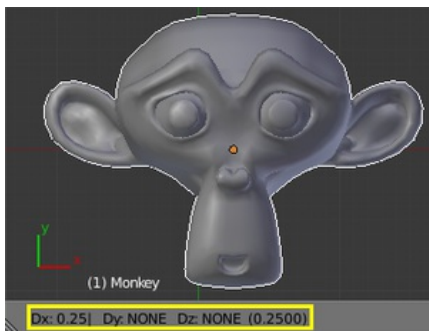
Scale transformations

(拡大縮小)

Ctrl◊ Shift を押し続けると 0.01 BU の刻み幅で拡大縮小します。

Numeric input

(数値入力)



3D ウィンドウヘッダーの数値入力

トランスフォームにマウスを使うのは便利ですが、もっと精密な制御が必要になれば、数値入力を使うことができます。G、R、Sのいずれかを押した後、トランスフォームの強さを表す数字を入力します。

入力した値を3D ウィンドウヘッダーの左下隅の表示で確認できます。負の数値や小数も、マイナス (-) とピリオド (.) をそれぞれ使って入力可能です。

Translation

(移動)

オブジェクト、頂点、面、辺の選択要素を移動させるには、G を押して数値を入力します。デフォルトでは、他にキーを押さなければ移動は X 軸に沿って行われます。移動を確定するには \leftarrow Enter か LMB を押します。移動をキャンセルするには Esc か RMB を押します。値を打ち間違えた場合は、 \leftarrow Backspace を押して現在の入力を取り消し、新しい値を再度入力してください。

複数の軸に数値を入力するには、軸を指定するキーの後に \leftrightarrow Tab キーを押してください。例えばオブジェクトを3軸すべての方向に 1 Blender (BU) 単位だけ動かすには、G、1、 \leftrightarrow Tab、1、 \leftrightarrow Tab、1 と押します。オブジェクトは X 軸に沿って 1 BU 動き、Y 軸、Z 軸に沿ってもそれぞれ 1 BU ずつ動きます。

数値入力は軸ロックと組み合わせ、特定軸に移動を制限することもできます。このためには G に続けて X、Y、Z のいずれかを押してその軸を指定してください。それから移動量を 0-9 で指定し、さらに \leftarrow Enter で確定します。X、Y、Z のいずれかを押すと初めはグローバル軸に制限されます。X、Y、Z のいずれかを再度押すと、移動は 3D ウィンドウヘッダーの座標軸 で設定されている座標系に制限されます。

[座標系についてもっと詳しく](#) »

[軸ロックについてもっと詳しく](#) »

Rotation

(回転)

時計回りの回転値を指定するには、R、0-9 を押して、 \leftarrow Enter で確定します。反時計回りの回転値を指定するには、R、-, 0-9 を押して、 \leftarrow Enter で確定します。なお、時計回りの 270 度は -90 度と等しくなります。

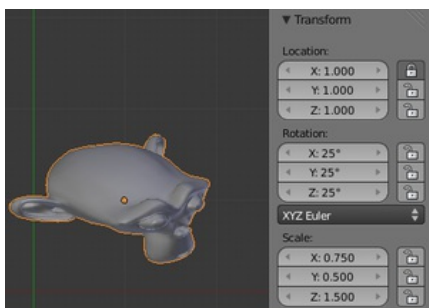
Scaling

(拡大縮小)

オブジェクト、面、辺は S、0-9 を押して拡大縮小し、その後 \leftarrow Enter を押して確定します。拡大縮小トランスフォームは特定軸に制限することもでき、X、Y、Z のいずれかを S の後に押します。基本的には数値入力での拡大縮小は移動とまったく同じ方法で動作します。主な違いは、デフォルトでは拡大縮小が3軸すべてに等しく用いられることです。例えば、S、0.5、 \leftarrow Enter という一連のキー入力はオブジェクトを3軸すべての方向に 0.5 ずつスケールします。

Numeric input via the Properties shelf

(プロパティシェルフを使った数値入力)



トランスフォームはProperties/パネルのトランスフォーム欄からも入力可能です。

トランスフォームはProperties/パネル(N)のトランスフォーム欄からも入力可能です。トランスフォーム欄では、南京錠のアイコンをクリックして、トランス

フォームを特定軸に制限することもできます。

Transform Properties

(トランスフォームのプロパティ)

各オブジェクトは自分の位置、方向、拡大縮小値を記憶します。この値の数値入力やリセット、適用 (apply) 操作をする機会があるかもしれません。

Transform Properties Panel

(トランスフォームのプロパティパネル)

モード: 編集 および オブジェクトモード

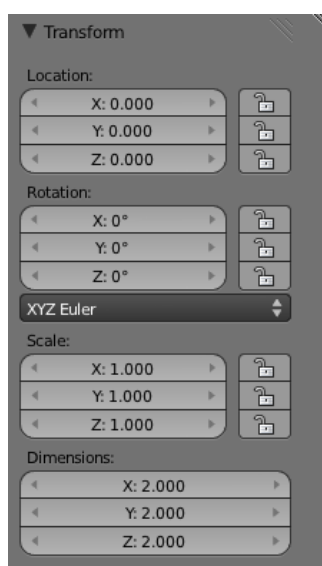
ホットキー: N

メニュー: View(ビュー) » Properties(プロパティ)

オブジェクトモードのとき、プロパティパネルの Transform 欄でオブジェクトの位置、回転やその他のプロパティの閲覧、手動/数値入力による調節ができます。編集モードでは主に、頂点や頂点グループ(辺/面を含む)の中心の、正確な座標を入力できます。オブジェクトの種類によって、編集モードで Transform 欄にあるオプションは変わります。[モデリングの章](#)のそれぞれの説明をご覧ください。

Options in Object mode

(オブジェクトモードにおけるオプション)



オブジェクトモードでの Transform Properties パネル

Location X, Location Y, Location Z

グローバル座標におけるオブジェクトの中心位置

Rotation X, Rotation Y, Rotation Z

オブジェクトの方向。グローバルな座標軸とオブジェクト自身の中心に相対的です。

Scale X, Scale Y, Scale Z

オブジェクトの拡大縮小値です。ローカル座標におけるオブジェクトの中心に相対的です(すなわち Scale X 値はローカル X 軸方向の拡大縮小値を表します)。オブジェクト(立方体、球他)は作られた時に、ローカルの各方向に対して 1 blender unit の拡大縮小値を与えられます。オブジェクトを大きくしたり小さくしたりするには、お望みの寸法に拡大縮小してください。





Dimensions X, Dimensions Y, Dimensions Z

オブジェクトの基本寸法で、外側の端からもう一方の端までを定規で測ったものと考えてください(blender unit 単位)。面数の多いもの向けに、この項目はバウンディングボックス(ローカル座標軸に沿うもの。オブジェクトをちょうど包み込む大きさの厚紙の箱を想像してください)の寸法を表します。

このパネルを、オブジェクトの位置、回転、拡大縮小などのトランスフォームプロパティの閲覧や編集に使ってください。これらの項目はオブジェクトの中心を変え、そのオブジェクトの *頂点* や面すべての、様態 (aspect) に影響します。

Ipo に関する注意点



位置、回転、拡大縮小の値は Ipo キーフレームからも影響を受けます。したがってオブジェクトに関連付けられた Ipo キーがある場合は、このパネルで変更を行ったあと、確実に Ipo キーをセットし直してください。そうしないとフレームが変わったとき、変更は失われます(手動セットしたプロパティを Ipo キーが上書きします)。

項目によってはスクロール領域など、さらに別の操作手段や機能を持つものがあります。このタイプの項目を編集するときは、LMB  の代わりに  Shift LMB  を使うほうが簡単です。項目の編集を終えたら、項目の入力欄の外側をクリックするか、 Enter を押して変更を確定します。変更は表示用ウィンドウに即座に反映されます。キャンセルするには Esc を押してください。入力欄の他の機能については [インターフェイスの章](#) にさらなる説明があります。

Transform Properties Locking

(トランスフォームプロパティのロック)

位置、回転、拡大縮小欄をロックする機能を使うと、そのトランスフォームプロパティをプロパティパネルだけで調節できます。いったんロックを有効にすると、他の方法でのトランスフォームはすべて遮断されます。たとえば Location X 項目をロックすれば、マウスでグローバル X 軸に沿ってオブジェクトを移動させることができなくなります。ただ、依然として Location X 入力欄を使った移動はできます。ロック機能は、変更をパネルのみで行えるようにする厳しい制約と考えてください。

項目をロックするには、項目の隣にある南京錠のアイコンをクリックします。アイコンが  のように表示されていればロックされておらず、 のように表示されていればロックされています。

Clear Object transformations

(オブジェクトのトランスフォームの消去)

モード: オブジェクトモード

ホットキー: AltG, AltS, AltR, AltO

メニュー: Object(オブジェクト) » Clear(クリア) » Clear Location/Clear Scale/Clear Rotation/Clear Origin(位置/回転/拡大縮小/原点)

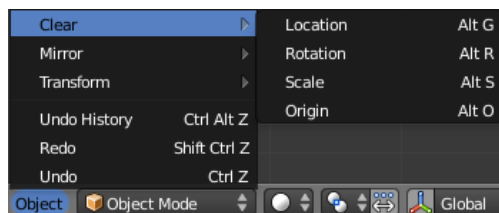
Description

(解説)

トランスフォームの消去は、単にトランスフォーム値をリセットします。オブジェクト位置と回転値は 0 に戻り、拡大縮小値は 1 に戻ります。

Clear Options

(クリアのオプション)



トランスフォームクリアのメニュー

位置のクリア(Clear Location) AltG

選択要素の位置をクリア(リセット)します。選択要素を座標 0,0,0 に戻します。

拡大縮小のクリア(Clear Scale) AltS

選択要素の拡大縮小をクリア(リセット)します。作成時のサイズに戻します。

回転のクリア(Clear Rotation) AltR

選択要素の回転をクリア(リセット)します。各平面に対して 0度の回転にします。

原点のクリア(Clear Origin) AltO

子オブジェクトの原点をクリア(リセット)します。子オブジェクトは親の座標に移動します。

Apply Object transformations

(オブジェクトのトランスフォームを適用する)

モード: オブジェクトモード

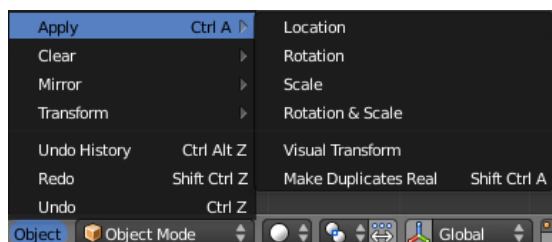
ホットキー: CtrlA

メニュー: Object(オブジェクト) » Apply(適用)

トランスフォーム値の適用は基本的に、オブジェクトの位置、回転、拡大縮小値をリセットしますが、実際にオブジェクトに何かを行うことはありません。中心位置は原点に移動し、トランスフォーム値はゼロにセットされます。拡大縮小については、拡大縮小値は1に戻ります。トランスフォームを適用するには Object(オブジェクト) メニューから Apply(適用) サブメニューを選ぶか、ショートカット CtrlA を使い、適用させるトランスフォームを選んでください。

Apply Options

(適用のオプション)



トランスフォーム適用のメニュー

位置の適用(Apply Location) CtrlA

選択要素の位置を適用(設定)します。Blender は、現在の位置が各面に対する 0 位置であるとみなします。言い換えれば、選択要素は移動せず、現在の位置が「デフォルト位置」であるとみなします。オブジェクトの中心が実際の 0,0,0 (各ビューで色のついた線が交差する部分で)に設定されます

回転の適用(Apply Rotation) CtrlA

選択要素の角度を適用(設定)します。Blender は、現在の回転が各平面に対して 0 度であるとみなします。言い換えれば、選択要素は回転せず、現在の回転が「デフォルトの回転」であるとみなします。

拡大縮小の適用 (Apply Scale) CtrlA

選択要素の拡大縮小を適用(設定)します。Blenderは現在の拡大縮小が各平面に対して0であるとみなします。言い換えれば、選択要素は拡大縮小されず、現在の拡大縮小が「デフォルトの拡大縮小」であるとみなされます。

回転と拡大縮小を適用 (Apply Rotation and Scale) CtrlA

選択要素の回転と拡大縮小を適用(設定)します。上の2つを適用するのと同じです。

ビジュアルトランスフォーム (Apply Visual Transform) CtrlA

コンストレイントの結果を適用(設定)し、オブジェクトの位置/回転/拡大縮小にこれを適用します。さらに詳しい議論は次の記事をご覧ください:
[Apply visual transform](#).

複製を実体化 (Make Duplicate Real) ⇧ ShiftCtrlA

オブジェクトにアタッチされているすべての複製を実体化し、編集できるようにします。

マニピュレータ

モード: オブジェクトモードおよび編集モード


ホットキー: CtrlSpace

通常の変形操作(移動G、回転R、拡大縮小S)は **軸ロック**と組み合わせて、オブジェクトをどの軸に沿っても操作できます。しかしながら、これだけでは不十分なこともあるでしょう。例えばランダムに回したオブジェクトの面に並行に、他の面を移動したい場合です。こういったときに **変換マニピュレータ**が役立つかもしれません。



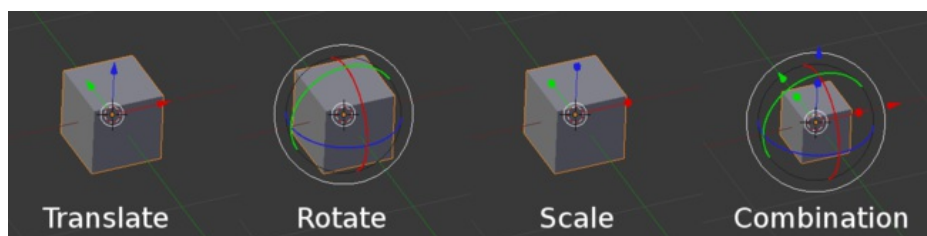
ウィンドウヘッダーにあるマニピュレータのオプション

変換マニピュレータは座標系を視覚的に表示し、どの軸/モード/3Dビューの方向に対しても移動/回転/拡大縮小が可能です。マニピュレータを有効にするには、ウィンドウヘッダーのマニピュレータオプション欄にある軸アイコンをクリックするか、ショートカットCtrlSpace を使います。

変換コマンドそれぞれに、別々のマニピュレータがあります。マニピュレータはどれも、別々に使うことも、組み合わせて使うこともできます。複数のマニピュレータアイコン(矢、円弧、箱)を **⇧ Shift LMB** すると、マニピュレータオプションを組み合わせられます。






マニピュレータには3Dビュー ウィンドウのヘッダーからアクセスできます:

- 軸: マニピュレータの有効化/無効化
- 矢印: 移動
- 円弧: 回転
- 箱: 拡大縮小
- 座標系メニュー: 座標系の選択



マニピュレータオプション

マニピュレータの操作

- Ctrlを押すと動きの速さを制限します。マニピュレータハンドルを LMB した「後に」**⇧ Shift**を押し続けると動きを緩やかにします。
- **⇧ Shift** を押し続けながらハンドルを LMB クリックすると、マニピュレータの動きが他の2つの軸に相対的になります(クリックした後は **⇧ Shift** を離しても構いません)。例えば、**⇧ Shift** のあと移動マニピュレータの Z軸のハンドルを LMB クリックすれば、動きはXY平面のみに制限されません。
- 回転モードで白い輪(回転マニピュレータの周りにある一番大きな輪)を LMB クリックすると、Rを押すのと同じ操作になります。
- 回転モードで灰色の輪(回転マニピュレータの中央の、内側の小さな輪)を LMB クリックすると、Rを2回押すのと同じ操作になります。これはトラックボール回転を開始します。

[変換の制限についてさらに読む](#) » (eng)

[軸ロックについてさらに読む](#) »

[トラックボール回転についてさらに読む](#) » (eng)

マニピュレータの設定



マニピュレータの設定

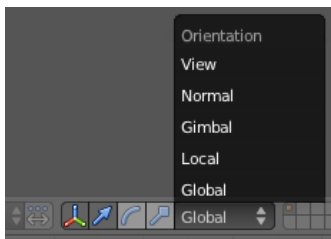
マニピュレータの設定(大きさ等)はユーザー設定(User_Preferences) ウィンドウの インタフェース(Interface) タブにあります。

- 大きさ(Size): マニピュレータの直径です。単位は 10 pixelです。
- ハンドルの大きさ(Handle Size): マニピュレータのハンドルの大きさです。マニピュレータの半径(大きさ/2)に対するパーセンテージです。
- ホットスポット(Hotspot): マニピュレータのハンドルのクリック可能範囲にあたる、ホットスポットの大きさです。単位は pixel です。

座標系 (Transform Orientation) の選択

モード: オブジェクトモードおよび編集モード

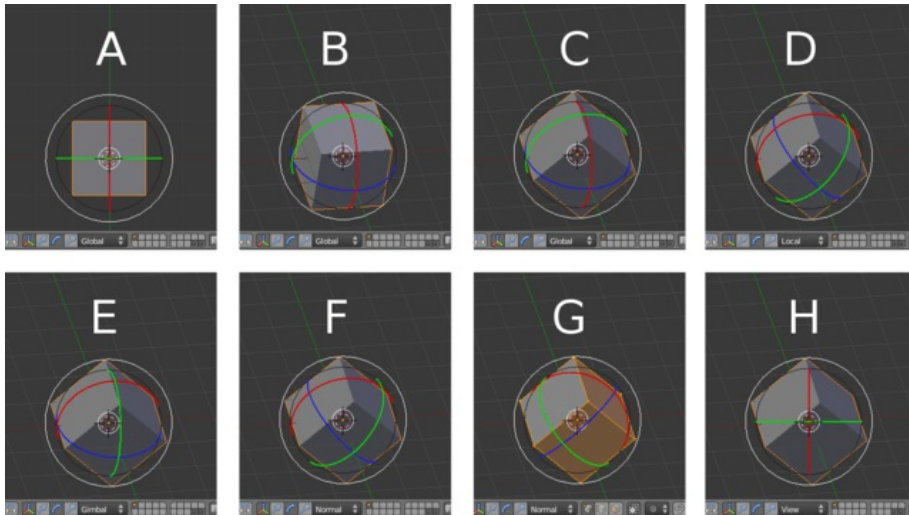
ホットキー: AltSpace



座標系オプション

座標系メニューから、**座標系**をグローバル(global)、ローカル(local)、ジンバル(gimbal)、法線(normal)、またはビュー(view)に変えることもできます。下の画像は、立方体の、様々な座標系が選ばれた回転マニピュレータです。座標系によってマニピュレータがどのように変化するか、注意してください(AとFを比べてみてください)。

同様に、法線座標系が選ばれていると(FとG)マニピュレータがオブジェクトモードと編集モードでマニピュレータがどのように変化するか注意してください。法線座標系のマニピュレータは、編集モードで何を選んでいるかによっても変化します。マニピュレータの方向は選択要素の法線にもとづいていて、いくつどの面を/辺を/頂点を選択しているかによって変わるということです。



マニピュレータの座標系オプション

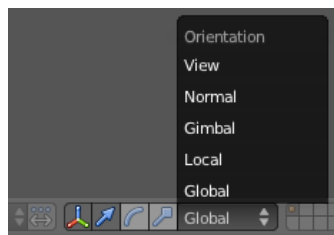
- A: デフォルトの上面図/グローバル座標系
- B: 少し回転/グローバル座標系
- C: ランダム回転/グローバル座標系
- D: ランダム回転/ローカル座標系
- E: ランダム回転/ジンバル座標系
- F: ランダム回転/法線座標系
- G: ランダム回転/法線座標系/頂点を選択中
- H: ランダム回転/ビュー座標系

Transform Orientations

(座標系)

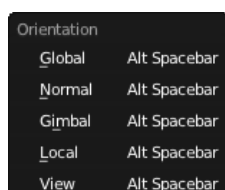
モード: オブジェクトモードおよび編集モード

ホットキー: AltSpace



座標系の選択メニュー

座標系はトランスフォーム (移動/回転/拡大縮小) の挙動に影響します。(軸ロックのような) 移動制限だけでなく、3Dマニピュレータ(選択部分中央にあるウィジェット)にも効果が現れます。GXを押すとグローバルの X軸に制限し、GXXと押すと現在の座標系の X軸に制限します。



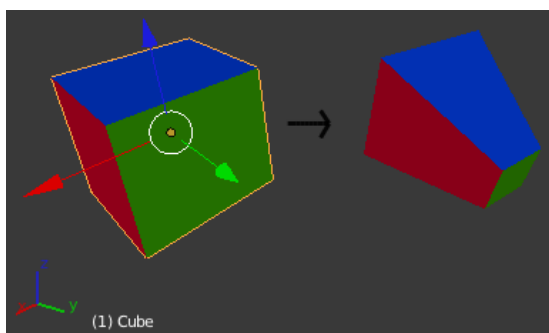
Alt+Space メニュー

座標系オプションは 3D ビューのヘッダーか(またはフッター。デフォルトではビューの下部にあります)、AltSpace、もしくは 3D ビューのヘッダーにある 座標系 から設定できます。

4つの組み込みのオプションに加えて、[カスタム座標系](#)の定義ができます。

Our Demo Cube

(キューブによる見本)



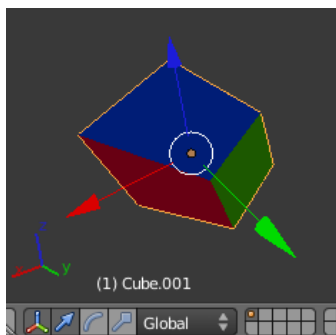
様々な特徴を実際にお見せするため、デフォルトのキューブに色をつけ、ローカルの Z軸および X軸に沿って -15° 回転させ、"y"面を小さくしました。

ふたつの点に注意してください:

- 左下隅の "小さな軸" はグローバル座標系の x/y/z 軸を表しています。
- 選択部分から生えている [オブジェクトマニピュレータ](#) ウィジェットは現在の座標系を表しています。
 - マニピュレータのいずれかの軸を LMB クリックすると、この方向にだけ移動ができるようになります。同様のキー操作は GZZ です。
 - \diamond Shift LMB クリックすると、クリックした軸をロックして、残りの2つの軸のある平面に移動を制限します。同様のキー操作は $G\diamond$ ShiftZ \diamond ShiftZ です。

Orientations

(座標系)

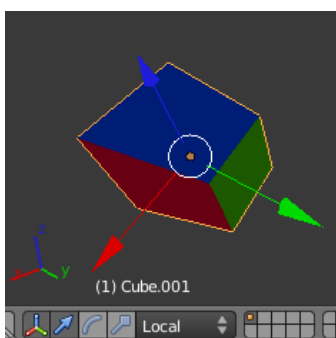


グローバル (Global)

Global

(グローバル)

マニピュレータはグローバル軸に一致します。グローバル座標系では、x、y、zはワールド座標のx、y、zと同じです。このモードが選ばれると、オブジェクトのローカル座標はグローバル座標の影響を受けます。これはシーン内にオブジェクトを置く際は役立ちます。軸を限定するには、Gと対象軸のキーを押してください。ローカル軸に限定するには、対象軸のキーを二回入力してください。グローバルとローカルの違いは、原点がオブジェクトの正確な中心に置かれておらず、グローバル座標に一致しないオブジェクトがあるとよくわかります。

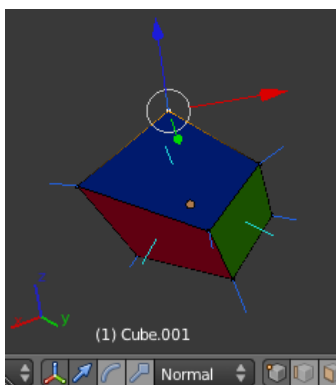


ローカル (Local)

Local

(ローカル)

マニピュレータはローカル軸に一致します。ここで、マニピュレータが少し傾いていることに注意してください(特にオブジェクトのY軸、緑の矢印でよくわかります)。これはオブジェクトを15°回転させていたためです。これはローカル座標とグローバル座標の違いを示しています。もしオブジェクトをX軸に沿って90°回転させていれば、オブジェクトの「上」がワールドの「前」になっていたでしょう。もしくはオブジェクトのZ軸がワールドのY軸になっていたでしょう。この座標系はインターフェースの多くの部分に影響するので、この区別を理解することが重要です。



法線 (Normal)

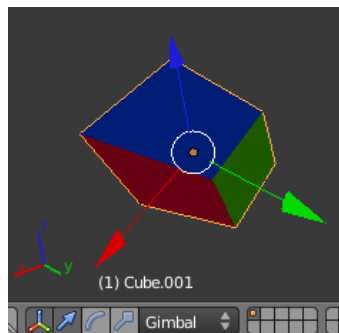
Normal

(法線)

マニピュレータのZ軸は選択オブジェクトの法線ベクトルに一致します。オブジェクトモードではこれはローカル座標系と同じですが、編集モードではもっと面白くなります。

ご覧のように、薄い青の線は面の法線を示しており、濃い青の線は頂点の法線を示しています(NプロパティパネルのMesh Display(メッシュ表示) » Normals(法線)のFace and Vertex(面と頂点)をオンにしています)。いずれかの面を選択すると、マニピュレータのZ軸がその法線に沿います。頂点選択モードでも同じことができます。辺選択は違います。辺選択モードでは、Z軸は選択された辺に沿います(このため、XとYの方向を知るためにマニピュレータウィジェットを見る必要があるでしょう)。複数要素を選択すると、Z軸はそれらの法線の平均に向きます。

この便利さを説明する非常に良い例は、頂点選択モードにあります。頂点を選び、GZZを押してメッシュから引き離し、メッシュに押し込んでみてください。もっと便利にするためには、近くの頂点を選び、 \diamond ShiftRを押して同じ動作を繰り返してみてください。



ジンバル(Gimbal)

Gimbal

(ジンバル)

ジンバルの挙動は現在の[回転のモード](#)(N プロパティパネルの3Dビューの一番上にあるTransform(トランスフォーム)で設定できます)に強く依存しています。

XYZ Euler(XYZ オイラー角)

デフォルトの回転モードです。オブジェクトのマニピュレータのZ軸は常にグローバルのZ軸を指し、それ以外の2つの軸はお互いに直角のままです。

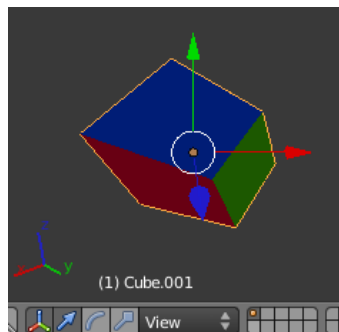
他のオイラー角の回転モードでは、最後に使われた軸がマニピュレータを固定するのに使われます。したがって YZX オイラー角 (YZX Euler) では、マニピュレータの X軸がグローバルの X軸と同じになります。

Axis Angle(軸角度)

x、y、z 値はオブジェクトの原点に相対的な位置を表し、原点には仮想的に「串(skewer)」を通します。w 値は串の回転です。マニピュレータのZ軸はこの串に沿っています。

Quaternion(クォータニオン)

クォータニオン回転は、オイラー角や軸角度の回転モードとは非常に異なります。マニピュレータはローカルモードと同じように動作します。



ビュー(View)

View

(ビュー)

このマニピュレータは 3Dビューと一致します。Y → 上下、X → 左右、Z → あなたに近づく/遠のく。

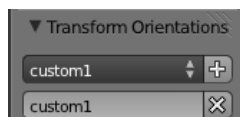
GXXを使ってビューの一つの軸に動きを制限できます。

Custom Orientations

(カスタム座標軸)

モード: オブジェクトモードおよび編集モード

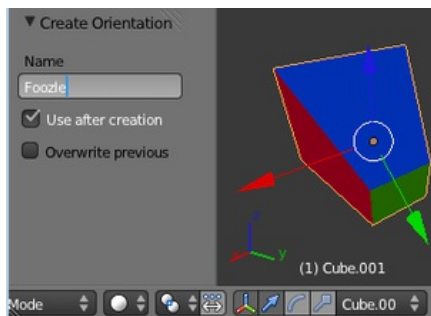
ホットキー: CtrlAltSpace



カスタム座標系

オブジェクトやメッシュ要素を使って、独自の座標系を定義できます。オブジェクトから定義されたカスタム座標系は、オブジェクトのローカル座標系を使います。選択メッシュ要素(頂点、辺、面)から定義されると、選択要素の法線座標系を使います。

Transform Orientations (座標系) パネルは N プロパティパネルにあり、アクティブな座標系を選んだり、カスタム座標系の追加削除をしたりするなどの座標系の管理に使います。



カスタム座標系の改名

何であれ選択していたものの名前が座標系のデフォルトの名前になります。辺であれば「辺(Edge)」になり、オブジェクトならそのオブジェクト名前になり、といった具合です。ツールシェルフ(3DビューでT)ではCtrlAltSpace を押すと、カスタム座標系を改名できます。

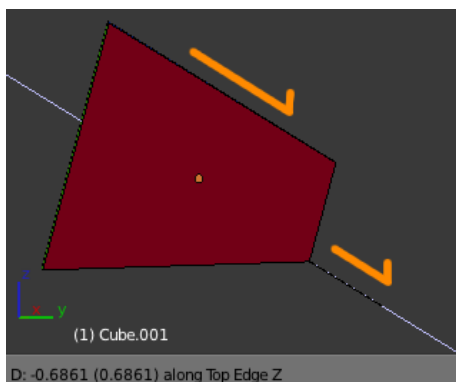



図 1

精密なメッシュを作る際に、カスタム座標系の作成テクニックは重要になり得ます。図 1 では次の効果を実行しています:

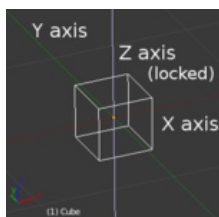
1. オブジェクトの一番上の傾いた辺を選びます
2. CtrlAltSpace でカスタム座標系を作り、"Top Edge" に改名します
3. オブジェクトの下と、右にある辺を選びます
4. E で押し出します
5. 押し出しのデフォルトの動きを RMB  か Escape を押してキャンセルします
6. G を押して移動を再開します
7. ZZ を押して「一番上の辺」方向に動きを制限します

Axis Locking

(軸ロック)

Description

(解説)




軸のロック

オブジェクトモードと編集モードの [トランスフォーム\(移動/回転/拡大縮小\)](#) (編集モードでの押し出しも同様)は、現在の[座標系](#) の特定の座標軸にロックすることができます。トランスフォームを特定軸にロックすると、変形を一方向に制限できます。

Usage

(使い方)

ロックされた軸は、されていない軸よりも明るい色で表示されます。例えば、右の画像では、Z軸が明るい青色で表示され、移動がこの軸に制限されています。この例は次の2つの方法で再現できます:

- Gを押して移動可能にし、Zで移動をZ軸に制限します。
- Gを押して移動可能にし、マウスをZ方向に動かしてから MMB を押します。


Axis locking types

(軸ロックの種類)

Axis locking

(軸ロック)

モード: オブジェクトおよび編集モード(移動、回転、拡大縮小)






ホットキー: X、Y、Zもしくは希望の方向にマウスを動かしてから MMB 

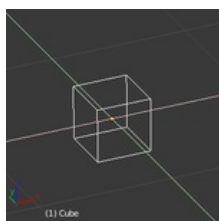
軸ロックはトランスフォームを一つの軸に制限します(言い換えれば2つの軸のトランスフォームを禁止します)オブジェクト、面、頂点、その他の選択可能な要素は1つの方向にだけ移動、回転、拡大縮小できます。

Plane locking

(面ロック)

モード: オブジェクトおよび編集モード(移動、回転、拡大縮小)

ホットキー:  ShiftX、 ShiftY、 ShiftZもしくは希望の方向にマウスを動かしてから  Shift MMB 



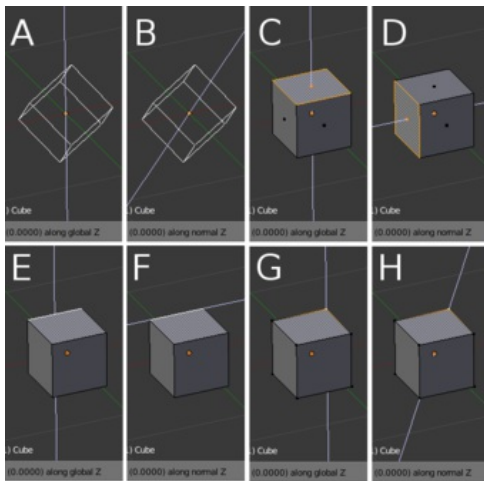
面ロック

面ロックはトランスフォームを2つの軸に制限します(言い換えれば1つの軸方向のトランスフォームを禁止します)。したがって平面を作れば、この要素は自由に移動や拡大縮小ができます。面ロックは移動と拡大縮小のみに影響します。

回転について注意すべきことは、回転は常に1つの軸に制限されるものなので、軸ロックと面ロックのどちらの効果も同じであることです。トラックボールタイプの回転 RR は一切ロックできません。

Axis locking modes

(軸ロックのモード)



軸ロックのモード

キーを1つ押すと対応するグローバル軸に移動を制限します。同じキーを続けて押した場合、現在の座標系に移動を制限します(グローバルにセットされる場合を除きます。この場合ローカル座標系が使われます)。最後に、3度目のキーを押すと、制限を取り除きます。

例えば、現在の座標系が法線であるときに G を押すと移動を開始し、続けて Z を押すと移動をグローバル座標系の Z 方向に制限し、再び Z を押すと移動を法線座標系の Z 軸に制限します。さらに Z を押すとすべての制限を取り除きます。現在のモードは 3D ウィンドウのヘッダーの左側に表示されます。

画像 [軸ロックのモード](#) をご覧ください。トランスフォームの方向は選択要素にも関係しています。A と B はそれぞれ、グローバル座標系と法線座標系で Z 軸がロックされています。C と D は同じ状況で面を選択している場合です。E と F は辺を、G と H は頂点を選択している場合です。

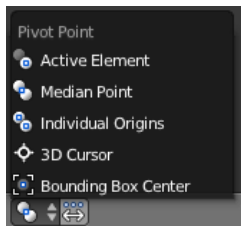
なお、軸ロックをしても、キーボードを使って [数値によるトランスフォーム](#) の値を入力できます。

ピボット(Pivot Point)

ピボットの選択

モード: オブジェクトモードと 編集モード

メニュー: 3Dビューのヘッダーにあるドロップダウンリスト



ピボットのモード

ピボットは空間内の一点で、その点を基準にすべての回転、拡大縮小、ミラー変形が行われます。3D領域のヘッダーにあるドロップダウンリストから、5種類のピボットを選べます(右図)。ピボットの各モードの詳細については下記ページをご覧ください。

- [アクティブ要素 \(en\)](#)
- [中点 \(en\)](#)
- [それぞれの原点 \(en\)](#)
- [3Dカーソル \(en\)](#)
- [バウンディングボックスの中心 \(en\)](#)


メッシュだけでなく他のタイプ(曲線、サーフェス...)にも同じルールが用いられることに注意してください。

Active Element as Pivot

(「アクティブな要素」ピボット)

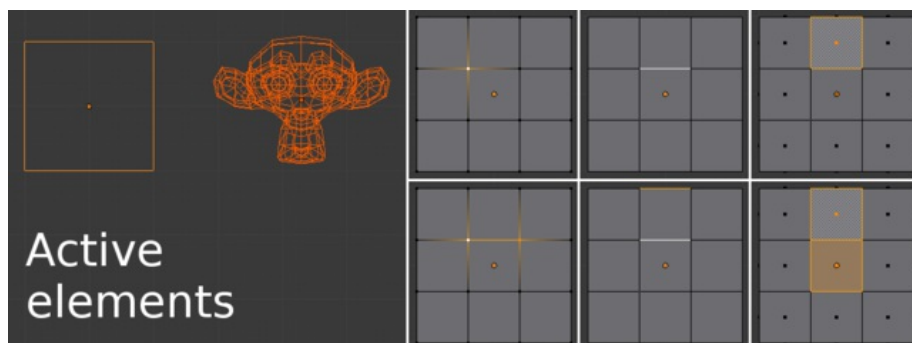
モード: オブジェクトモードおよび編集モード

ホットキー: Alt.

メニュー: 3D ウィンドウヘッダーで次のアイコンを選びます。window header 

オブジェクト、頂点、辺、面は **アクティブな要素**になることができます。アクティブな要素は最後に選択された要素で、オブジェクトモードでは薄いオレンジ色、編集モードでは白色で描かれます。アクティブ要素をピボットにすると、すべてのトランスフォームがアクティブ要素に相対的に行われるようになります。

[ピボットの切り替えについてもっと詳しく»](#)

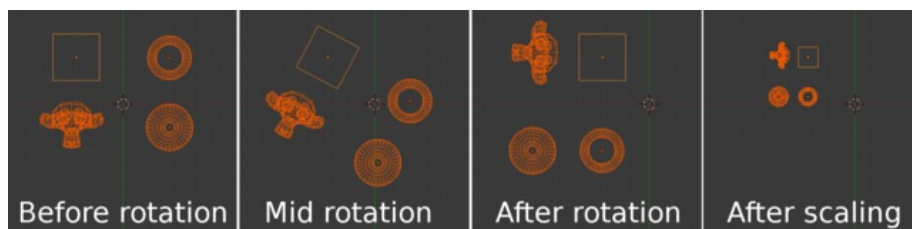


画像の左側はオブジェクトモードでのアクティブ要素の表示で、アクティブ要素である立方体は薄いオレンジ色です。画像の右側に見えるように、編集モードでの頂点、辺、面のアクティブ要素は白色で表示されています。

In Object mode

(オブジェクトモードでは)

オブジェクトモードでは、回転と拡大縮小はアクティブオブジェクトの中心を中心にして行われます。下図で示されるように、アクティブオブジェクト(立方体)は、他のオブジェクトがアクティブ要素に関して回転や拡大縮小を行う間も、同じ場所に留まります(この位置は 3D カーソルに相対的であることに注意してください)。



立方体をアクティブ要素とした回転と拡大縮小

In Edit mode

(編集モードでは)

編集モードでアクティブ要素をピボットにすると複雑に見えるかもしれませんが、発生するトランスフォームはすべて、次のルールに従います:

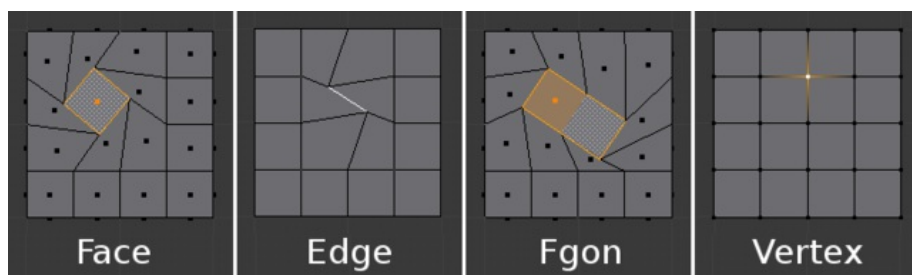
- ピボットは常にアクティブ要素(複数の場合もあります)の中心です。
- トランスフォームは選択要素の **頂点** のトランスフォームによって発生します。非選択の要素が選択中の要素と1つ以上の頂点を共有していれば、非選択のものもある程度トランスフォームします。

次の例で調べてみましょう: どのケースでもこの2つのルールが当てはまるのがわかります。

Single selection

(1つだけ選択)

1つだけ要素を選ぶと、その要素が自動的にアクティブになります。下の図では、トランスフォームが行われるとその頂点が移動することがわかります。その結果、アクティブ要素と1つ以上の頂点を共有している隣接要素もすべて、トランスフォームされています。



編集モードで1つだけ要素を選択

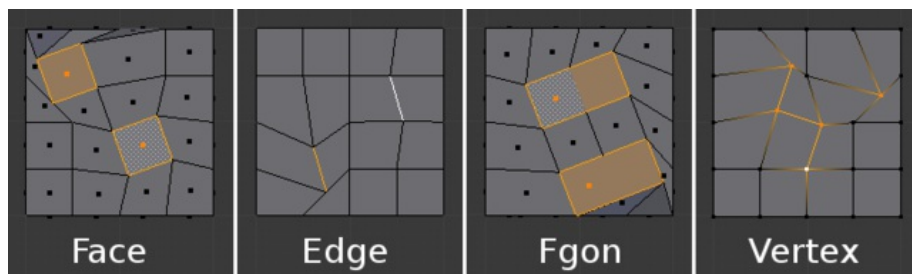
各例を検証してみましょう:

- *Faces* (面)ではピボットが面の選択ドットのある位置にあります。これは面の構成頂点の中点です。
- *Edges* (辺)ではピボットが中央にあります。これは辺の中点はいつもそこにあるからです。
- *Fgons* (Fゴン)の挙動は面と同じです。
- 単独の *Vertex* (頂点)は大きさをまったく持たないのでトランスフォームできません(ピボットに影響されない移動操作を除きます)。

Multiple selection

(複数の選択)

複数要素が選ばれているとすべてトランスフォームします。上の例で見たように Fゴン以外では、ピボットは同じ場所に留まります。下の画像では、選択要素が回転されています。



編集モードで複数の選択

- *Faces*(面)では、アクティブな面の選択ドットの周囲でトランスフォームが起きています。
- *Edges*(辺)では、ピボットを中点にしたときと同じ挙動になっています。
- *Fgons* (Fゴン)は面と同じ挙動です。
- 今回は *Vertex*(頂点)のケースがあります: ピボットのある位置にアクティブな頂点があります。他のすべての頂点はその頂点に相対的にトランスフォームされます。

Individual Origins as Pivot

(「それぞれの原点」ピボット)

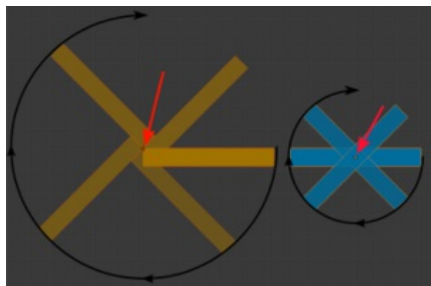
モード: オブジェクトモードおよび編集モード

ホットキー: Ctrl.

メニュー: 3D ウィンドウヘッダーで次のアイコンを選びます。 

In Object mode

(オブジェクトモードでは)



それぞれの原点を中心にした回転

オブジェクトの原点は 3D ビューにオレンジ色の小さな円で表示されています。右の画像では原点が赤い矢印で強調されています。原点は *オブジェクトが 3D 空間のどこにいるかを Blender に伝えます。あなたが 3D ビュー(頂点、辺など)で見た通りにオブジェクトは作り出されます。*

原点は *オブジェクト* の中心と同じですが、*メッシュ* の中心に置く必要はありません。つまり、オブジェクトはその中心をメッシュの端に置いたり、メッシュから完全に外側に置いたりできるということです。例えば、画像内のオレンジ色の矩形は原点がメッシュのかなり左にあります。

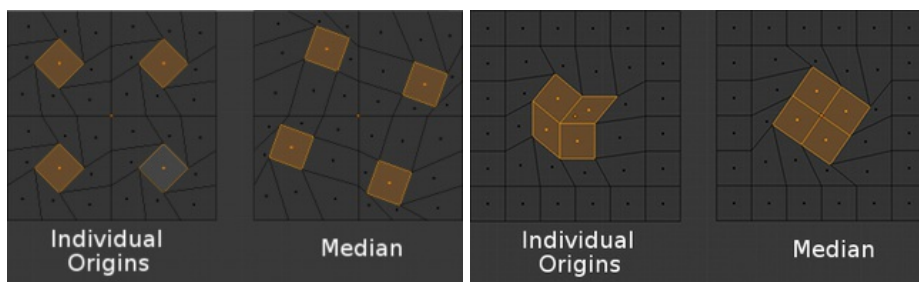
それでは、右の画像「*それぞれの原点を中心にした回転*」を調べてみましょう。

- 青い矩形は原点がメッシュの中央にあり、一方オレンジの矩形は原点が左側にあります。
- ピボットをそれぞれの原点にすると、各オブジェクトの中心(赤い矢印)はオブジェクトが黒い矢印で示された経路上を回転する間、同じ場所に留まります。

In Edit mode

(編集モードでは)

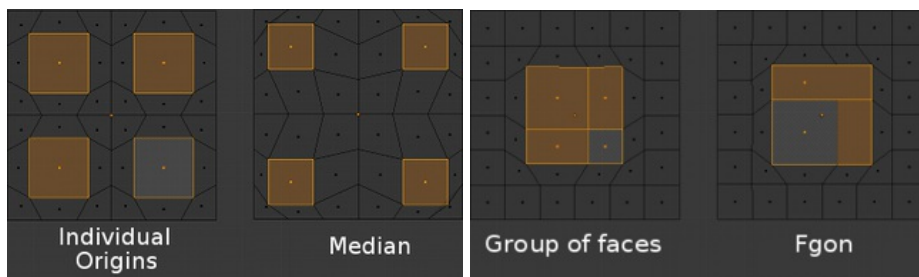
編集モードでは、ピボットをそれぞれの原点にすると、選択モードに頂点、辺、面のどれが選ばれているかによって違った結果になります。例えば、頂点モードではピボットを中点にしたのと似た結果になり、辺モードではしばしば歪んだ結果になります。「それぞれの原点」を面モードで使うと、結果は最も予測しやすいものになるでしょう。



別のピボットを使ったそれぞれの面の回転。左: それぞれの原点、右: 中点

別のピボットを使った面のグループの回転。左: それぞれの原点、右: 中点

上の図でわかるように、ピボットをそれぞれの原点にして回転すると接触しあっている面は変形します。接触していない面はそれぞれの原点(中心)の周りを回転します。



非接触面の拡大縮小。左: それぞれの原点、右: 中点

接触している面の拡大縮小。左: 面のグループ、右: Fゴン

面とFゴンのグループは外側の境界線を変形させずに拡大縮小できます。しかし、内側にある面は均一に拡大縮小されません。



ピボットをそれぞれの原点にした面によるモデリング

一旦その制限と落とし穴に気づけば、このツールは時間を大幅に節約でき、ユニークな形状を作れるようになるでしょう。この「アネモネ」は 12面柱から次の手順で 10 分ほどで作ったものです：各面の押し出し、ピボットとして中点を使った拡大縮小、ピボットとしてそれぞれの原点を使った面の拡大縮小と回転

3D Cursor as Pivot

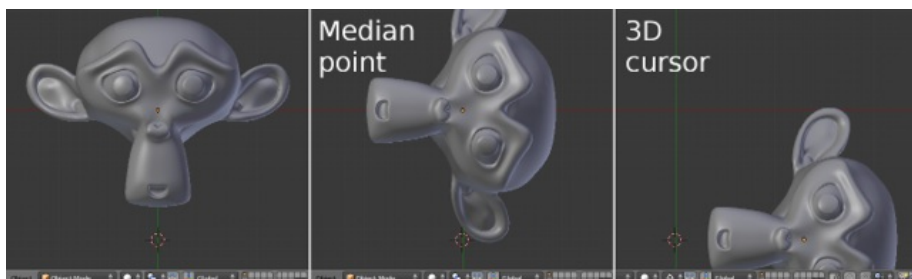
(「3D カーソル」のピボット)

モード: オブジェクトモードおよび編集モード

ホットキー: .

3D カーソルは一番直感的なピボットです。3D カーソルをアクティブなピボットとして選んだら(ウィンドウヘッダー または . ホットキーを使います)、3D カーソルの位置を決めて、必要なトランスフォームを実行するだけです。回転と拡大縮小のトランスフォームはすべて、3D カーソルの位置に相対的に行われるようになります。下の画像はオブジェクトの開始位置(左の枠)、中点をピボットにした 90 度回転(中央の枠)、3Dカーソルをピボットにした 90 度回転(右の枠)を示しています。

[ピボットの切り替えについてもっと詳しく»](#)



中点および3D カーソルを中心にした回転の比較

Positioning the 3D cursor

(3D カーソルの位置を決める)

3D カーソルの位置を決める方法はいくつかあります。

Direct placement with the mouse

(マウスで直接置き換える)

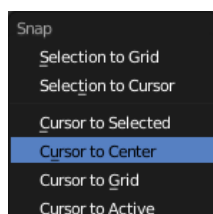


2つの直行するビューで3D カーソルの位置を決める

- 3D 領域で LMB (マウス左ボタン) を使うと、3D カーソルが直接マウスポインタの下に置かれます。精密さが必要なら、2つの直行する 3D ビューを使うべきです。すなわち上面図(7 NumPad)、正面図(1 NumPad)、側面図(3 NumPad)のいずれかの組み合わせです。この方法で1つのビューで2軸に沿って位置を調整し、もう1つのビューで奥行きを決めることができます。

Using the Snap menu

(スナップの利用)



スナップメニュー (⇧ ShiftS または Object(オブジェクト)/Mesh(メッシュ) » Snap(スナップ)) を使うとカーソルを次の方法でスナップさせることができます:

- Cursor to Selected (カーソル → 選択物): カーソルを現在選択中の頂点、辺、面にスナップします。オブジェクトモードではこのオプションは現在選択中のオブジェクトの中心にカーソルをスナップします。
- Cursor to Center (カーソル → 中心): カーソルをグリッドの原点(0,0)にスナップします。
- Cursor to Grid (カーソル → グリッド): カーソルを一番近い見えているグリッドにスナップします。
- Cursor to Active (カーソル → アクティブ): カーソルを アクティブな (最後に選択した) オブジェクト、辺、頂点にスナップします。

Cursor to Selected (カーソル → 選択物) オプションは、選択している要素の数や現在のピボットに影響されます。例えば、複数要素が選択されていてピボットが バウンディングボックスの中心 であるとき、カーソル → 選択物 オプションは 3D カーソルを次の位置にスナップします:

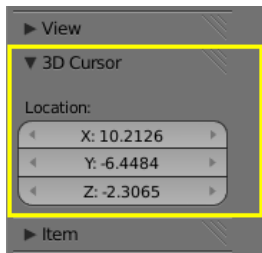
- オブジェクトモードなら選択オブジェクトすべての中心を囲むような **バウンディングボックスの中心**、編集モードなら選択している頂点を囲むような **バウンディングボックスの中心**

中点ピボットが選択されていると Cursor to Selected (カーソル → 選択物) オプションは 3D カーソルを次の位置にスナップします:

- オブジェクトモードなら選択オブジェクトのすべての中心の中点、編集モードなら選択頂点の中点

Numeric input

(数値入力)



プロパティシェルフの3D カーソルパネル


3D カーソルは、プロパティシェルフ (N) にある 3D cursor に数値で入力して位置を決めることもできます。

Median Point as Pivot

(「中点」のピボット)

モード: オブジェクトモードおよび編集モード

ホットキー: Ctrl,

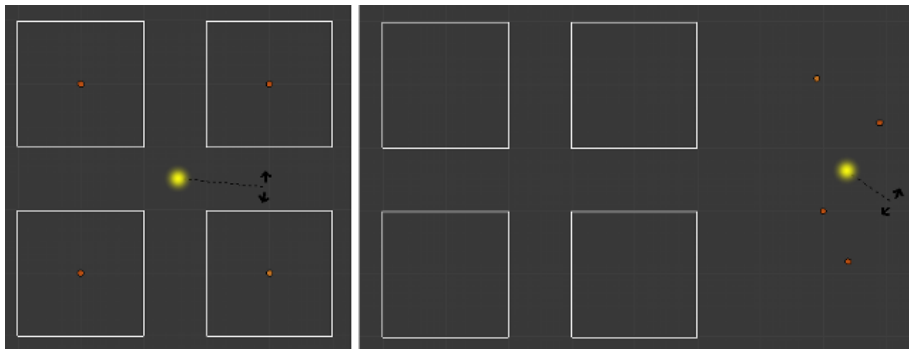
メニュー: 3Dウィンドウヘッダーにある次のアイコンを選びます。 

中点 (Median Point) は広い意味で重心の概念に似たものと考えられます。選択している要素 (オブジェクト、面、頂点など) にどれも同じ質量があると仮定すると、中点は選択要素の均衡点 (重心) に位置するでしょう。

In Object Mode

(オブジェクトモードモードでは)

オブジェクトモードでは、Blender は中点を決めるとき、オブジェクトの中心だけを考慮します。このため直感に反した結果になるかもしれません。下の画像「オブジェクトモードの中点」を見ると、中点はオブジェクトの中心にあることも、オブジェクトのメッシュとかけ離れた位置に置かれることもあるとわかります。

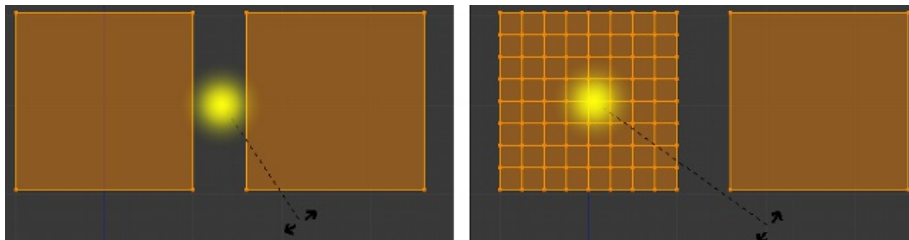


オブジェクトモードの中点。中点を黄色の点で示しています。

In Edit Mode

(編集モードでは)

編集モードでは、一番多く要素を含む選択部分によって中点が決まります。例えば、画像「編集モードでの中点」を見ると、同数の頂点を持つ2つの立方体があるとき、中点は2つの立方体の間にあります。ところが、片方の立方体を複数回細分化して頂点数を増やしてみると、もっとも頂点の多い領域に中点がずれていることがわかります。




編集モードでの中点。中点を黄色の点で示しています。

Bounding Box Center as Pivot

(「バウンディングボックスの中心」ピボット)

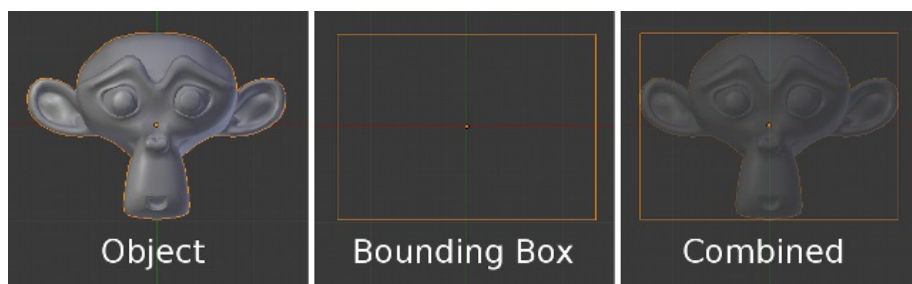
モード: オブジェクトモードおよび編集モード

ホットキー: ,

メニュー: 3D ウィンドウのヘッダーにある次のアイコンから選択 

バウンディングボックスは、選択要素の周りを可能な限りきつく包んだ四角い箱です。その向きはワールドの座標軸に平行です。このモードではバウンディングボックスの中心にピボットがあります。、ホットキーか ウィンドウヘッダー にあるメニューでピボットをバウンディングボックスに設定できます。下の画像はオブジェクトのバウンディングボックスのサイズが、オブジェクトのサイズからどのようにして決まるかを示しています。

[ピボットポイントの切り替えについてもっと詳しく»](#)



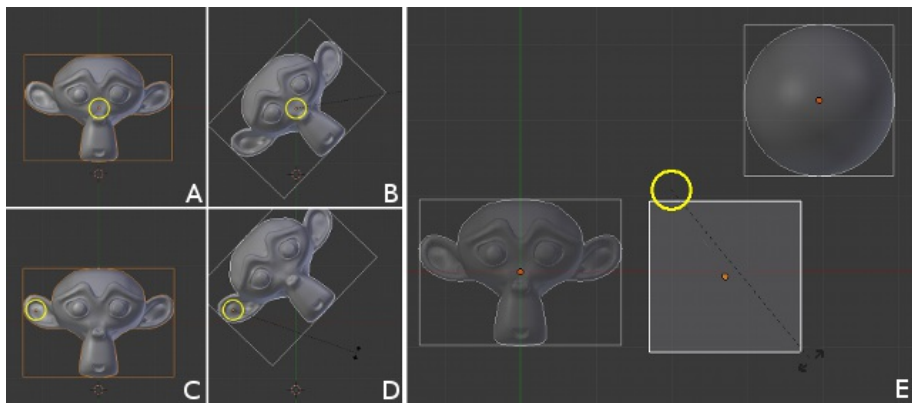
オブジェクトとそのバウンディングボックスの関係。左: オブジェクト、中央: バウンディングボックス、右: あわせもの

In Object mode

(オブジェクトモードでは)

オブジェクトモードでは、バウンディングボックスはオブジェクトを包み込み、トランスフォームはオブジェクトの中心(黄色の円)に相対的に行われます。下の画像は様々な状況でバウンディングボックスをピボットとして使った結果を示しています。

例えば、画像 A (回転の前)と B は、オブジェクトの中心がデフォルトの位置にあるときの回転を示しています。一方画像 C (回転の前)と D は、オブジェクトの中心が動かされていた場合の結果を示しています。画像 E は、複数のオブジェクトが選択されると、ピボットが選択オブジェクト全体のバウンディングボックスを基準に計算されることを示しています。

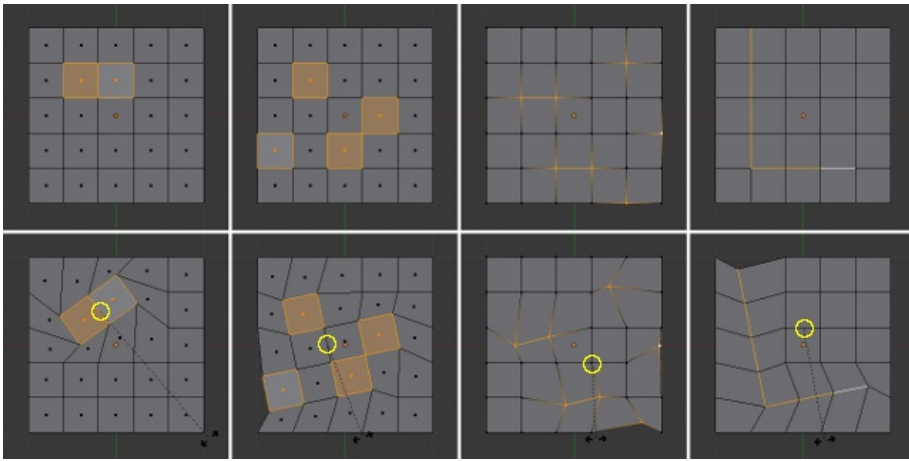


左側の4枚の画像 (ABCD) はピボットがバウンディングボックスに設定されているときのオブジェクトの回転結果を示しています。右側の画像 (E) は複数オブジェクトが選択されたときのバウンディングボックスピボットの位置を示しています。ピボットを黄色の円で示しています。

In Edit mode

(編集モードでは)

編集モードではバウンディングボックスに含まれるのは ObData です。編集モードのバウンディングボックスはオブジェクトの中心とは無関係で、選択頂点の中心とだけ関係します。



様々なメッシュ選択モードでピボットをバウンディングボックスに設定したときの回転の効果。ピボットは黄色の円で示されています。

スナップ

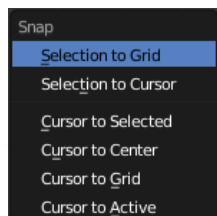
Blender では 2種類のスナップ操作を使えます。1つは選択しているものやカーソルを指定位置にスナップするもので、もう1つはトランスフォーム(移動、回転、拡大縮小)中に使われ、選択しているものをシーン内の要素にスナップするものです。

スナップ(Snap)

モード: オブジェクトモードおよび編集モード

ホットキー: ⇧ ShiftS

スナップメニュー(もしくは、オブジェクトモードでは 3Dヘッダーの Object » Snap、編集モードでは 3Dヘッダーの Mesh » Snap から)も選べます。このメニューには、カーソルや選択物を指定した位置(カーソル、選択物またはグリッド)に動かすためのオプションがあります:



スナップメニュー

選択物をグリッドに

現在の選択オブジェクトを最も近いグリッドポイントにスナップします。

選択物をカーソルに

現在の選択オブジェクトをカーソル位置にスナップします。

カーソルを選択物に

カーソルを選択オブジェクトの中心に移動します。

カーソルを中央に

カーソルをグリッドの中心に移動します。

カーソルをグリッドに

カーソルを最も近いグリッドポイントに移動します。

カーソルをアクティブオブジェクトに

カーソルをアクティブなオブジェクトの中心に移動します。

トランスフォームのスナップ

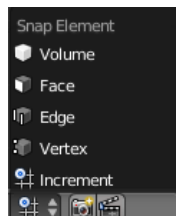
トランスフォーム中に、オブジェクトとメッシュ要素を様々なシーン要素にスナップする機能です。3Dビューのヘッダーボタンにある磁石のアイコンを押すと利用できます(赤色に変わります)。



3Dビューヘッダーの磁石アイコン(有効時に赤)

スナップモード

スナップ要素



スナップ要素メニュー

ボリューム(Volume)

マウスカーソルの下に最初に見つかったボリューム内の領域にスナップします。他のオプションと違って、奥行き(現在のビュー空間のZ座標)を制御します。ボリュームの中心を決めるときにオブジェクト群を全体とみなすかどうかを、スナップ対象メニュー(後述)の右にあるボタンで切り替えます。

面(Face)

メッシュオブジェクトにある面のサーフェスにスナップします。リトポ(retopologizing)に役立ちます。

辺(Edge)

メッシュオブジェクトの辺にスナップします。

頂点(Vertex)

メッシュオブジェクトの頂点にスナップします。

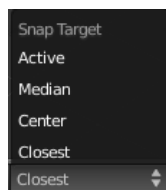
増加(Increment)

グリッド上の点(格子点)にスナップします。平行投影ビューではグリッドの刻み幅が拡大率によって変化します。ここで言うグリッドは、表示されて

いるグリッドと必ずしも一致しないことに注意してください。スナップはグリッドの刻み幅で行われますが、(スナップ操作前の)初期位置を基準にすべてのトランスフォームが行われます。

(訳注:この「ローカルな」グリッドは表示されません。オブジェクトを編集モードで移動/回転/拡大縮小すると、それに合わせて移動/回転/拡大縮小します。したがって、トランスフォーム後もメッシュの形状を基準にしたスナップができます。「表示されている」グリッドとスナップ位置が一致なくなります)。

スナップ対象



スナップ対象メニュー

スナップ対象オプションは、スナップ要素が頂点(Vertex)、辺(Edge)、面(Face)またはボリューム(Volume)のとき、利用可能になります。選択物のどの部分を対象オブジェクトにスナップするかを決めます。

アクティブ(Active)

アクティブ要素(編集モードでは頂点、オブジェクトモードではオブジェクト)を対象オブジェクトにスナップします。

中点(Median)

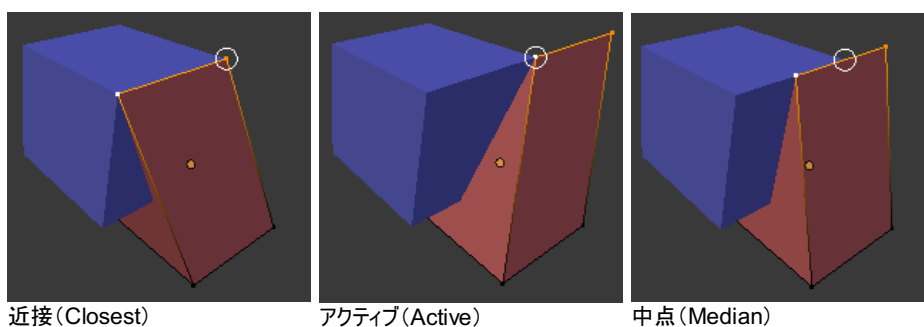
選択物の中点を対象オブジェクトにスナップします。

中心(Center)

現在のトランスフォームの中心を対象オブジェクトにスナップします。(訳注:オブジェクトが移動/回転/拡大縮小すると、3Dカーソルのスナップする仮想的な「グリッド」がそれにに応じて移動/回転/拡大縮小追従します)

近接(Closest)

選択物のうち、対象にもっとも近い点を対象にスナップします。



近接 (Closest)

アクティブ (Active)

中点 (Median)





さらなるスナップ操作



オブジェクトモード

編集モード

上の画像で赤くハイライトされた部分に見える通り、スナップ動作を変えるためのコントロールが利用できます。スナップ要素と同じで、これらのオプションはオブジェクトモードで編集モードで変化します。4つのオプションが利用可能です:

-  回転をスナップ対象にそろえる
-  オブジェクトのサーフェスに個々の要素を投影
-  要素をそれ自身のメッシュにスナップ
-  ボリュームの中心を見つけるときオブジェクトを全体とみなす

註釈

このページの英語記事は [Snap](#) と同じ内容です。同ページではメッシュ要素へのスナップについても説明されていますので、そちらをご覧ください。このページは今後元記事に変更が入れば更新します。

Proportional Edit

(プロポーショナル編集)

プロポーショナル編集は、選択要素(例えば頂点)のトランスフォームを、近くにある他の要素に影響させながら行う方法です。例えば一つの頂点を移動して、一定範囲内にある非選択の頂点をあわせて動かすことができます。非選択の頂点の移動量は選択頂点に近いものほど増えます(つまり選択要素との距離に比例して (proportionally) 動きます)。プロポーショナル編集は近くにあるジオメトリに影響するので、密集したメッシュの表面を滑らかに変形したいとき、とても役立ちます。

スカルプト

Blender には [スカルプトモード](#) もあります。頂点を個別に見ることなく、メッシュのプロポーショナル編集をするためのブラシやツールがあります。

Object mode

(オブジェクトモード)

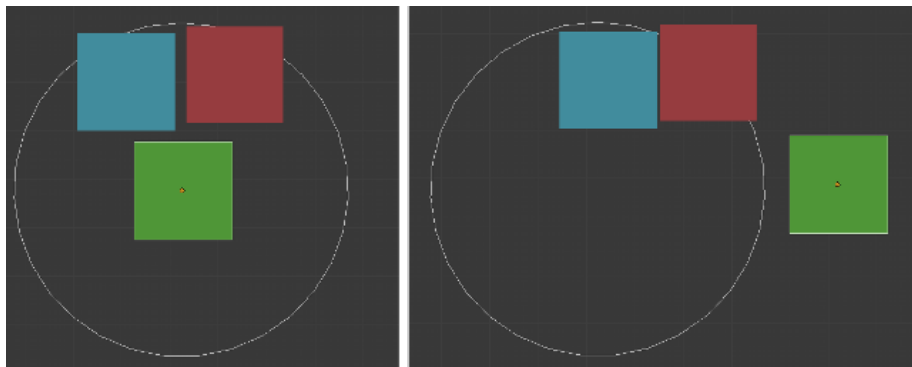
モード: オブジェクトモード

ホットキー: O

メニュー: ヘッダーの、次の画像で黄色で示されているアイコンから。



プロポーショナル編集は編集モードで使われることが多いですが、オブジェクトモードでも使えます。オブジェクトモードではツールは各メッシュ要素ではなく、オブジェクト全体に対して働きます。下の画像では、緑の立方体がアクティブオブジェクトで、赤と青の立方体はプロポーショナル編集ツールの影響範囲に置かれています。緑の立方体が右に動かされると、他の2つも動きに追従します。



オブジェクトモードのプロポーショナル編集

Edit mode

(編集モード)

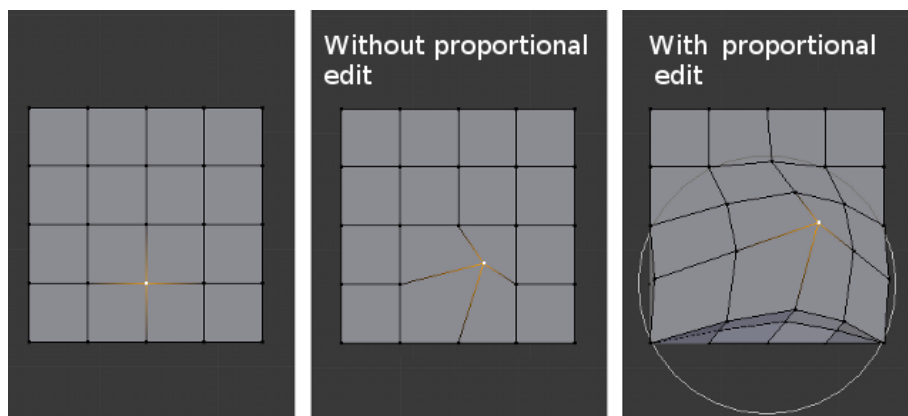
モード: 編集モード

ホットキー: O / AltO / ⇧ ShiftO

メニュー: Mesh(メッシュ) » Proportional Editing (プロポーショナル編集) または次の画像のハイライトされたアイコンから。





密集したジオメトリの作業では、頂点の細かな調整を、モデル表面のこぶやしわを目立たせずに行うことが難しくなります。こうした状況に直面したら、プロポーショナル編集ツールをモデル表面を滑らかに変形するのに使えます。一定範囲内にある非選択の頂点を、ツールが自動的に編集します。

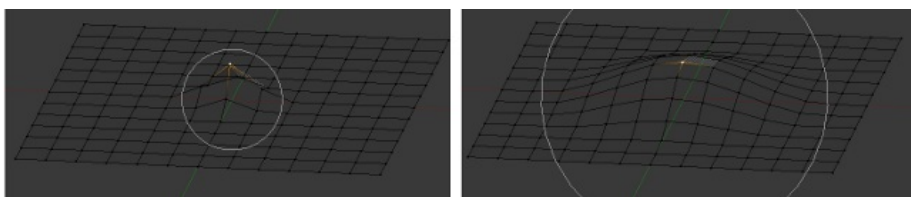


編集モードのプロポーショナル編集。中央: プロポーショナル編集なし、右: プロポーショナル編集あり

Influence

(影響範囲)

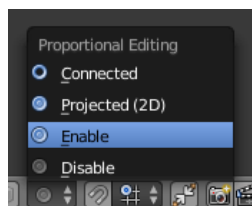
プロポーショナル編集の影響する範囲は円形で、半径をマウスホイール WheelUp /WheelDown か PageUp/PageDown で増減できます。範囲の増減に応じて選択要素を取り囲む点の位置が変わります。



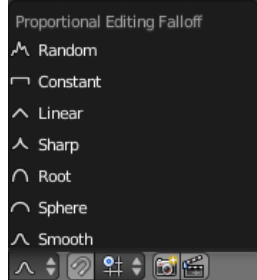
影響範囲

Options

(オプション)



プロポーショナル編集ツール



減衰メニュー

プロポーショナル編集モードのメニューは 3Dビューのヘッダーにあります。

無効(Disable) (O か AltO)

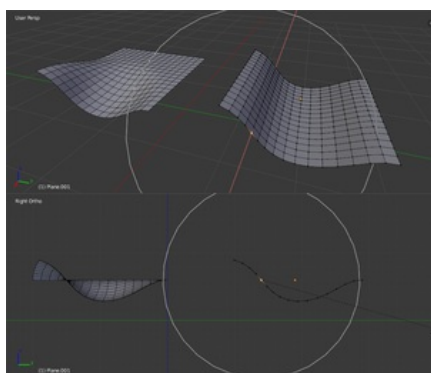
プロポーショナル編集は無効で、選択された頂点だけが影響を受けます。

有効(Enable) (O か AltO)

定義された範囲内にある、非選択の頂点だけが影響を受けます。

投影(2D)(Projected(2D))

範囲を決めるとき、ビュー方向の奥行きは無視されます。




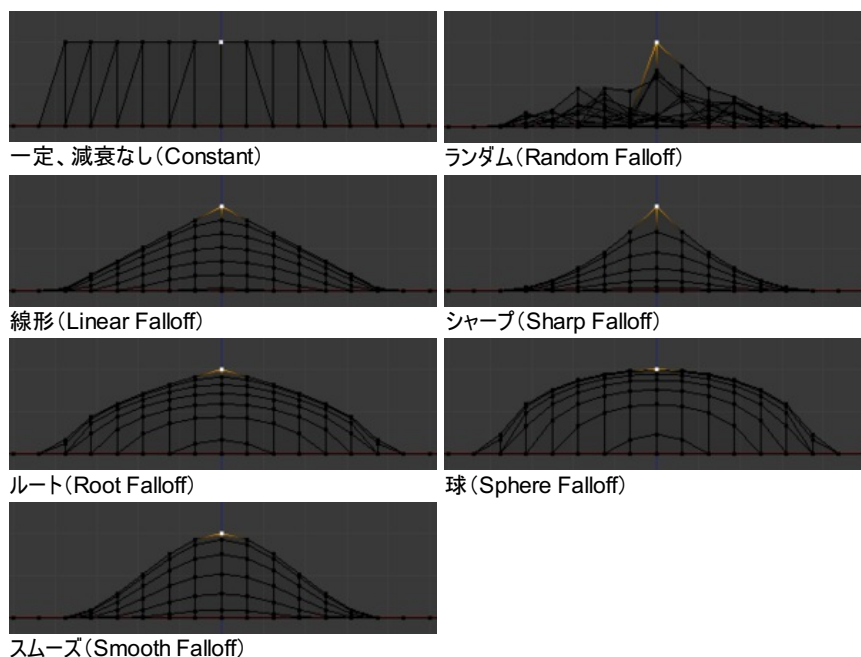
通常と Projected (2D) オプション(右)の違い

接続(Connected) (AltO)

範囲内にあるだけでなく、ジオメトリが繋がっているときにだけ影響が伝わります。つまり指の頂点のプロポーショナル編集を、他の指に影響を与えずに行えるということです。3D空間上の物理的な距離に近い頂点も、メッシュの辺の繋がりで見ると距離が離れていることがあります。接続がアクティブなときは、アイコンの中央に穴が空きます。このモードは編集モードでのみ利用できます。

減衰(Falloff)

側面の形状を Mesh(メッシュ) » Proportional Editing Falloff(プロポーショナル編集の影響減衰) サブメニューかツールバーのアイコン (減衰(Falloff)メニュー)を使うか、 ShiftO を押して多様なオプションを切り替えられます。



一定、減衰なし (Constant)

ランダム (Random Falloff)

線形 (Linear Falloff)

シャープ (Sharp Falloff)

ルート (Root Falloff)

球 (Sphere Falloff)

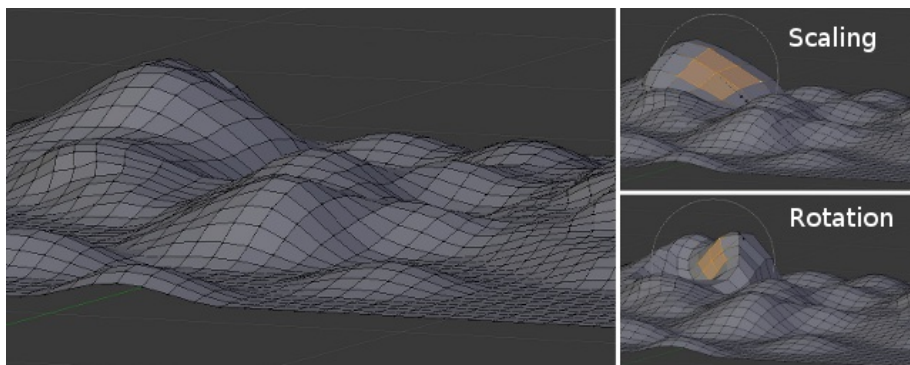
スムーズ (Smooth Falloff)

Examples

(例)

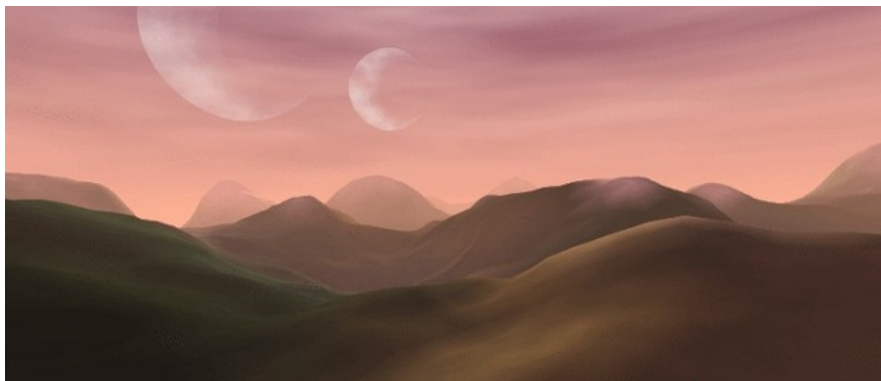
正面図に切り替えて (1 NumPad) 移動ツールを開始します G。上に頂点をドラッグしたとき、近くにある頂点の引き寄せられ方に注意してください。満足できる配置になれば、LMB を押して位置を固定します。満足していなければ、操作をキャンセルして RMB (または Esc)、メッシュを以前の見た目に戻してください。

プロポーション編集ツールは拡大縮小 (S) や回転 (R) で使うと、図「プロポーション編集で作られた地形」のような大きな効果を生み出せます。



プロポーション編集で作られた地形

このテクニックと頂点ペイントを組み合わせると、素敵な景観を作れます。下図「最終的なレンダリングがされた地形」ではテクスチャとライティングを適用後の、プロポーション編集の結果が示されています。



最終的なレンダリングがされた地形

はじめに

このページはバージョン 2.48 のグリース・ペンシル (Grease Pencil) に関する項目に基づいて編集されています *This page and its sub-pages are based on the [grease pencil release logs for blender 2.48](#).*

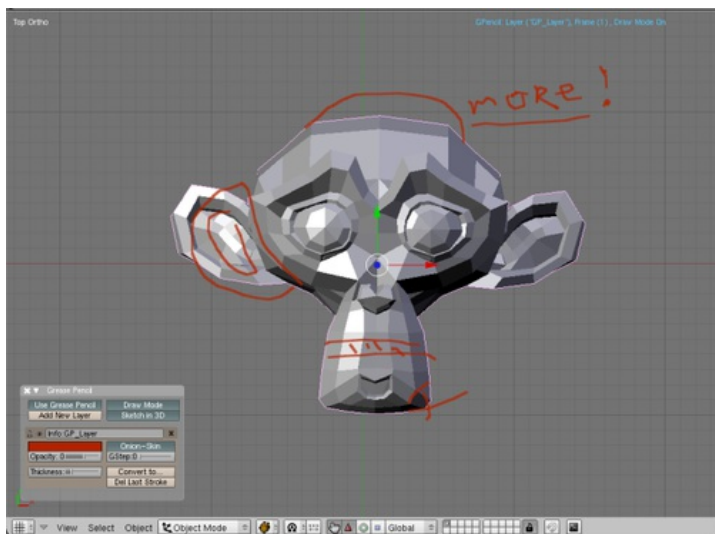
ビュー・ポート (Viewport) 上にフリーハンドによる線をスケッチや注釈として描画する機能があり、共同作業中のやりとりや計画立案に非常に有用です。これは従来からの、ラフな"ガイドライン"で計画やアイデアについて即座に話し合っていた、鉛筆と紙による平面上の作業を基にしています。製作中の作品に、別の場所を用意する事無く (つまり別のウィンドウや、あるいは同時に別のアプリケーションを起動する事無く)、直接的な描画を行える事はしばしば有用になります。

グリース・ペンシル (Grease Pencil) とは？

"Grease Pencil" という名称は、初期のCGアニメーターらが CRT ディスプレイに線を描いたり計画を立てる際に、植物の油脂を利用したクレヨンあるいは鉛筆を利用した事への敬意の表れに由来します。

アニメーターがポーズやモーショーカーブをについて考える場合だけでなく、グリース・ペンシルは以下に示す物を含んだ多くの状況でも有用になります:

- 物体の形あるいはモデルのレイアウトを考える場合
- ディレクターが動画のショットを評価するツールとして利用する場合
- 教育をする人間が"ホワイトボード"や課題を評価するツールとして利用する場合

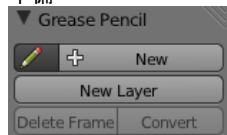


The Grease Pencil in action.

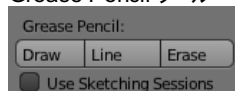
以下に示すいくつかのページではこのツールの使い方について述べられています:

- [スケッチを描く](#)
- [レイヤーとアニメーション](#)
- [スケッチから幾何形状への変換](#)

準備



Grease Pencil ツール



Grease Pencil 設定パネル

- グリース・ペンシル (Grease Pencil) を利用する際に最初に行うのは、グリース・ペンシルによるディスプレイ上の描画を有効にしておくことです。これを行うため、ツール・シェルフ (Tool shelf) (表示されていない場合は 3D View 上で T キーを押してください。) 内の Grease Pencil に移動して下さい。続いて プロパティ (properties) パネル (表示されていない場合は 3D View 上で N キーを押してください。) を開いてください。
- その次に、プロパティ パネル上のグリース・ペンシル設定パネル内に有る New Layer ボタンをクリックして描画するための新たなレイヤーを作成してください。この操作はグリース・ペンシルの描画を初めて行うときは、線の幅や色や透明度を描画の前に調整しない限り、不要です (自動で新しいレイヤーが生成されます。)。しかしながら、既に存在するレイヤーに描画したい場合はグリース・ペンシル設定パネル内のレイヤーのボタンをクリックしてください。

グリース・ペンシルは以下のウィンドウタイプでのみ有効です: 3D View および UV/Image Editor。

線を描く

Mode: 全てのモード

Panel: Object Tools » Grease Pencil » **Draw**

Hotkey: D LMB

説明

新たな線 (複数の短い線、つながっている線等) を描画をします。この線の描画はマウスのボタンを離すと終了します。

ヒント

D キーを押している間は、LMB を押して離すと点の描画ができます。

直線を描く

Mode: 全てのモード

Panel: Object Tools » Grease Pencil » **Line**

Hotkey: CtrlD LMB

説明

ゴムを引っ張るようにして直線を描きます。マウスのボタンを離すと直線の描画が終わります。

描画の消去

Mode: 全てのモード

Panel: Object Tools » Grease Pencil » **Erase**

Hotkey: D RMB

説明

描画の一部を設定された消去"ブラシ"の範囲内で消します。マウスのボタンを離すまで消去が続きます。

オプション

消去"ブラシ"の半径の大きさは User Preferences » Editing » Grease Pencil » Eraser Radius から設定できます。

ヒント



Erase を押してから操作を始めると、RMB あるいは LMB のいずれでも描画が消去されます。

Sketching Session について



Mode: 全てのモード

Panel: Object Tools » Grease Pencil » Use Sketching Sessions

説明

Sketching Session は必要なだけ Grease Pencil を迅速に利用できるようにします。このオプションを有効にすると、Grease Pencil の描画で Sketching Session が始まります。この状態では、LMB  が押される度に各レイヤーで定められた新しい線の描画を行えます。Sketching Session を終了するには Esc あるいは  Enter キーを利用します。

ヒント

LMB  あるいは RMB  のいずれでも描画の消去を行えます。

Grease Pencil の設定の共有に関して

描画の設定

Transform Panel » Grease Pencil » Drawing Settings パネルに移動すると、描画に関する4つの選択肢があります。これらの設定は一つのレイヤー内で混在させることができます。

- **View** - 描いた線が view 上に固定されます。これらの線は 3D 空間上に存在せず、スクリーン上に描画するように利用できます。
- **Cursor** (3D view でのみ設定可能) - 描いた描線が 3D カーソルの位置に基づいた 3D 空間上の場所に描画され、view とともに回転されます (言い換えると view の方向に向き 3D カーソルが交わる平面上に描画されます。)
- **Surface** (3D view でのみ設定可能) - 3D 空間上に新たな線を描画しますが、その位置は現在 "描画が行われている" view 上の視認できる最も手前に有るサーフェイス上になります (言い換えるとサーフェイス上に描画されます。)
- **Stroke** (3D view でのみ設定可能) - 3D 空間上に新たな線を描画しますが、その位置は現在 "描画が行われている" view 上の既に存在する描画の位置に描画されます (言い換えると既に存在する描画上に描画します。)。View 上に作られた描線は 3D 空間上に存在せずその位置は問題にならない事に注意してください。

上記の Surface あるいは Stroke の設定を利用している場合、その"上"に描画するサーフェイスや既に存在する描画上に新しく描画されなかった場合、Grease Pencil の描画は保存されません。もし新しく描画した場合はすべての描画が保存されます。

Transform Panel » Grease Pencil » Drawing Settings » Only Endpoints が有効な場合は終点でのみ描画の設定を切り替えられます。どこに描画されるかは終点がどの平面上に置かれるかで決まります。

制限

UV/Image Editor においても Cursor の設定は可能ですが、描画されると View の設定と同様な振る舞いをします。

描画する際の感度について

マウスやタブレットを利用して描画する際の感度の標準の設定は、細部の描画時にあまり変化しないように設定されています。しかしながら、時にはこの設定が適切でない場合があり、この場合は、User Preferences window » Editing » Grease Pencil のおける標準の設定を見直します。

- Manhattan Distance: 線を記録するためにカーソルを水平方向あるいは鉛直方向に移動させなければならない最小のピクセル数です。この数値を小さくすると曲線をより描きやすくなります。
- Euclidean Distance: 線を記録するために必要とされるカーソルの移動距離です。最後に記憶された位置とマウスカーソルの間の糸を想像してください。
- Eraser Radius: 表現通り Grease Pencil を消去するブラシのサイズであり、この値を変更すると消去するブラシに覆われる範囲が変わります。
- Smooth Stroke: 描線のギザギザ感を取り除く後処理の強さの段階を決定します。まっすぐでない線を描いた時のみ影響が現れます。標準状態ではオンに設定されています。この機能は描画を"縮める"作用がある事に留意しておき、その結果に問題がある場合はこの設定をオフにしてください。

タブレットのユーザーのために

- ある点における描線の太さはその部分を描いているときの筆圧の影響を受けます。
- 一端が"消しゴム"となるスタイラスは描線の消去に利用できます。

Drawing Planes

2.5x 系からの追加機能

この節を実際に編集する際には以下のビデオが有効です。

[\[video link\]](#)

レイヤー

Grease Pencil は GIMP や Photoshop で見られるようなレイヤー内に描画されます。これらのレイヤーは Blender のシステムのレイヤーとは無関係であり、利用できる数の上限が存在しません。上述のアプリケーションのようにこれらのレイヤーはリネーム、ロック、非表示そして削除が可能です。

レイヤーの主な目的はスケッチを意味の有るまとまりとして互いに集める事です ("割り当てノート"、"ディレクターの割り当てに対するコメント"等)。このような理由から、(何らかの変更を施した後に限らず) 1つのレイヤー上の描画はそのレイヤーの設定した色、透明度、線の太さの影響を受けます。

標準では、ほとんどの操作はアクティブなレイヤーに対してのみ影響を与えます。このアクティブなレイヤーはパネル上で異なる色 (標準では、明るめの濃い橙色) によって表示されます。1つのレイヤーをクリックする、もしくはその設定を変更するとそのレイヤーが新たなアクティブなレイヤーになります。

アクティブなレイヤーは (Grease Pencil が表示されている全ての View の右上側の角に有る) ステータス・インジケーターによっても確認できます。

スケッチのアニメーション

Grease Pencil は基本的なテストを行う事ができます (2Dのバラバラ漫画アニメーションのような物)。スケッチは描画されたフレーム上に保持され、(存在するレイヤー上に) 独立した描画として扱われます。各々の描画は次の描画がレイヤー上に現れるまで見える状態にあります。その例外は1つのレイヤーで最初に描画された物だけで、これはフレームに描画されていない時点でも見える状態にあります。

そのため、ペンシルテストやスケッチのアニメーションを作ることは簡単です:

1. 関係するフレームの最初のフレームに移動し、スケッチを描画してください。
2. 関係するフレームの次のフレームに移動し、再び描画してください。
3. 必要なだけ上記の工程を繰り返します。

Onion Skinning

Onion-skinning (もしくは ghosting) とは、Blender 上で前後のフレームを薄く表示する、アニメーターにとって便利なツールです。この機能によってアニメーターは異なるフレームと動きの比較を行って、動作について判断しやすくなります。

利用時の注意点:

- Onion-skinning は Grease Pencil のプロパティパネル上のレイヤー毎に Onion Skin ボタンを押して有効になります。
- Onion Skin ボタンのすぐ下に有る Frames フィールドでは、描画されるフレーム数を調整できます。Frames の値が0の場合は、現在表示されているフレームの前後1つずつだけのフレームが描画されます。それ以外の場合は、現在表示されているフレームの前後両側について、このフィールドで指定した値のフレーム数までのフレームが描画されます。

(NdT: 訳者によるメモ: 以下の内容が 2.5x の一部に適合しない説明のため和訳保留)

スケッチのタイミングの調整

Action Editor 上に表示される Grease Pencil ブロックで描画のタイミングを調整する事ができます。この機能はアニメーターがショットを組み立ていくのに特に有用で、その組み立てを行う事ができるようになる事が全体の目標になります。

1. In an Action Editor window, change the mode selector (found beside the menus) to Grease Pencil (by default, it should be set to Action Editor).
2. At this point, the Action Editor should now display a few "channels" with some "keyframes" on them. These "channels" are the layers, and the "keyframes" are the frames at which the layer has a sketch defined. They can be manipulated like any other data in the Action Editor can be.

All the available Grease-Pencil blocks for the current screen layout will be shown. The Area/Grease-Pencil datablocks are drawn as green channels, and are named with relevant info from the views. They are also labeled with the area (i.e. window) index (which is currently not shown anywhere else though).

スケッチのコピー

It is possible to copy sketches from a layer/layers to other layers in the Action Editor, using the "Copy"/"Paste" buttons in the header. This works in a similar way as the copy/paste tools for keyframes in the Action Editor.

Sketches can also be copied from one screen (or view) to another using these tools. It is important to keep in mind that keyframes will only be pasted into selected layers, so layers will need to be created for the destination areas too.

スケッチを他の形式に変換

3D ビューで、アクティブなレイヤー上のスケッチは、その時のビューの設定に基づいて幾何形状に変換可能です。スケッチはそれを (現在のビューの設定に基づいた) 3D 空間上に描画した時 (描線を描ききった時) に記録された点の情報を幾何情報に変換することができます。現状では、スケッチ上の全ての点を利用されるため、生成された幾何形状を利用に応じて部分的に単純化あるいは細分化する必要があるかもしれません。

現在スケッチは2通りのタイプへの変換が可能で、グリース・ペンシルのプロパティ・パネル内の Convert... ボタンを押したときに出現する Grease Pencil Convert メニューで利用できます:

- **パス (Path)**
- **ベジエ曲線 (Bezier Curve)**

描線はそれぞれ分割され、アクティブなレイヤー名に由来した名前を持つ曲線 (Curve) オブジェクトとして変換されます。曲線のハンドル (Handle) は自動的に "フリー (free)" ハンドル (黒いタイプ) に設定され、曲線のコントロール・ポイント (Control-point) と同じ場所に配置されます。曲線の各コントロール・ポイントにおけるウェイト (Weight) や半径は記録された各点での描線の太さに等しくなるように設定されます。しかしながら、それを把握するためには、エディット・モード時にの トランスフォーム・プロパティ (Transform Properties) パネルで Weight フィールドを参照する必要があります。

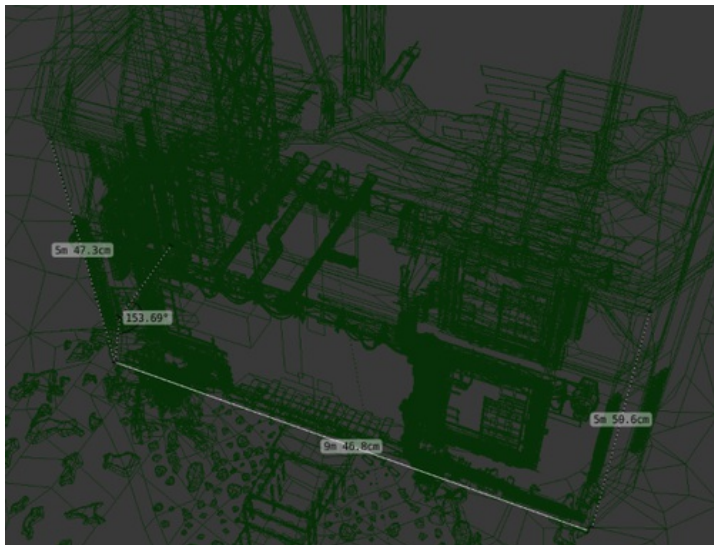
メッシュへの変換

もしスケッチをメッシュへと変換したい場合は、最初に Active Layer to Bezier を選択し、続いて生成された曲線をメッシュへと変換するだけです。

Ruler and Protractor

(3D定規と分度器)

ツールシェルフから使えます。実行すると、シーン内の長さや角度の計測ができます。



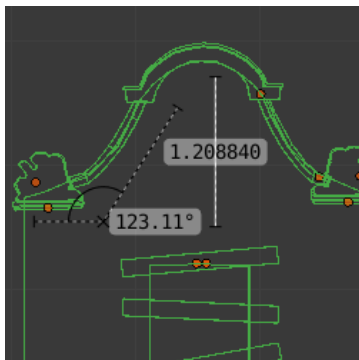
3D定規と分度器の例



3D定規で厚さを測る例

Usage

(使い方)



以下は3D定規の一般的な利用手順です




- ツールシェルフから 3D定規 (Ruler) を実行します
- ビューポートをクリック・ドラッグして 3D定規の最初の始点/終点を決めます。
- ビューを回して定規のいずれかの端をクリックし、再配置します。Ctrl を押し続けると要素へのスナップが有効になります
- 中央をクリックして角度を測ります。
- Enter を押してあとで使うために保管するか、Esc を押してキャンセルします

註記

- 3D定規の実行中も編集操作を行えますが、ナイフのようなツールとは同時に使えません。
- 単位の設定とシーンの拡大率が寸法の表示に使われます。

Shortcuts

(ショートカット)

- Ctrl LMB  3D定規を追加
- 端点を LMB  ドラッグして配置変更、Ctl キーを押し続けてスナップ、Shift キーを押し続けて厚さを計測
- 中心を LMB  ドラッグして角度を計測、ビューの外までドラッグすると3D定規に戻ります。
- Delete 3D定規を削除
- CtrlC 3D定規の値をクリップボードにコピー
- Esc 終了
- Return 3D定規を保存して次回実行時に引き継ぐ

Blender におけるモデリング

[Quick Start](#) でご覧になったように、3D シーンの作成に最低限必要な物は以下の3つになります:モデル、マテリアル、ライトです。この章ではこの内最初にあげたモデリングについて理解を一層深めます。モデリングは現実世界の物あるいは想像された抽象的な物に似た形の表面を作る科学と芸術です。

Object は多種多様な形、姿や大きさを持ち、そのため Blender は迅速かつ効率的にモデルを製作する助けとなるツールを持っています:

[Objects](#)

Object 全般の取り扱い方

[Meshes](#)

Object の形を定義するメッシュの取り扱い方

[Curves](#)

モデルに Curve を利用して Object を操作する方法

[Surfaces](#)

NURBS 曲面の利用

[Text](#)

3D 空間上に文字を表示するテクスチャのツール

[Meta Objects](#)

Glob と Globule について

[Duplications](#)

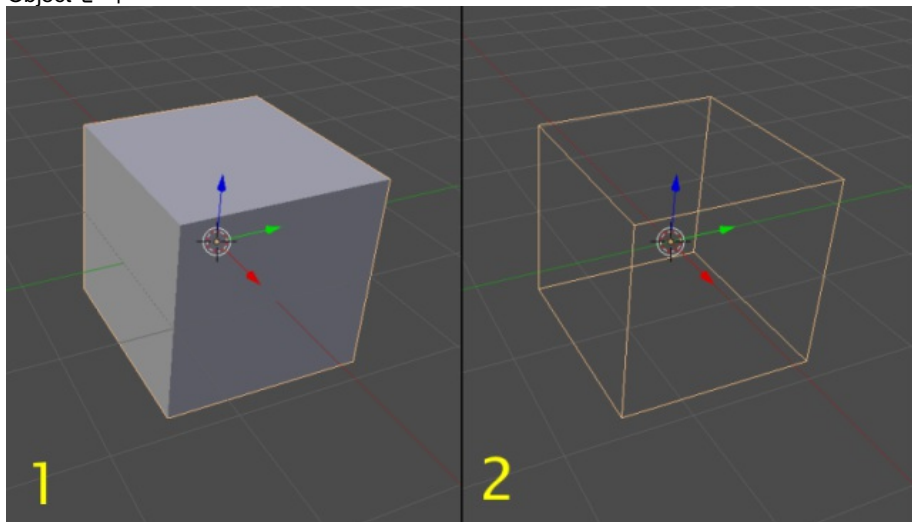
Object の複製について

[Modeling Scripts](#)

Blender の Python による拡張機能を利用した、モデリングの際に有用となる多数のスクリプトについて

多くの人々は1つの単純な立方体から"ボックス・モデリング"を利用し、より大きく複雑なメッシュを作るために、Face の押し出しや頂点の移動を行います。壁やテーブルの天板のように平坦な Object には、ベジエ曲線や NURBS 曲線によって外形を定めて必要な厚みに押し出す"カーブ・モデリング"を利用できます。いずれの方法もモデリング・ツールを利用する Blender で完全にサポートされています。

Object モード



Selected object.

1つのシーンの構造は1つ以上の Object によって構成されます: 例えばランプ、Curve、サーフェイス、カメラ、メッシュ等 "[Mesh Primitives](#)" で説明されている基本的な Object です。

それぞれの Object は Object モードで移動、回転、拡大縮小が可能です。



他の方法で Object の構造を変えるには [Edit モード](#) を利用してください。



基本的な Object をシーンに追加する ("[Mesh Primitives](#)" を参照してください) と、Object モードが表示されているはずですが、Blender の初期のバージョンでは、ここで追加される Object が Mesh、Curve あるいは Surface だった場合は自動的に [Edit モード](#) へと画面が遷移していました。

⇔ Tab キーを押すことで Object モードと Edit モードとを切り替えることができます。

Object のワイヤーフレームが、場合によっては橙色に見えており、それはその Object が *Selected object* の図で見られるように現在選択されてアクティブな状態にある事を意味します。

Selected object の図では標準状態で表示される立方体の Solid ビューと Wireframe ビューを示しています。これら Solid ビューと Wireframe ビューとの切り替えは Z キーを押すことで可能になります。

消去

モード: Edit あるいは Object モード

ホットキー: X あるいは Del

メニュー: Object → Delete

選択された Object の消去を行います。

結合

モード: Object モード

ホットキー: CtrlJ

メニュー: Object → Join Objects

種類が同じ Object を結合し、センター・ポイントがアクティブな Object から決まる1つの Object にします。結合の実行は Edit モード 時のオブジェクトの追加と同じ結果になります。

Select Links

モード: Object モード

ホットキー: ⇧ ShiftL

メニュー: Select → Select Linked

アクティブな Object と1つのリンクを共有する全ての Object を選択します。 *Selecting links* の図で見られるようにデータ、マテリアル、テクスチャ、パーティクル・システムあるいはライブラリへのリンクを共有している複数の Object を選択できます。

Select Linked	
Object Data	Shift L
Material	Shift L
Texture	Shift L
Dupligrpoup	Shift L
Particle System	Shift L
Library	Shift L
Library (Object Data)	Shift L

Selecting links.

- Object Data: *datablock* 情報 (同じ Mesh、同じ Curve、同じマテリアル他) を共有している複数の Object を選択します。
- Material: マテリアルの情報を共有している複数の Object を選択します。
- Texture: テクスチャの情報を共有している複数の Object を選択します。
- Dupligrpoup: Dupligrpoup の情報を共有している複数の Object を選択します。
- Particle System: パーティクル・システムの情報を共有している複数の Object を選択します。
- Library: ライブラリの情報を共有している複数の Object を選択します。 (Needs clarification)
- Library (Object Data): (Need details).

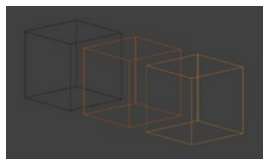
はじめに

多くのプログラムにおいて、選択とは操作の対象となる要素がどれになるかを定める事です。そのため、選択ツールの改善がより進めば意図する操作を行いやすくなります。Blenderにおけるツール類と機能はとても多様であり、それは選択方法についても同様です。

以下に示すのは概要と Object モード において利用可能な選択ツールについてです。


選択とアクティブな Object

Blender は選択において2つの異なる状態を区別します:




未選択の Object は暗く、選択された Object はオレンジ、アクティブな Object は黄色


- Object モード では最後に選択 (選択解除)されたアイテムを "アクティブな Object" と呼び、外形が黄色で (その他はオレンジで) 表されます。アクティブな Object は (たとえ何も選択されていない状態であっても)ただ1つだけいつでも存在します。

Blender における多くの操作は、例えばリンク操作のように、アクティブな Object を参照します。あなたが既に Object を選択していて現在と違う Object をアクティブにしたい場合は、その Object を **⇧ Shift RMB**  によって選択します。


- アクティブでないそのたの Object は単なる選択された Object です。Object はいくらでも選択する事ができます。


ポイント選択


最も単純な Object の選択方法で、Object 上で **RMB**  を押すだけです。

選択に追加するためには、他の Object 上で **⇧ Shift RMB**  を押してください。

View 上で Object 同士が重なり合っていた場合、**Alt RMB**  によって選択可能な Object のリストが得られます。

この方法で選択された Object に追加するためには **⇧ Shift Alt RMB**  というショートカットを利用します。

既に選択された Object をアクティブ化するには、その Object 上で **⇧ Shift RMB**  をクリックしてください。

アクティブな Object を選択解除するには、**⇧ Shift RMB**  をもう一度押してください - つまりアクティブではないが選択された Object を選択解除するには2回押すことになります。


境界(矩形)選択


モード: Object モード

ホットキー: B

メニュー: Select → Border Select

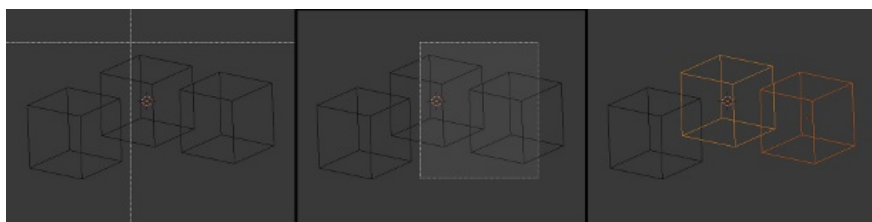
説明

境界選択では **LMB**  を押しながら長方形を描きます。部分的でも描かれた長方形の領域内に存在した全ての Object が選択されます。



Object を選択解除するには、**MMB**  を利用してください。

選択をキャンセルするには、**RMB**  を利用してください。

例



境界選択の段階

上図では、最も左の画像で境界選択が適用され交差する点線のカーソルの表示によってそれが分かります。続いて真ん中の画像では**選択範囲**が **LMB**  を押しながら長方形を描くことで選択されています。最も右の画像では選択は **LMB**  ボタンを離すことで完了します。

最も右の画像において、左側のより明るい色の立方体に注目してください。これが"アクティブな Object" であり、境界選択ツールを利用する直前に選


択されていた Object です。

ヒント

境界選択は直前までの選択に Object を追加するので、長方形で囲まれた Object のみを選択する場合は、最初に A キーを押して全ての選択を解除してください。

投げ縄選択

モード: Object モード

ホットキー: Ctrl LMB 


メニュー: メニューには項目無し

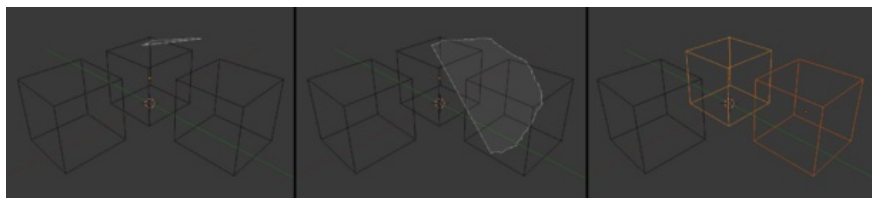
説明

投げ縄選択は Object モードにおいて Object のピボット・ポイントの周りを点線で囲んで利用します。

使い方

Ctrl キーを押している間、ピボット・ポイントの周りで選択したい Object を LMB を押しながら囲みます。

投げ縄選択は直前の選択範囲に追加されます。投げ縄選択の範囲を選択解除をするためには、Ctrl+Shift LMB を利用してください。



投げ縄選択

メニューによる選択

ここまで示された方法は一般的な選択手法です。3D ビューの Select メニューにはさらに多くの利用可能なオプションが存在します。

それぞれの小節ではより場合ごとの操作に関連した説明がなされます。

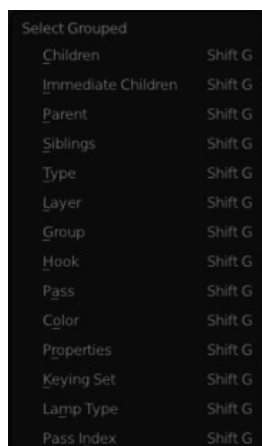
グループ選択

モード: Object モード

ホットキー: ⇧ ShiftG

メニュー: Select → Grouped

説明



Grouped selection menu

1つの Object をその他1つに関連付けるには2通りの方法があります。最初の方法は親子関係にする事、もうひとつが単なるグループ化です。これらの関係を、それぞれの親子関係やグループ毎の要素を選択する上で利用できます。

オプション

Object モードの Select → Grouped メニューではアクティブな Object を他の Object を選択する際の基礎として利用します。

利用可能なオプションは:

Children

アクティブな Object の全ての子を再帰的に選択します。

Immediate Children

アクティブな Object の直接的な子を全て選択します。

Parent

アクティブな Object の親が存在すれば選択します。

Siblings

アクティブな Object の親が共通の Object を選択します。これは (親となる Object が存在しない時に) ルート・レベルの Object を全て選択する際にも利用できます。

Type

アクティブな Object とタイプが同じ Object を選択します。

Layer

アクティブな Object と少なくとも1つのレイヤーを共有している Object を選択します。

Group

アクティブな Object が含まれるグループ内の Object が選択されます。

Object Hooks

アクティブな Object に対してフックしている Object 全てを選択します。

Pass

Render passe を共有している Object が選択されます。Render passe は Properties → Object → Relations で設定でき、Node Compositor (Add → Converter → ID Mask) で利用されます。

Color

共通の Object Color を持つ Object が選択されます。Object Color は Properties → Object → Display → Object Color で設定できます。

Properties

共通した Game Engine の Properties を持つ Object が選択されます。

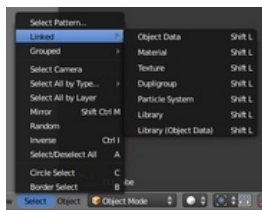
リンク選択

モード: Object モード

ホットキー: ⇧ ShiftL

メニュー: Select → Linked

説明



リンク 選択メニュー

アクティブな Object と1つのデータブロックを共有している Object が全て選択されます。

オプション

Object モードの Select → Linked メニューではアクティブな Object を他の Object を選択する際の基礎として利用します。

利用可能なオプションは:

Object Date

同じ Object データにリンクされている全ての Object を選択します。データ・ブロックとは Object のタイプ (メッシュ、Curve等) と構造 (頂点、空間上では調整された頂点等) を特徴付ける物です。

Material

共通したマテリアルのデータ・ブロックにリンクされた全ての Object を選択します。

Texture

共通したテクスチャのデータ・ブロックにリンクされた全ての Object を選択します。

Dupligroup

共通した Duplication のグループを利用している全ての Object を選択します。

Particle System

共通したパーティクル・システムを利用した全ての Object を選択します。

Library

共通したライブラリを持つ全ての Object を選択します。

Library (Object Data)

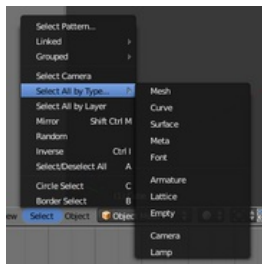
タイプによる選択

モード: Object モード

ホットキー: 無し

メニュー: Select → Select All by Type

説明



タイプによる 選択メニュー

ここに有るタイプは Mesh、Curve、Surface、Meta、Armature、Lattice、Text、Empty、Camera、Lamp です。

このツールによって特定のタイプの見える Object を選択する事ができます。

オプション

Select メニューの Select All by Type メニューは ObData データブロックに記述された全てのタイプの選択肢を提供します。

ご自由に選択ください。

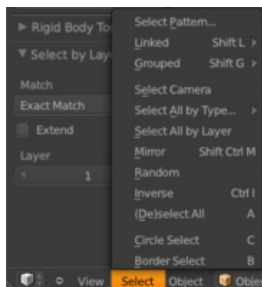
レイヤーによる選択

モード: Object モード

ホットキー: 無し

メニュー: Select → Select All by Layer

説明



レイヤーによる選択メニュー

レイヤーは Object 群を分けするという目的に適している別の方法です。

この選択方法は可視不可視に関わらず、1つのコマンドでレイヤー毎に属する Object の選択を可能にします。

オプション

Tool Shelf において以下のオプションが適用可能です:

Extend

この設定を有効にすると、現在の選択を置き換えるのではなく、Object を追加できます。

Layer

Object が有るレイヤーを示します。



Object の選択

Select All by Layer を利用するよりも、必要なレイヤーを可視状態にして A キーを押してそれらを全て選択するほうが効果的かもしれません。この方法は Object を選択解除する際にも使えます。

その他の選択方法

3D View のメニューの第1段階で利用可能な選択法は:

Random

現在アクティブなレイヤーで未選択の Object からパーセントで表される可能性に基づいてランダムに選択されます。このコマンドを選択した際は Tool Shelf 内の数値選択ボックスを利用できます。
注意すべき事はこの設定されるパーセンテージは未選択の Object から選択される可能性を示しているであり、選択される全ての Object の量に対する割合を示している訳ではない事です。

Inverse (CtrlI)

現在選択されていない全ての Object を選択し、選択されている Object は全て選択解除されます。

Select/Deselect All (A)

何らかの Object が選択されていた場合はそれらが選択解除されます。そうで無い場合は可視状態にあるすべての Object の全ての選択と選択解除を切り替えます。

Border Select (B)

既に表示した [境界\(矩形\)選択](#) をご参照ください。

Introduction


(序章)

この節は オブジェクトモード でのオブジェクト編集用のツールについて説明します。

いくつかの追加機能については [3D 操作](#) で説明されています。

Object Mode

(オブジェクトモード)

 Object Mode
オブジェクトモードボ
タン

デフォルトでは新しいファイルは オブジェクトモード が有効にされた状態で開きます。有効にするには 3D ビューウィンドウ→ ヘッダー にある オブジェクトモードボタン をクリックします (図「 オブジェクトモードボタン」をご覧ください)。

編集用のツールはすべて選択されたオブジェクトに対してのみ動作します。詳細は [オブジェクトの選択](#) をご覧ください。

以下で説明されるコマンドはすべて *Object* (オブジェクト) メニューか *Object tools* (オブジェクトツール) パネルにあります (下図参照)。



オブジェ
クトツ
ルパ
ネル


Creation and deletion

(作成と削除)

もっとも基礎的な編集にはオブジェクトの存在の操作があります。以下はさまざまな作成と削除のツールです。

Add

(追加)

ホットキー:  ShiftA

メニュー: Add (追加)

次のプリミティブから一つを追加できます:

- **Mesh** (メッシュ): Plane, Cube, Circle, UV Sphere, Icosphere, Cylinder, Cone, Grid, Monkey, Torus.
- **Curve** (カーブ): Bezier, Circle, NURBS Curve, NURBS Circle, Path.
- **Surface** (サーフェス): NURBS Curve, NURBS Circle, NURBS Surface, NURBS Cylinder, NURBS Sphere, NURBS Torus.
- **Metaball** (メタボール): Ball, Capsule, Plane, Ellipsoid, Cube.
- **Text** (テキスト).
- **Armature** (アーマチュア): Single Bone.
- **Lattice** (ラティス).
- **Empty** (エンプティ): Plane Axis, Arrows, Single Arrow, Circle, Cube, Sphere, Cone, Image.
- **Speaker** (スピーカー).
- **Camera** (カメラ).
- **Lamp** (ランプ): Point, Sun, Spot, Hemi, Area.
- **Force Field** (フォースフィールド): Force, Wind, Vortex, Magnetic, Harmonic, Charge, Lennard-Jones, Texture, Curve Guide, Boid, Turbulence, Drag, Smoke Flow.
- **Group Instance** (グループインスタンス): (ユーザー定義のオブジェクトグループ)。

Duplicate

(複製)

ホットキー: ⇧ ShiftD

メニュー: Object(オブジェクト) » Duplicate Objects(複製)

ホットキー: AltD

メニュー: Object(オブジェクト) » Duplicate Linked(リンク複製)

オブジェクトのコピーを作ります。リンク複製では一部属性がリンクされます。[複製](#)をご覧ください。

Join

(結合)

ホットキー: CtrlJ

メニュー: Object(オブジェクト) » Join(結合)

選択したすべてのオブジェクトから一つの単独のオブジェクトを作ります。オブジェクトは同じタイプでなければなりません。原点は直前に アクティブだったオブジェクトから取得されます。Join の実行は 編集モード で新しいオブジェクトを追加することに相当します。非アクティブなオブジェクトは削除され(このオブジェクトのメッシュはアクティブオブジェクトに取り込まれます)、アクティブオブジェクトだけが残ります。このツールはメッシュとカーブを含む編集可能なオブジェクトにだけ機能します。

Delete

(削除)

ホットキー: X, D or Delete, D

メニュー: Object(オブジェクト) » Delete(削除)... » Delete(削除)

選択オブジェクトを消去します。

Transformation tools

(トランスフォームツール)

オブジェクトはさまざまな方法でトランスフォーム可能です。以下はトランスフォームのさまざまな種類です。

Translate

(移動)

ホットキー: G

メニュー: Object(オブジェクト) » Transform(トランスフォーム) » Grab/Move(つかむ/移動)

オブジェクト位置の変更操作です。座標の中心からの相対的な、オブジェクトの原点の X、Y、Z 座標を変更します。

Rotate

(回転)

ホットキー: R

メニュー: Object(オブジェクト) » Transform(トランスフォーム) » Rotate(回転)

オブジェクトの角度の変更操作です。現在の座標系に相対的なオブジェクト座標系の X、Y、Z 軸周りの角度を変更します。各オブジェクトのどの部分も、同じオブジェクトの他の部分との相対位置は変わりません。

Scale

(拡大縮小/スケール)

ホットキー: S

メニュー: Object(オブジェクト) » Transform(トランスフォーム) » Scale(拡大縮小)

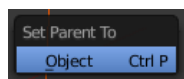
オブジェクトのサイズを変えます。オブジェクト座標系の X、Y、Z 軸に沿って、比例的に伸縮します。

グループ化と親子関係

一つのシーンの中には多数のオブジェクトを配置できます。標準的なステージのシーンは調度品、支持部材、光源、背景を持っています。Blender ではそのようなオブジェクトを互いにグループ化し、全て体系的に管理できます。

例えばあなたが、腕時計のような複雑なオブジェクトをモデリングする際、オブジェクトを異なる部品毎に分割して作成するかもしれません。しかし、これら全ての部品は互いに何らかの形で関係している場合があります。このような場合、その関係する部品全ての親としてオブジェクトを扱えます。親となったオブジェクトの移動、回転等は子であるオブジェクトにも影響を与えます。

親子関係の作成



Set Parent To ポップアップメニュー

親子関係を作成するためには、少なくとも2つのオブジェクトを(子となるオブジェクトを先に、親となるオブジェクトを最後に)選択し、Ctrl P を押しします。親子関係の作成に関する確認ダイアログがポップアップします。親子関係の作成から選択して確認すると一つ以上の子と親となるオブジェクト間の親子関係が作成されます。最後に選択されたオブジェクトはアクティブなオブジェクト(黄色で縁取られています)となり、これが親オブジェクトになります。親となるオブジェクトを選択する前に複数のオブジェクトを選択していた場合、それら全てが同じ階層の子オブジェクトとなります(これらの子オブジェクトは"兄弟"です)。

non-inverse モードを利用する場合は ◊ ShiftCtrlP というショートカットを上記の代わりに行います。このモードは子オブジェクトが、その原点を含め、完全に親オブジェクトに依存するという、通常とは異なった親子関係を作成します。これは例えば CAD のような利用を行う際に有用です。

親オブジェクトの移動及び回転は通常、子オブジェクトも同様に移動あるいは回転させます。しかし、子オブジェクトを移動あるいは回転させても、親オブジェクトには影響を与えません。違った言い方をすれば、影響は通常 descendant (親オブジェクト → 子オブジェクト群) であり、ascendant (子オブジェクト群 → 親オブジェクト) でないということです。

頂点の親子関係

1つのオブジェクトを1つの頂点もしくは3つの頂点からなるグループの子オブジェクトとして設定出来ます。この時親となるメッシュが変形させられた時、子オブジェクトが、脈打つ動脈上の止血鉗子のように移動します。Object モードで、子オブジェクトを選択し、続いて親オブジェクトを選択します。その状態で ⇨ Tab キーを押して Edit モードに入り、親オブジェクト上の1つの頂点、もしくは3つの点を(これらの3つの点は1つの面上に存在する必要はありません。親オブジェクト上の任意の3点を対象として)選択します。最後に CtrlP のショートカットを利用し確定します。

この時点で、もし1つの点が選択されていた場合、親子関係を示す点線は親となった頂点と子オブジェクトを結ぶように描画されています。もし3つの点が選択されていた場合、点線は親オブジェクトの3点の中心から子オブジェクトを結ぶように描画されます。この状態で親オブジェクトのメッシュを変形させたり、親となる頂点を移動させたりすると、子オブジェクトが同様に移動します。

Note

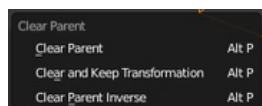
これは“あべこべの” [フック](#) のような物と考えられるでしょう。

オプション

子オブジェクトの移動

子オブジェクトの原点をクリアすることで親オブジェクトの場所へ子オブジェクトを移動させる事ができます。この操作を行っても、親オブジェクトと子オブジェクトの関係には影響を与えません。子オブジェクトを選択して AltO のショートカットを利用してください。ダイアログで確認を行うと子オブジェクトが親オブジェクトの場所へ移動します。Outliner ビューを利用し親子関係が維持されたままであることを確認してください。

親子関係の解消 / 親オブジェクトのクリア



Clear Parent ポップアップメニュー

AltP を利用することで、親子関係を削除できます。

各メニューの詳細:

Clear Parent

もし親オブジェクトが選択された状態であれば何も起きません。もし子オブジェクトが選択されていた場合は親オブジェクトとの関係が解消され、子オブジェクトの本来の位置、角度、大きさになります。

Clear and Keep Transformation

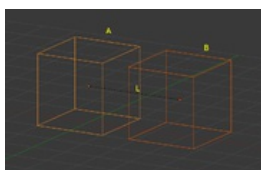
子オブジェクトを親オブジェクトから開放しますが、親オブジェクトの影響を受けた位置、角度、サイズは維持されます。

Clear Parent Inverse

親オブジェクトによって決まっていた位置をグローバルな座標で定義されている場所に移動させます。これは事実上親オブジェクトによる、子オブジェクトの座標の変換をクリアします。例えば、親オブジェクトが X 軸に沿って 10 ブレンダー・ユニット移動して Clear Parent Inverse を使用されたらとすると、選択された子オブジェクトは X 軸に沿って -10 ブレンダー・ユニット移動します。この"逆変換"は最後の移動に対してのみ有効で

す。もし親オブジェクトが2度移動したとし、それぞれ 10 ブレンダー・ユニットずつの移動で合計 20 ブレンダー・ユニットの移動があったとしても"逆変換"は直前1回の移動に対してのみ有効であり、子オブジェクトは 20 でなく 10 ブレンダー・ユニットだけ戻ります。

親子関係の例

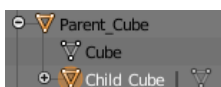


親子関係の例

アクティブなオブジェクト(立方体 A)は黄色で表示され、そのグループ内の他のオレンジで表されたオブジェクト群(立方体 B)の親オブジェクトになる事ができます。子オブジェクトの中心は親オブジェクトの中心と点線で繋がっています。親子関係の例の図を参照してください。この図の場合、親オブジェクトは立方体 "A"、子オブジェクトは立方体 "B" となります。それらのリンクは "L" と表示されています。

この状態では、親オブジェクトに対する移動、回転、サイズの変換は子オブジェクトに同様に作用します。親子関係は発展的な利用において非常に重要なツールとなっており、後の章(例としてアニメーションの応用)で参照する事になるでしょう。

ヒント



Outliner ビュー

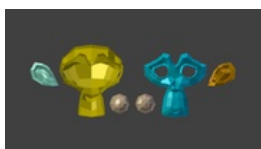
グループ内の親子関係を参照する別の方法として [Outliner ウィンドウ](#)内の Outliner ビューを利用する方法があります。画像 Outliner ビューは 親子関係の例において立方体 "A" を "Cube_Parent"、立方体 "B" を "Cube_Child" とした時の Outliner ビューの例です。

オブジェクトの分割

時には一つのメッシュを切ることで部品を切り離す必要が出てくる事が有ります。その操作は容易なものです。

オブジェクトを分割するためには、対象とする箇所の頂点(あるいは面)を選択した後に、方法が幾つか有りますが、分割します。Edit モードで P ショートカットを利用し以下のいずれかの方法を選択します。

オプション



綺麗に分割された Suzanne

Selected

選択された箇所を新たなオブジェクトとして分割します。

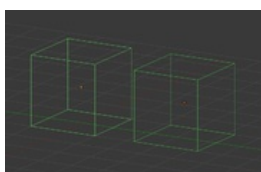
All Loose Parts

選択された範囲を持つオブジェクトに接続されていない部分を分割します。

By Material

マテリアル毎にオブジェクトのメッシュの分割を行います。

オブジェクトのグループ化



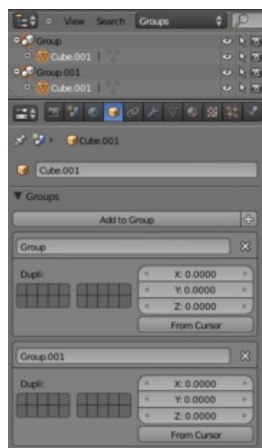
グループ化されたオブジェクト

移動・回転・拡大縮小のような変形の依存関係無しにオブジェクトを互いに結びつけます。シーン上で文字通り関連付けるだけ、あるいはファイルやシーンをまたいだオブジェクト間の調整や接続をワンステップで行いたい場合はグループを利用してください。1つのグループをなすオブジェクトは選択されると、グループ化されたオブジェクトの図で示されるように、ライトグリーンで表示されます。

オプション

Creating a Group

CtrlG 1つの新しいグループを作り、選択されたオブジェクトをそのグループに追加する。



グループの命名

Naming a Group

1つのオブジェクトが登録されたグループ全てが Object Properties パネル内の Relations パネルに一覧表示されています。グループの名前を変更するには、名前を変更したいグループの名前の領域をクリックしてください。

Outliner ウィンドウ内のグループの名前を変更したい場合は、ヘッダ内のコンボボックスから Groups を選択し、グループ名をクリックしてください。名前の欄が編集可能な領域に変化するので、希望の名前を入力して **↵ Enter** を押してください。

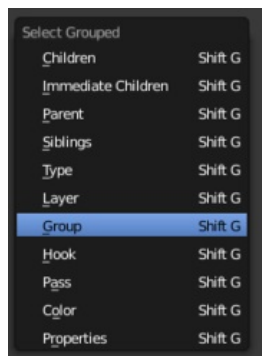
Restricting Group Contents via Layers

各グループ毎に関連付けられたレイヤーボタンの集まりは、グループオブジェクトが複製された時に、どのレイヤーのオブジェクトが複製物に含まれるかを決定します。もしあなたが作成したグループが、レイヤー 10, 11, および 12 上のオブジェクトを含んでいる時、グループコントロールのレイヤー 12 に対応したボタンを非選択状態に切り替えると、([Dupligroup](#) 機能を用いて) グループの複製を行った場合、複製したグループの内レイヤー 10 及び 11 上のオブジェクトだけが表示されます。

Appending or Linking Groups

1つの別の .blend ファイルをグループに追加します。詳しくは[こちらのページ](#)を参照してください。要点をおさえると、File → Append or Link → filename → Group → グループ名 を選択という操作を行なう事になります。

グループ化されたオブジェクトの選択



Selected Grouped ポップアップメニュー

☞ ShiftG のショートカットを利用すると親子、グループ関係の状態に基づいてオブジェクトを選択できるダイアログがポップアップします。

オプション

Children

現在アクティブなオブジェクトの子オブジェクト、そして子オブジェクトの子オブジェクトを再帰的に選択します。

Immediate Children

現在アクティブなオブジェクトの子オブジェクトを全て選択しますが、子オブジェクトの子オブジェクト以下(孫オブジェクト以下)は選択されません。

Parent

現在アクティブなオブジェクトの親オブジェクトを選択し、現在アクティブなオブジェクトを非選択状態にします。

Siblings

現在アクティブなオブジェクトの兄弟オブジェクトを全て選択します。

Type

現在アクティブなオブジェクトのタイプに基づいてオブジェクトを選択します。

Layer

アクティブなオブジェクトと同一のレイヤー上のオブジェクトを全て選択します。

Group

アクティブなオブジェクトと同一のグループ上のオブジェクトを全て選択します。

Hook

アクティブなオブジェクトに [hook](#) されているオブジェクトを全て選択します。

Pass

アクティブなオブジェクトと同じ PassIndex 値を持っているオブジェクトを全て選択します。このオプションの詳細に関しては [ID Mask node の利](#)

[用法](#)を参照してください。

Color

アクティブなオブジェクトと同じ色 (Draw パネルで設定します) を持つオブジェクトを全て選択します。

Properties

アクティブなオブジェクトと同じ [game properties](#) を持っているオブジェクトを選択します。

複製

Mode: Edit and Object modes

Hotkey: ⇧ ShiftD

Menu: Object → Duplicate

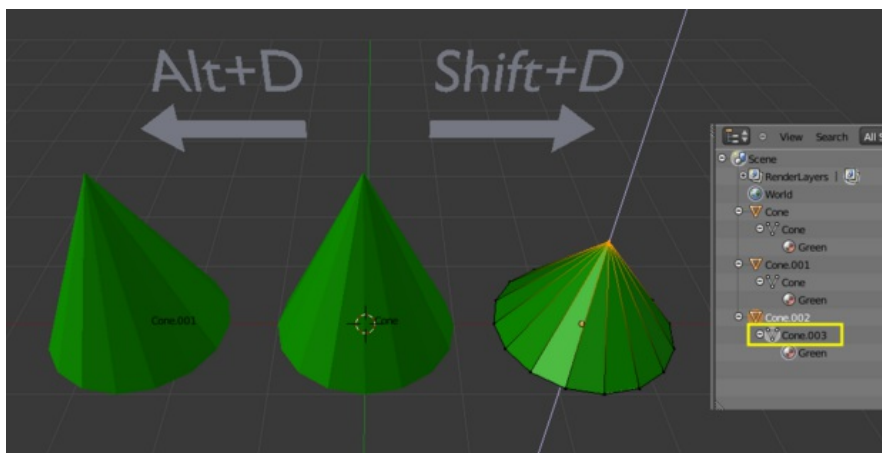
概要

これは、選択しているオブジェクト(複数可)と視覚的に同一なコピーを作成します。コピーは元のオブジェクトと同じ位置に作成され、あなたは自動的に Grabモードに入ります。以下の記述では (*Duplicate Example*) 図を参考に。

このコピーは新規のオブジェクトで、いくつかのデータブロックを元のオブジェクトと“共有”していますが(デフォルトでは、すべてのマテリアル、テクスチャ、lpos)、しかしそれ以外、例えばメッシュは、コピーになります。この方式の複製が時々“浅いリンク”と呼ばれる理由がこれで、すべてのデータブロックが共有されるわけではなく、その一部が“ハードコピー”されます!

複製する際にどのタイプのデータブロックをリンクまたはコピーするか選択できることを覚えておきましょう: User Preferencesの(File メニューで利用可能) Editing “tab”で、本当にDuplicate Dataリストにコピーしたいものだけを、そのタイプのデータブロックで有効にしましょう — 他のは単にリンクされます。

例



The mesh Cone.006 of object Cone.002 is being edited. The mesh's Unique datablock ID name is highlighted in the Outliner.

真ん中のコーンから、(1)リンク複製を左へ、(2)複製を右へ行なっています。

- 複製した右のコーンには編集を加えていても、中央のオリジナルコーンは変更されません。メッシュデータが、リンクではなくコピーされています。
- 同様に、右側のコーンがオブジェクトモードで編集されている場合も、オリジナルのコーンは変わりません。新しいオブジェクトの変形プロパティまたはデータブロックは、リンクではなくコピーされます。
- 右の円錐を複製すると、中央の円錐のマテリアルを継承します。マテリアルの特性は、コピーではなくリンクされます。

データブロックが標準的にリンクされた別個のコピーが必要な場合は、上記を参照してください。

リンク複製

Mode: Object mode

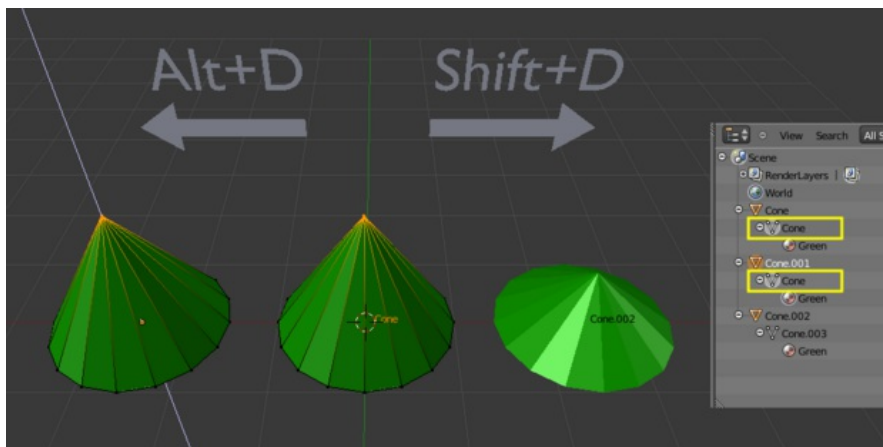
Hotkey: AltD

Menu: Object → Duplicate Linked

Description

また、複製のかわりにリンク複製を作成する選択肢もあります。これはディープリンクと呼ばれています。この方法で作成する新規オブジェクトは、元のオブジェクトにリンクされている **すべての** データを持ちます。リンクされたオブジェクトのいずれかを編集モードで変更した場合、リンクされているすべてのコピーが変更されます。変形プロパティ (object datablocks)はまだコピーのまま、リンクではないので、回転、拡大縮小、および移動は他のコピーに影響を与えることなく自由に行うことができます。以下の記述では (*Duplicate Example*) を参考に。

Examples



オブジェクト Cone.001はリンク複製されたものです。これらのコーンは両方とも、Coneという名前のついたメッシュとは別の一意の単一のオブジェクトですが、アウトラインで強調表示されているところは、両方で共有されています。

左のコーンは、中央のコーンをリンク複製したものです (AltDを使用)。

- いずれかのオブジェクトの頂点を 編集モード で移動させると、オリジナルのコーンの同じ頂点も同様に移動されます。メッシュデータはコピーではなく、リンクされます。
- これとは対照的に、これら2つの円錐形の一方をオブジェクトモードで回転または再スケーリングした場合、他方はそのまま変わりません。変形プロパティはコピーされ、リンクはされません。
- 前の例のように、新しく作成されたコーンは、元のコーンのマテリアルを継承しています。マテリアル特性はコピーではなく、リンクされます。

一般的なテーブルは、天板と4本の脚を持っています。脚を1本モデリングし、次にリンク複製を三回行って、残りの脚それぞれに使用します。後でメッシュに変更を加えると、すべての足を同じに揃えてくれます。リンク複製はコップのセット、車のホイール、にも適します... 繰り返しや対称性はあらゆるところにあります。

Procedural Duplication

Mode: Object mode and Edit mode

Panel: Object settings

オブジェクトを手続き的に複製する方法は、Blenderには現在4つあります。これらのオプションは、Objectメニューに配置されています。

[Verts](#)

これは、このオブジェクトの各頂点にあるすべての子のインスタンスを作成します (メッシュオブジェクトの場合のみ)。

[Faces](#)

これは、このオブジェクトのそれぞれの面のすべての子のインスタンスを作成します (メッシュオブジェクトの場合のみ)。

[Group](#)

これはグループのインスタンスを、オブジェクトの変形と一しょに作成します。グループ複製は、アクションを使用してアニメーション化することができ、また [Proxy](#) になってもらうこともできます。

[Frames](#)

アニメーションオブジェクトにおいて、これがすべてのフレーム上でのインスタンスを作成します。このトピックのサブページでわかるように、これは、オブジェクトを配置したり、それらをモデリングする用途でも非常に強力なテクニックです。

Linked Library Duplication

Hotkey: ⇧ ShiftF1

Menu: File → Link Append

[Linked Libraries](#)

Linked Libraries もまた、複製のいち形態です。他の.blendファイル内のあらゆるオブジェクトやデータブロックが、現在のファイルで再利用できます。

ヒント

- 変形プロパティ (つまり object のデータブロック) を “リンク”、させたい場合は、[親子化](#)のページを参照しましょう。
- マテリアルの透過度は、dupli-groups で実体化させると表示されません。これはblenderの view-port からくる制限だということがわかっています。

DupliVerts

Mode: Object mode

Panel: Object → Duplication

Duplication VerticesまたはDupliVertsは、メッシュの頂点の位置へのベースオブジェクトの複製です。言い換えると、メッシュ上DupliVerts使用すると、ベースオブジェクトのインスタンスが、メッシュの各頂点に配置されます。

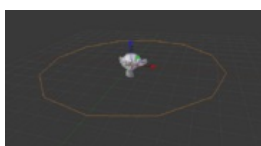
実際にはDupliVertsを使用したモデリングには2つのアプローチがあります。それらは配置ツールとして使用することができ、オブジェクトの幾何学的な配置をモデルリングできるようにしてくれます (例えば、ギリシャ神殿の柱、庭の木、ロボット兵士の軍隊、教室の机)。オブジェクトは、Blenderがサポートする任意のオブジェクト型にすることができます。第二のアプローチは、その一部分から始めて、それらをオブジェクトモデルに使用することです (例: 棍棒のスパイク、ウニのトゲ、壁のタイル、花の花弁)。

Download example .blend file

このページに掲載した例を収めたファイルをダウンロードすることができます。この [.blend](#) で、最初の例は、猿が円へと親子化されていて、レイヤー1の上にあります。一方、触手がicosphereへと親子化されているものはレイヤー2上にあります。ファイルは Blender 2.55.1 (r33567)を使用して製作されました。

配置ツールとしてのDupliVerts

Setup



A monkey head and a circle

必要なのは、ベースオブジェクト (例えば 木や 柱) と、あなたが意図するパターンに従った頂点を持つパターンメッシュです。このセクションでは、あるシンプルなシーンを以下のパートに使用していきます。座標系の原点に位置する猿の頭をベースオブジェクトとして使用し、そして同じ場所にある円を親メッシュとして使っていきます。



Duplivered monkeys

まず、Object modeで、ベースオブジェクトを選択して **⇧ Shift RMB** で円も追加選択し (順序はここで非常に重要です)、**CtrlP** でベースオブジェクトを円に親子化します。今、円が猿の親であり、円を移動した場合、猿は円についていきます。

円だけを選択した状態で、Object panel → Duplication → VertsでDuplication verticesを有効化します。猿の頭が円の各頂点に配置されているはずです。

中心のオリジナルの猿の頭と親メッシュはまだ3Dビューに表示されていますが、どちらもレンダリングはされません。猿の頭の位置や回転が変になった場合は、その回転 (**AltR**)、スケール **AltS**、場所 **AltG**、原点 **AltO** をクリアする必要があるかもしれません。

Rearranging

今ベースオブジェクトを選択し、オブジェクトモードと編集モードのいずれかでそれを変更した場合、すべての変更がすべての複製オブジェクトの形状に影響を与えます。また、親メッシュを選択して複製の配置を変更することもできます。頂点を追加すると、追加されるベースオブジェクトもより多くなります。ベースオブジェクトが継承するのは、オブジェクトモードで親メッシュに加えられた変更で、編集モードでの変更は継承しないことに注意してください — なのでオブジェクトモードで円をスケールアップすると、猿の頭も拡大され、一方編集モードで円をスケールアップすると、ベースオブジェクトの間の距離だけを増加させます。

Orientation



Orientation enabled, orientation +Y

Tベースオブジェクトの向きの制御は、DuplicationパネルでRotationを有効化することで行えます。こうすると、親メッシュの頂点の法線に従って、すべてのベースオブジェクトを回転させます。

複製したオブジェクトの向きを変更するには、ベースオブジェクトを選択し、Object → Relations extrasパネルでTracking Axesを変更します。

配置方向を様々に変えた出力



Negative Y

Positive X

Positive Z, up X

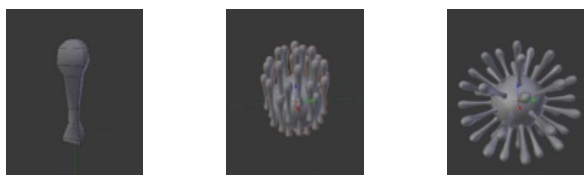
Note

オブジェクトの軸の可視化は、Object→Displayパネルで行うことができます。

親メッシュの頂点の法線を表示するには、タブで編集モードに入り、Properties (N)→Displayパネルでこの機能を有効化します。このパネルでは必要に応じて表示される法線のサイズを変更することもできます。

モデリングツールとしてのDupliVerts

DupliVertsと標準的な基本形状を使うと、非常に興味深いモデルを作ることができます。この例のシンプルな触手は、キューブを数回押し出すことによって製作しました。触手オブジェクトは、icosphereにペアレントされています。親メッシュ(the icosphere)で dupli Rotationを有効化すると、ベースオブジェクト(触手)の向きを親メッシュの頂点の法線に合わせます(この場合は触手を編集モードでX軸周りに-90°回転させた)。



A simple tentacle set to smooth Tentacle dupliverged onto the parent mesh Rotation enabled to align duplicates

前の例のように、配列の形状や比率はすぐに調整できます。

すべての複製を実際のオブジェクトに変えるには、シンプルにicosphereを選択してObject→Apply→Make Duplicates Real(Ctrl⇧ShiftA)と選びます。icosphereと触手をひとつのオブジェクトにするには、それらすべてが選択されていることを確認して、Object→Join (CtrlJ)と進みましょう。

関連項目

他の複製方法は [こちら](#)に一覧表示されています。

DupliFaces

Mode: Object mode

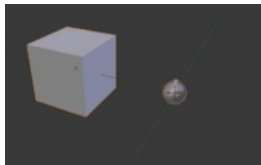
Panel: Object → Duplication

Duplication Faces または DupliFacesは、親オブジェクトの各面の上にオブジェクトを複製する機能です。これを説明するためにいちばん良い方法は、イラスト例を通じて解説することでしょう。




Example .blend file

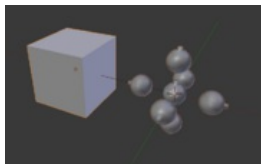
このページの例に使用された .blend ファイルのダウンロードは [こちら](#)

Basic usage



A cube and a sphere

この例では、"北の極"を押し出したUV球をベースオブジェクトとして使用し、そして立方体を親メッシュとして使用しています。立方体を球の親にするには、Object modeで、まず球をRMB  で選択し、続いて立方体も  Shift RMB  で選択し (順序はここで非常に重要です)、最後に CtrlP で親子化します。



Duplication Faces applied to the cube

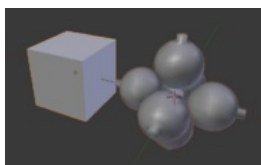
次に、ObjectコンテキストのDuplicationパネルで、Facesを有効化しましょう。球が立方体の各面に1つずつ複製されます。

Inherited properties

複製される子の位置、方向、スケールは、親の面のそれと一致します。なので、複数のオブジェクトがキューブに親子化されている場合、それらはすべてキューブの各面ごとに一度ずつ複製されます。キューブが細分化されている場合 (Edit ModeでW)、子はすべて、キューブの各面に複製されます。

親オブジェクトとオリジナルは両方とも、編集可能な"テンプレート"として3Dビューに表示されますが、どちらもレンダリングはされません。

Scale



Scale enabled



Top face of cube scaled down

親オブジェクトのScaleを有効化することによって、子オブジェクトのスケールが親オブジェクト内の各面のサイズに適用されます。

したがって、親オブジェクトの面を拡大縮小することにより、複製オブジェクトのサイズもそれに応じて変わります。

DupliGroup

Mode: Object mode

Panel: Object → Duplication → Group

Duplication Group または DupliGroup では、グループのインスタンスを、別のオブジェクトの各インスタンス用に作成することができます。

基本的な使用方法

- 多数のオブジェクトを作成し、それをグループ化するには
 1. それらをすべて選択して、
 2. CtrlG、そして
 3. 最後にグループの名前を Object → Groups で変更する
- DupliGroupを作成するには
 1. 別のオブジェクト、たとえばEmptyを追加 (⇧ ShiftA)して、
 2. Object → Duplication で Groupを有効化し、そして
 3. 新しく作成したグループの名前を、表示される選択ボックス内から選ぶ。

DupliGroup and Dynamic Linking

別の.blendファイルから現在のファイルにデータを動的にリンクさせる方法を理解するには[Appending and Linking](#)を参照してください。あるblendファイルから他方へ、グループを動的にリンクすることができます。そのようにすると、グループインスタンスの表示場所を制御するオブジェクトが作成されるまで、リンクされたグループはシーンのどこにも表示されません。

例

- [Appending and Linking](#)で説明したように、シーン内の別のファイルからグループをリンクさせましょう
ここからは、イージーな方法かハードな方法を使用することができます:
- 簡単な方法:
 1. Add → Group Instance → [name of group you just linked]と選択する。
- 難しい方法:
 1. Add → Emptyと選択し、追加したエンティティを選択状態にする。
 2. Objectコンテキストに切り替えて、Duplication/パネルで、Groupをクリックする。
 3. Group:の横に表示されるドロップダウンボックスで、リンクさせたグループを選びとる。

この時点で、グループのインスタンスが表示されます。エンティティを複製したり、各エンティティにDupliGroup設定を保持させたりできます。このやり方で、リンクされたデータの複数のコピーをとることが非常に簡単にできます。

Making a DupliGroup Object Real

さて、DupliGroupインスタンス上でさらに編集をくわえたり、DupliGroupのレンダリングをYafarayや他のいくつかのレンダラのような、DupliGroupsの直接インポートをサポートしていないレンダラでしなくなったら:

シンプルにDupliGroupを選択して Ctrl⇧ ShiftA を押すと、DupliGroupが通常のオブジェクトに変換されて、変形やアニメーション化が普通にできるようになります。

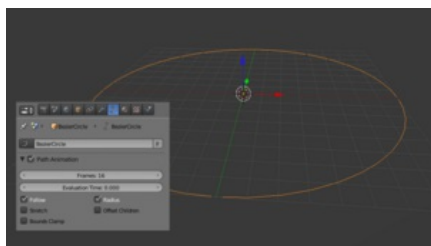
Note

DupliGroupが外部ファイルからリンクされている場合でも、オブジェクトデータ(メッシュ、マテリアル、テクスチャ、変形)もまだ元のグループからリンクされたままだということに注意してください。ただし、各種オブジェクトの親子関係は持ち越されません。

DupliFrames

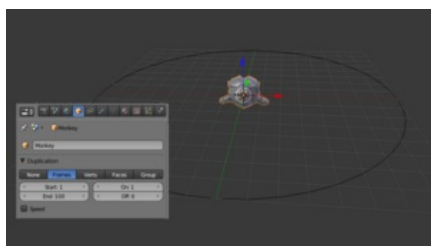
DupliFrames は、パスに沿って分布するフレーム位置にオブジェクトを複製するツールです。これは、オブジェクトすばやくを配置したいときに便利なツールです。

Examples



Settings for the curve

⇧ ShiftA で Bezier Circle を追加し、それを拡大します。Path Animation の下の Curve メニューで Follow を有効化し、100 になっている Frames をもっと妥当ものに設定しましょう (たとえば 16)。

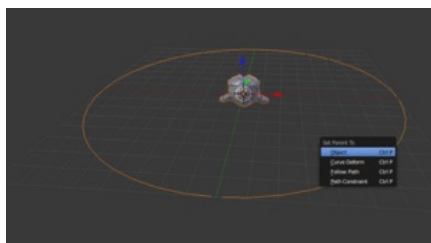


Settings for the object

Monkey を追加します。Duplication の下の Object メニューで、Frames を有効化し、Speed を無効化しましょう。

Speed

Speed オプションは、親子関係が Follow Path に設定されているときに使用されます (下記参照)。この例では、猿は円に沿って 16 フレームをかけて移動します。



Parenting

猿をベジェサークルの親にするには、まずサルを選択して、続いて曲線も選択し (カーブがアクティブなオブジェクトになるように)、そして CtrlP を押します。猿を選択して、AltO で原点にリセットしましょう。



Orientation tweaks

ここで、サルの向きを変更することができます。回転させるやり方でも Edit mode または Object mode のいずれでも)、Animation Hacks 下で Tracking Axes を変更するやり方でも (サルを選択した状態で) どちらでも行えます。サルの配置は、もちろん、曲線を編集することで、さらに磨きをかけることができます。

すべてのサルを実体のオブジェクトに変換するには、まず Ctrl⇧ ShiftA から Make Duplicates Real。これでもうすべてのサルは実体のオブジェクトになっていますが、ただしリンクされているコピーです。これを変更するには、Object→Make Single User→Object&Data そして All を選択しましょう。

Note

Dupliframes には代用が多くあります。どのツールを使用するかは状況によります。

1. 小さなカーブをプロファイルとして使用し、大きなカーブをパスとして使用するときは、前者を後者への Bevel Object として使用しましょう。
2. 曲線に沿ってオブジェクトを配置するには、Array Modifier と Curve Modifier 組み合わせると便利です。
3. Dupliverts を使用してオブジェクトを配置すると、例えば、円周に沿わせたり、または分割面に交差させたりできます。

外部リンク

- [Blender Artists: Dupliframes in 2.5](#)

エディット・モード (Edit Mode)

エディット・モードについて

あなたはオブジェクトの構造を2つのモードで扱えます。

オブジェクト・モード (Object Mode)

オブジェクト・モードでの操作はオブジェクト全般に影響します。オブジェクト・モードは 3D View において下図のヘッダを持っています:



オブジェクト・モード (Object Mode) のヘッダ

エディット・モード (Edit mode)

エディット・モードでの操作はただ1つだけのオブジェクトに作用しますが、その位置や回転角度のようなグローバル・プロパティ (Global Propertie) には影響を与えません。エディット・モードは 3D View において下図のヘッダを持っています:



エディット・モード (Edit Mode) のヘッダ

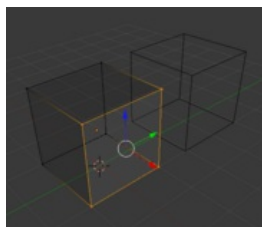
3D View におけるツールとモード類は(左から右へ):

View, Select および Mesh メニュー
Blender のモード (Blender Mode)
3D view における表示方法 (Display method for 3D view)
ピボット・センター (Pivot center)
3D マニピュレーターのウィジェット (3D manipulator widget)
選択モード (Selection mode)
深度バッファのクリップ (Depth buffer clipping (隠れている))
プロポーショナル・エディット (Proportional editing)
スナップ (Snap)
OpenGL レンダー (OpenGL render)

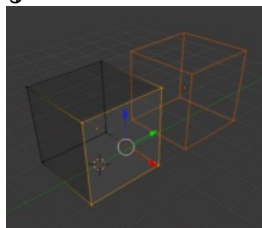
これら2つのモードは ⇄ Tab キーあるいは 3D View のヘッダ内のメニューから必要に応じて選択できます。

1つオブジェクトを作成した後は、User Preferences 内の Editing タブに有る Switch to Edit Mode ボタンの状態によって、即座にエディット・モードに遷移します。エディット・モードではアクティブな (active)オブジェクト、あるいは直前までに最後に選択されたオブジェクトただ1つだけに適用されます。

表示の仕様



1つの立方体が選択されている



エディット・モードに遷移する前に2つの立方体が選択されている

標準では、Blender はオブジェクト・モードとエディット・モードの両方で選択された物は橙色で強調されます。この色は User Preferences で変更できます (CtrlAltU→Themes)。

ワイヤーフレーム (Wireframe) シェーディングが有効になっている(Zキーで切り替え可能)オブジェクト・モードでは、選択されていないオブジェクトは黒く表示され、選択されると橙色で表示されます。もし1つ以上のオブジェクトが選択されていた場合、アクティブなオブジェクト、普通は最後に選択されたオブジェクト、以外の選択されたオブジェクトは全て暗い橙色で表示されます。同様に、エディット・モードでは選択されていない構造は黒く表示され、選択された面 (face)、辺 (edge) あるいは頂点 (vertex) が橙色に表示されます。アクティブな面は白色で強調されます。

エディットモードでは、一度に1つのメッシュのみを編集可能です。しなしながら、複数のメッシュを1つに結合する事 (オブジェクト・モードでCtrlJ)と、結合した物を再び分割する事 (エディット・モードでCtrlP) が可能です。エディットモードに遷移する前に複数のオブジェクトが選択されていた場合は、アクティブな選択範囲をなす物として選択されたオブジェクトは橙色で強調されたままになります。

1つのエッジ (edge) で繋がった2つの頂点が Vertex selection mode で選択された場合は、その2つの頂点の間のエッジも強調されます。同様に、1つの面を形成できる数の頂点やエッジが面によって選択された場合、その面もまた強調されます。

ツール・シェルフ (Tool Shelf)



エディット・モードにおけるツール・シェルフパネル (レイアウト上の問題からパネルは2つに分割されています。)

メッシュ・ツール (Mesh Tools) パネルは T キーを利用して展開・格納ができます。エディット・モードでは幾つかのメッシュ・ツールを利用できるようになります。

これらのツールの多くはショートカット (各ツールの Tooltips に表示される) 、あるいは Specials メニュー (W キー、CtrlE でアクセスできる Edge メニューもしくは CtrlF でアクセスできる Face メニュー) から利用できます。各ツールの状態依存のメニューはツール・シェルフの下部に表示されます。

User Preferences の Add-Ons メニューから更に多くのメッシュ編集用ツールを利用できます。新たなツールの開発に関しては Blender 関連のサイトやフォーラムで頻繁に公表されます。

このパネルに関するさらなる情報は [Reference panels](#) の節を参照ください。

プロパティ・シェルフ (Properties Shelf)



エディット・モードにおけるプロパティ・シェルフパネル (レイアウト上の問題からパネルは2つに分割されています。)

プロパティ・シェルフパネルは N キーを利用して展開・格納ができます。

プロパティ・シェルフにおいてメッシュ編集に直接関わるパネル類は、数値入力可能な Transform パネルと、法線あるいは距離、角度、領域の数値に関する Mesh Display パネルが利用できます。

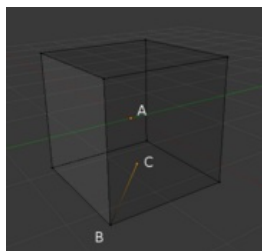
その他の有用なツールは表示オプションと Vertex groups を含む Object と Object Data の Context ボタン 内の プロパティ・エディタ (Properties Editor) で見ることができます。

このパネルに関するさらなる情報は [Reference panels](#) の節を参照ください。

頂点 (Vertex)、エッジ (Edge) と面 (Face)


基本的なメッシュ構造では、全てが3つの基礎的な構造から成り立ちます。即ち *頂点 (Vertex)*、*エッジ (Edge)* 及び *面 (Face)* です (ここではカーブ (Curve) や NURBS 曲線等に関しては触れません)。だからといって心もとなく感じる必要は有りません。これらの単純な構造が私達のモデルの基礎として余りある可能性を与えてくれるのです。

頂点 (Vertex)



頂点の例

頂点は 3D 空間における基本的な1つの点あるいは位置です。オブジェクト・モードあるいはレンダリング中は表示されないのが普通です。1つのオブジェクトの中心にある点は頂点ではありません。これは似ていますが、一回り大きく選択することもできません。*頂点の例*の図ではオブジェクトの中心点を "A" と表しています。"B" および "C" が頂点です。

新たな頂点を作る簡単な方法はエディット・モード (Edit mode) 上で Ctrl LMB  というショートカットを利用する物です。当然コンピューターのスクリーンは二次元でありますので Blender では頂点の位置を司る3つの要素を1回だけのクリックで定めることはできず、新しい頂点の位置は 3D カーソル (3D Cursor) の奥行きに置かれます。ここまで示した方法を用いると、直前までに選択された全ての頂点が自動的に新たに作られた頂点とエッジによって連結されます。上の図で C と定められた点は、エッジを B 点と C 点の間に追加されて新しく生成された頂点です。

エッジ (Edge)

エッジ (Edge) はいつも2つの頂点を1つの真っ直ぐな線でつなぎます。エッジはあるメッシュをワイヤーフレーム・ビューで見るときの"ワイヤー"です。レンダリングの結果には通常表示されません。面 (Face) を構築するために利用されます。2つの頂点を選択して F キーを押すことで1つのエッジを作ることができます。

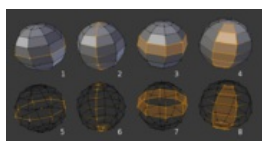
面 (Face)

面はメッシュ構造の中で最も高いレベルに位置します。面はオブジェクトの実際の表面を構築するために用いられます。メッシュのレンダリングの結果として見えるのはこの面です。1つの面は3つもしくは4つの頂点によるエッジで囲まれた三角形もしくは四角形の領域として定義されます。三角形は常に平面であり、計算を簡単にします。一方で、四角形は"上手くまるめられて"おり、細分化 (subdivision) モデリングでよく用いられます。

4つの辺を持つ面 (四角形の面) を利用する際は、内部的に単純に2つの三角形に分割されている事に注意してください。4つの辺を持つ面はそのフェイスが完全に平坦な (すべての頂点が1つの仮想的な面上に存在する) 場合でありかつ凸状である (内角の角度が180度以下の) 場合にのみ正常に動作します。立方体 (Cube) の面がこの条件を満たす例となります。そのワイヤーフレーム上で対角線を見ることが無いのは、暗に2つの三角形の面に分割されているためです。

三角形の面で立方体を構築することもできますが、エディット・モードで若干混乱を招くように見えます。3つもしくは4つの頂点で囲まれ、エッジで外形が囲まれているだけの領域は面ではありません。その領域が面を持っていない場合は、レンダリング結果では透明もしくは存在しない物として出力されます。面を作るためには、3つもしくは4つの頂点を選択して F キーを押してください。

ループ (Loop)



エッジ (Edge) と面 (Face) のループ (Loop)

エッジ・ループ (Edge Loop) と面ループ (Face Loop) は図で示されているように連続して"ループ"をなしている面あるいはエッジの集まりです。上側の 1~4 ではソリッド・ビューで、下側の 5-8 ではワイヤーフレーム・ビューで同じループを表示しています。

2 と 4 で示されたループはモデル全体をまわっていないことに注意してください。ループはそれ以上ループが続かないポール (Pole) と呼ばれる場所で止まります。ポールは3つ、5つもしくはそれ以上のエッジを繋ぐ頂点です。したがって、1つ、2つもしくは4つのエッジを繋ぐ頂点はポールではありません。

上の図において、ポールでループが終わっていない物はサイクリック (Cyclic, 1 と 3) であると呼ばれます。サイクリックならばループの始点と終点と同じ頂点となりモデルを2つに分断します。ループは1つのメッシュの特殊で連続した部分を高速で有効に扱い、有機的なアニメーションを作る上で必須となるツールです。Blender におけるループの扱いに関する詳細はマニュアルの [Edge and Face Tools](#) をご参照ください。

エッジ・ループ (Edge Loop)

上図において 1 および 2 で示したループはエッジ・ループ (Edge Loop) です。これはループ上の各頂点が、始点と終点がポールとなる頂点である場合を除いて必ず、ループ上に存在せずループの両側に隣り合った、2つの頂点を持つように連結されている物です。

エッジ・ループは有機的な (サブサーフェイスの) モデリングとキャラクター・アニメーションにおいて重要な存在です。正しく利用されれば、この機能は細分化のためのサーフェイス (Subdivision Surface) として利用した場合に、比較的少ない頂点ですべても自然に見えるモデルの作成と、アニメーションにおけるきれいな変形を可能にします。

体などの有機的なモデリングにおけるエッジ・ループを例として考えると、エッジ・ループは自然な姿の外側、そして肌およびその下の筋肉の変形ラインに沿って、キャラクターが動いたときにより激しく変形する、肩やひざ等の、部位には密集して置かれます。

エッジ・ループに関する詳細は [Edge Loop Selection](#) をご参照ください。

面ループ (Face Loop)

これはエッジ・ループの理論的な拡張であり、上図における ³ あるいは ⁴ で見られるように、2つのエッジ・ループの間の面からなります。⁴ で示されるように、面がポールを含み、サイクリックでない (non-circular) ループは面ループに含まれない事に注意してください。

面ループに関する詳細は [Face Loop Selection](#) をご参照ください。

Mesh Primitives

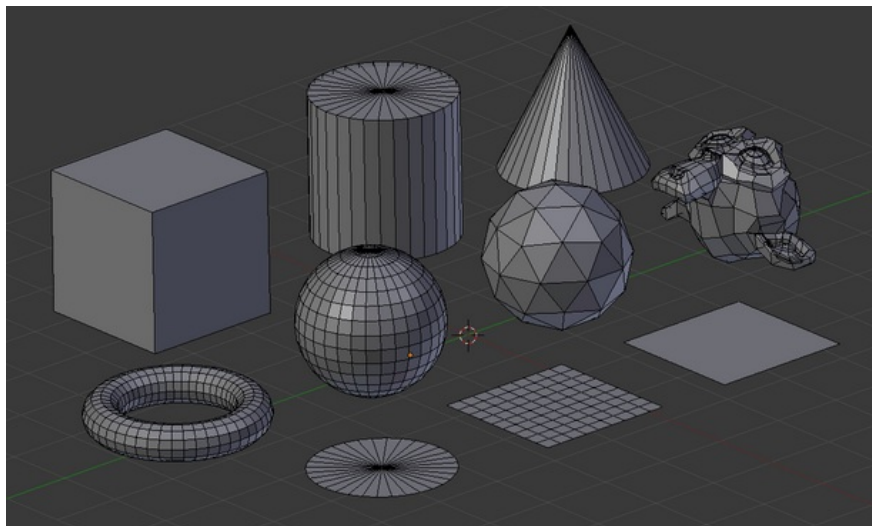
(メッシュプリミティブ)

モード: オブジェクトモード

ホットキー: ◊ ShiftA

メニュー: Add (追加) » Mesh (メッシュ)

3Dシーンで使われる一般的なオブジェクトがメッシュです。Blenderは、モデリングを始める礎となる「原始の(プリミティブ)」メッシュ形状をいくつか備えています。



Blenderの10種類のプリミティブ

複数のプリミティブで使えるオプション:

Radius (半径)

Circle (円)、Cylinder (円柱)、Cone (円錐)、UV Sphere (UV球) および IcoSphere (ICO球) の初期サイズを決めます。

Depth (深度)

Cylinder (円柱) と Cone (円錐) 用の高さの初期値です。

平らなプリミティブについて

平らなメッシュも、頂点を元の平面外に移動させれば立体的なメッシュになります (Plane (平面)、Circle (円)、Grid (グリッド) が該当します)。実際に、非常に複雑なメッシュを作るときにも、単純な円から始めることがよくあります。

Plane

(平面)

標準平面は4つの頂点、4つの辺と、1つの面を持ちます。机の上に広げた1枚の紙のようなものです。平らで厚みがないので、本当の3次元のオブジェクトではありません。平面を使って作れるものには、床、机の天板、鏡などがあります。

Cube

(立方体)

標準立方体は8つの頂点、20の辺、6つの面を持ち、本当の3次元のオブジェクトです。立方体から作れるものにはサイコロ、箱、木枠などがあります。

Circle

(円)

標準円は n 個の頂点でできています。頂点数と半径は、円を作った時に ツールシェルフ 下部に現れるコンテキストパネルで指定可能です。

Vertices (頂点数)

円を定義する頂点の数です。多いほど円弧が滑らかになります。逆に、頂点数が3の円は事実上三角形になります。円はよく、三角形や五角形などの多角形を追加するために使われます。

Radius (半径)

円の半径です。

Fill Type (面張りのタイプ)

円の面張りの方法を選びます。

Triangle Fan(三角形の扇型)

中央で頂点を共有する三角形で面を張ります。

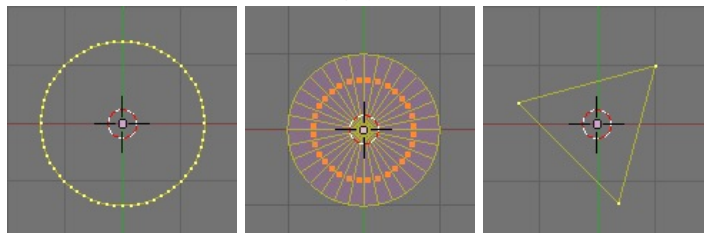
Ngon(Nゴン)

1枚のnゴンで面を張ります。

Nothing(なし)

面を張りません。頂点で円周だけを作ります。

さまざまな設定による円



64の頂点を持つ円は滑らかな円になります。

32の頂点を持つ円に面を張ったところ

3つしか頂点を持たない円は事実上三角形です。

UV Sphere

(UV球)

標準 UV球は n 個のセグメントと m 個のリングでできています。詳細度や半径は、UV球を作った時 ツールシェルフ に現れるパネルで指定可能です。セグメントやリングの数を増やすほど UV球の表面は滑らかになります。

Segments (セグメント)

縦方向のセグメントの本数です。地球の経線のように、極と極を結びます

Rings(リング)

横方向のセグメントの本数です。地球の緯線にあたります。

註記

UV球のセグメントとリングをどちらも6本にすると、リングを5つ、各極に点をそれぞれ1つ持つ形状ができ、上面図では(6つのセグメントによる)六角形のような形になります。このため、リングが1本足りない(逆に、極を半径0のリングとみなすなら1本多い)と思うかもしれませんが。

Icosphere

(ICO球)

ICO球は三角面でできた多面体です。細分化数と半径は、ICO球を作った時 ツールシェルフ に現れるパネルで指定可能です。ICO球は通常、UV球より頂点配置の等方性、効率性を得たいときに使われます。

Subdivisions(細分化)

球の定義に使われる反復の数です。数を増やすほど ICO球の表面は滑らかになります。細分化レベル1の ICO球は、正三角形 20枚でできた 20面体になります。細分化レベルを上げると各三角面が4つの三角面に分かれ、見た目が球に近づきます。

Size(大きさ)

球の半径です。

註記:

500回細分化された ICO球も作れますが、これだけ密集したメッシュを追加するとプログラムが確実にクラッシュします。10回細分化された ICO球でも 5,242,880 枚の三角面を持ちます。本当に気をつけてください!!

Cylinder

(円柱)

標準円柱は n 個の頂点でできています。断面の円の頂点数は、円柱を作った時 ツールシェルフ に現れるパネルで指定可能です。頂点数が多いほど断面の円が滑らかになります。円柱から作れるものには、取っ手や杖があります。

Vertices(頂点数)

円柱を定義するループ辺の頂点数です。

Radius(半径)

円柱の半径です。

Depth(深度)

円柱の高さです。

Cap Fill Type(ふたの面張りタイプ)

円と同じです(上記参照)。none(なし)にすると、オブジェクトはチューブ(管)になります。チューブから作れるものには、パイプやグラスがあります(円柱とチューブの基本的な違いは、前者の端が閉じていることです)。

Cone

(円錐)

標準円錐は「 n 個の」頂点でできています。円形の底面の頂点数や大きさ、および円錐の底面を閉じるオプションは、円錐を作った時 ツールシェルフ

に現れるパネルで指定可能です。頂点数が多いほど円錐の底面が滑らかになります。円錐から作れるものには、釘や三角帽子があります。

Vertices (頂点数)

円錐を定義するループ辺の頂点数です。

Radius 1 (半径1)

円錐底面の半径です。

Radius 2 (半径2)

円錐の先端の半径です。値を0にすると標準的な円錐形になります。

Depth (深度)

円錐の高さです。

Base Fill Type (基本の面張りタイプ)

円と同じです(上記参照)。

Torus

(トーラス)

ドーナツ型のプリミティブで、円を軸周りに回転させて作られます。全体の大きさは Major Radius (大半径) および Minor Radius (小半径) で指定します。(Major Segments (大セグメント数) と Minor Segments (小セグメント数))では、(セグメントの)頂点数を円ごとに指定できます。これらのオプションはトーラス作成時に ツールシェルフ に現れます。

Major Radius (大半径)

原点から断面の中心までの距離です。

Minor Radius (小半径)

トーラス断面の半径です。

Major Segments (大セグメント数)

トーラスの「緯線/大きな輪」のセグメント数です。トーラスを軸周りに「回転」したものだ考えると、これは回転の刻み幅を表します。

Minor segments (小セグメント数)

トーラスの「経線/小さな輪」のセグメント数です。円形のセグメントそれぞれの頂点数を表します。

Use Int+Ext Controls (外径と内径を使用)

トーラスの定義方法を変えます:

Exterior Radius (外径)

Use Int+Ext Controls (外径と内径を使用) が有効なら、回転の軸方向から見た時の、中心と外周間の距離です。

Interior Radius (内径)

Use Int+Ext Controls (外径と内径を使用) が有効なら、回転の軸方向から見た時の、中央にある穴の半径です。

Grid

(グリッド)

標準グリッドは 'n' × 'm' 個の頂点で作られています。X軸とY軸の解像度は、グリッドを作った時 ツールシェルフ に現れるパネルで指定可能です。解像度が高いほど多くの頂点で作られます。グリッドから作れるものには、例えば地形(プロポーション編集ツールや Displace モディファイアを使います)やその他の有機表面があります。プリミティブの平面を作って 編集モード で細分化してもグリッドと同じものを作れます。一方、User Preferences (ユーザー設定) から Landscape アドオンが利用できます。

X Subdivisions (X軸方向の分割数)

X軸方向の格子の本数です。最小値の3では2本のループ面ができます。

Y Subdivisions (Y軸方向の分割数)

Y軸方向の格子の本数です。

Size (サイズ)

グリッドの辺の長さです。

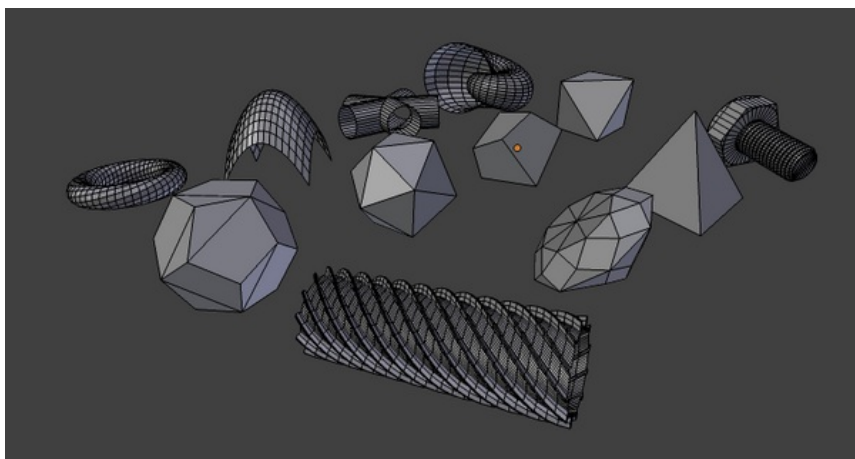
Monkey

(モンキー)

かつての NaN からコミュニティに寄贈されたもので、プログラマーのジョーク、あるいは「イースターエッグ」のようなものです。Monkey ボタンを押すと猿の頭が1つできます。名前は「Suzanne」で Blender のマスコットキャラクターです。Suzanne は [ユタティーポット \(Utah Tea Pot\)](#) や [スタンフォードバニー \(Stanford Bunny\)](#) と同じく、標準的なテスト用メッシュとして非常に便利です。

Add-ons

(アドオン)



アドオンとして利用可能なプリミティブの例

これらの幾何学的なプリミティブに加えて、Blenderにアドオンとしてあらかじめインストールされた、スクリプト生成によるメッシュが数を増やしています。こうしたメッシュは、User Preferences(ユーザー設定)のAddons(アドオン)タブで(Categories(カテゴリ)のAdd Mesh(追加メッシュ)でフィルターできます)有効化すると利用できます。さらつご紹介します:

[Landscape](#)

地形のプリミティブを追加します。ツールシェルフにパラメータやフィルターがたくさん現れます。

[Pipe Joints](#)

5種類のパイプジョイントのプリミティブを追加します。ツールシェルフで半径、角度、その他のパラメータを変えることができます。

[Gears](#)

歯車や [ウォームギヤ](#) のようなプリミティブを追加します。ツールシェルフに、形を制御する多くのパラメータが現れます。

Mesh Analysis

(メッシュの分析)

(訳注:この機能はバージョン 2.67 以降で利用できます)

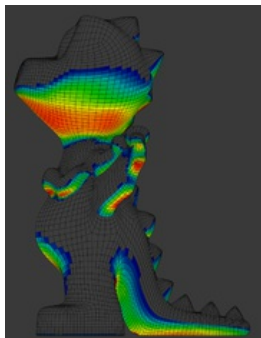
メッシュ分析は、特定用途に影響を与えるメッシュの属性を表示するのに役立ちます。

メッシュ分析は編集モードで動作し、高い値を持つ箇所を赤、低い値を持つ箇所を青で表示します。範囲外のジオメトリは灰色で表示します。設定パネルはプロパティシェルフ(N)に現れます。

現在のところ様々なモードの主用途は 3D プリントです。

Overhang

(オーバーハング/突き出し)

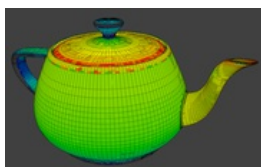


突き出し

Extrusion 式の 3D プリントは物理的に、限度を越えて突き出た部分を印刷できません。このモードは角度の範囲や選択軸に基づいて、突き出た部分を表示します。

Thickness

(厚み)

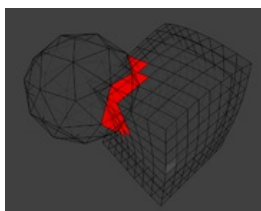


厚み

プリンターには壁の厚み制限があり、非常に薄い部分は印刷できません。この分析はレイカスティングとジオメトリの厚みの検出距離を使います。

Intersections

(交差)



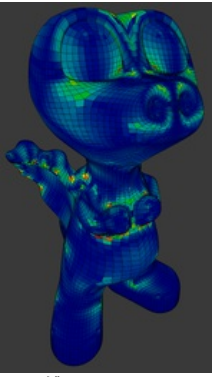
交差した面

もう一つ、印刷でよく問題になるのが表面同士の交差で、交差部分ではモデルの内側/外側が正確に検出できないことがあります。

交差の設定はオンとオフのどちらかで、他の表示モードと違って他に設定はありません。

Distortion

(歪み)



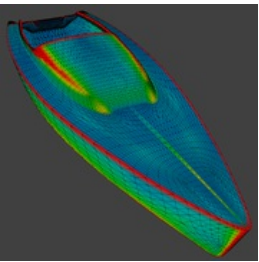
歪んだ面

歪んだ n ゴンの三角面化は未定義のため、歪んだジオメトリは問題を起こすことがあります。

歪み具合は、別の向きを指す部分のある、平らでない面を使って測られます。

Sharp Edges

(鋭い辺)



Sharp edges

壁の厚みと同じく、鋭いエッジは印刷できないほど薄くなる場合があります。

ご注意



メッシュ分析には既知の制限がいくつかあります。

- 現在のところ Deform (変形) タイプのモディファイアでのみ表示されます。
- ハイポリメッシュでメッシュ編集集中に使うと遅くなります。

Selecting Mesh Components

(メッシュ要素の選択)

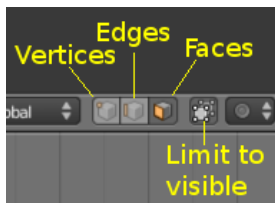
要素を選択する方法はたくさんあり、現在のメッシュ選択モードによって利用できる選択ツールが変わります。最初にこのモードを復習して、その後で基本的な選択ツールを見てみましょう。

Selection Modes

(選択モード)


Select Mode Header Widgets

(ヘッダーの選択モードウィジェット)



編集モードの選択モードボタン

編集モードには 3つの選択モードがあります。ツールバーにある3つのボタンの1つを選んで、そのモードに入ることができます。

⇧ Shift LMB  クリックでボタンを複数選択すると、同時に複数の選択モードに入れます。

頂点

選択頂点がオレンジ、非選択の頂点は黒、アクティブな頂点と最後に選択した頂点が白で描かれます。

辺

このモードでは頂点は描かれませんが、代わりに選択した辺がオレンジ、非選択の辺は黒、アクティブな辺と最後に選択した辺は白で描かれます。

面

このモードでは、面の中心に面選択用の選択点が描かれます。選択された面とその選択点はオレンジ、非選択の面は黒、アクティブと最後に選択した面は白で描かれます。

3つのメッシュ選択モードすべてで 回転、拡大縮小、押し出しなど、ほぼすべての編集ツールが利用可能です。もちろん 単一の頂点を回転したり拡大縮小したりしても、ピボットが他の場所に設定されているとき以外は役に立たないので、選択モードによって多かれ少なかれ適切でないツールがあります。

註記

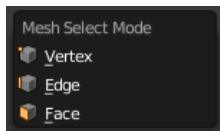
3つの選択モードボタンは 編集モードでのみ利用可能です。選択、非選択、アクティブ状態を示す色は、現在のテーマの設定に完全に従います。黒、オレンジ、白はデフォルトのテーマでの描画色です。

Select Mode Pop-up

(選択モードのポップアップ)

モード: 編集モード

ホットキー: Ctrl⇧ Tab



メッシュ選択モードメニュー

ポップアップメニューで選択モードの切り替えができます

選択モード » 頂点

Ctrl⇧ Tab を押してポップアップメニューで 頂点 を選ぶか、Ctrl⇧ Tab1 を押します。

選択モード » 辺

Ctrl⇧ Tab を押してポップアップメニューで 辺 を選ぶか、Ctrl⇧ Tab2 を押します。

選択モード » 面

Ctrl⇧ Tab を押してポップアップメニューで 面 を選ぶか、Ctrl⇧ Tab3 を押します。

Switching select mode

(選択モードの切り替え)

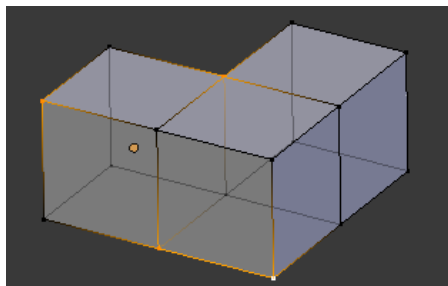
頂点から辺、辺から面のように「祖先」方式のモードに切り替えると(つまり単純なものから複雑なものに変えると)、選択していた複数の要素が切り替え後のモードでひとつの要素を形成できる場合には、その要素は選択されたままになります。

例えば、面の四辺を選択して 辺モードから 面モードに切り替えても、面は選択されたままです。新たなモードで要素を形成しない選択要素はすべて選択が解除されます。

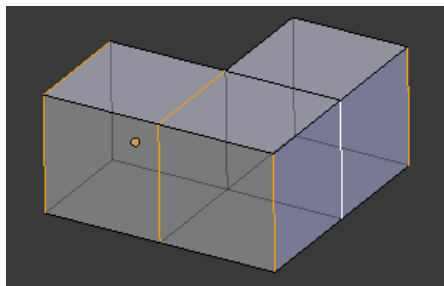
したがって「子孫」方式のモードに切り替えると(つまり複雑なものから単純なものに変えると)、「高レベル」要素を定義するすべての要素(例えば面)が選択されます(例えば四角形の4つの頂点または4つの辺)。

高レベルの選択モードに切り替えるときに Ctrl を押し続けると、現在の選択要素に接しているすべての要素が、完全な一要素を形成していなくても、追加されます。

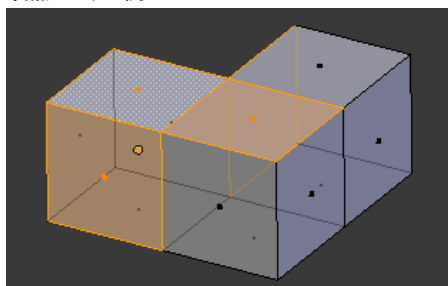
下の画像は各モードの例です。



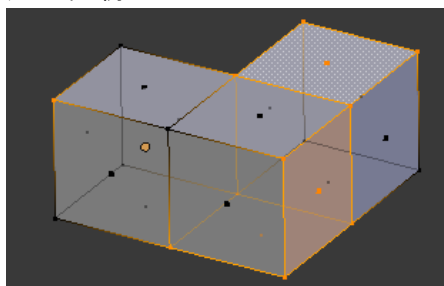
頂点モードの例



辺モードの例



面モードの例



混合モードの例

Selection Tools

(選択ツール)

編集モードの選択メニューには、要素選択用のツールが含まれます。これらについては、次のページで詳細を述べます。

Border Select (矩形選択)

四角形の領域を選択に使用します。

Circle Select (円選択)

円形の領域を選択に使用します。

(De)select All (すべてを選択/選択解除) A

メッシュの全要素を選ぶか何も選びません。

Invert Selection (選択の反転) Ctrl

非選択の要素を選択し、選択している要素の選択を解除します。

Select Random (ランダムに選択)

ランダムな頂点、辺、面のグループをパーセンテージ値に基いて選択します。

Checker Deselect (チェッカー選択解除)

一つおきに面を非選択にし、チェス盤のようなパターンを作ります。

Select Sharp Edges (辺の鋭さで選択)

二つの面の間にあって指定角度より狭い角度を作る辺を、すべて選びます。角度の入力パネルはツールの実行後に、ツールシェルフの最下段に現れます。狭くするほど鋭い辺だけが選択されます。**180°**にするとすべての「manifold (多様体)」の辺(後述)が選ばれます。

Linked Flat Faces (平らな面のつながり/リンクするフラットな面) (Ctrl⇧ ShiftAltF)

しきい値の角度より緩やかに繋がった面を選択します。平らな面の集まりを選ぶのに役立ちます。

Interior Faces (埋もれた面)

属する面が二つより多い辺だけで作られた面を選びます。

Side of Active (アクティブな側)

ひとつの軸方向にあるメッシュの構成項目をすべて選びます。

Select Faces by Sides (頂点数で選択)

指定数の頂点/辺を持つすべての面を選びます。

Select Non Manifold (非多様体) (Ctrl⇧ ShiftAltM)

ジオメトリに完全に囲まれていない頂点を選択します。これには端にある辺、浮いた辺、孤立した頂点を含みます。頂点モードでのみ使えます。

Loose(分離形状)

面の一部でない頂点や辺をすべて選びます。

Similar(類似)

特定のプロパティの類似度を基準に、要素を選びます。

More(より多く) CtrlNum+

選択中要素に隣接する要素を、選択要素に加えます。

Less(より少なく) CtrlNum-

選択中要素の境界線上にある要素の選択を解除します。

Mirror(ミラー)

ミラー反転した位置にあるメッシュの要素を選びます。

Linked(つながり/リンク) CtrlL

現在の選択要素に繋がっているすべての要素を選びます。

Select Every N Number of Vertices(N 個おきに頂点を選択)

N の倍数番目の頂点を選びます。

Shortest Path(最短パス選択)

二つの選択頂点間の頂点のパスを選びます。

Edge Loop(ループ辺)

選択要素からのループ辺を選びます。

Edge Ring(リング辺)

選択した辺と平行で、同じリング面にある辺を選びます。

Loop Inner-Region(ループの内側/内側領域のループを選択)

選択した辺のループ内にある面の領域を選びます。

Boundary Loop(境界ループ)

選択中の面の周囲の境界部分の辺を選びます。

Selectable Elements

(選択可能な要素)

[メッシュの構造のページ](#) で見たように、メッシュは「別種類の」要素である「頂点」、「辺」、「面」で作られています (実際には相関関係があるとしても。というのは、同じ基本データの別の「視点」/「表現」であるとも言えるからです...)。

ゆえに、メッシュの部分はこの3種類のうちのひとつを使って選択することができます。理解すべきキーポイントがひとつ、ここにあります。「1種類の要素を選ぶと(例えば辺)、これは暗に付随する他の種類の要素(例えばこの辺を定義するすべての頂点です。同じ辺により完全に定義される面も同様です)を選択しています」。これはとても重要で、ツールには動作対象が頂点、辺、面のいずれかに絞られるものがあるからです。頂点の選択部に対して「面の」ツールを使うと、その頂点で定義される面だけが影響を受けます。

通常は、同時に選択するのは使っている「選択モード」に応じた1種類の要素だけになるでしょう。しかしながら選択モードを切り替えて、引き続き別の種類の要素を同じ選択対象の中に付け加えることができます (選択モードの切り替え後に選択されるものについて、[この下](#)をご覧ください)。もしくは「混合」選択モードを使うことができますが、これも後述します。

Select Modes

(選択モード)

選択モードの切替方法は2つあります:

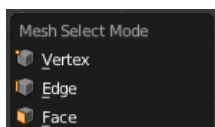
Select Mode popup

(選択モードのポップアップ)

モード: 編集モード

ホットキー: Ctrl⇧ Tab

編集モードにはメッシュ用の3つの選択モードがあります (図「選択モードメニュー」をご覧ください)。



選択モードメニュー

Select Mode (選択モード) » Vertices (頂点)

Ctrl⇧ Tabを押してポップアップから頂点を選ぶか、もしくはCtrl⇧ Tab1 NumPadを押します。選択した頂点は黄色で、非選択の頂点はピンクで描かれます。

Select Mode (選択モード) » Edges (辺)

Ctrl⇧ Tabを押してポップアップから辺を選ぶか、もしくはCtrl⇧ Tab2 NumPadを押します。このモードでは頂点は描かれませんが、選択された辺は黄色で、非選択の辺は黒で描かれます。

Select Mode (選択モード) » Faces (面)

Ctrl⇧ Tabを押してポップアップから面を選ぶか、もしくはCtrl⇧ Tab3 NumPadを押します。このモードでは面の中央に、面の選択に使うための点が描かれます。選択した面は黄色、非選択の面は黒で描かれます。

回転、拡大縮小、押し出しを含むほとんどすべての編集ツールを、これら3つのモードでは使うことができます。ただ、単体の頂点を回転させたり拡大縮小させたりするのは当然ながら無意味です。ツールによっては多かれ少なかれ適切でないモードがあります。

Select Mode header widgets

(選択モードウィジェット)

モード: 編集モード

パネル: 3Dビューのヘッダー



編集モードの選択モードボタン

ツールバーにある3つのボタンを使っても、選択モードの切り替えが可能です (図「編集モードの選択ボタン」をご覧ください)。

ボタンを ⇧ Shift LMB クリックすると「混合」もしくは「組み合わせ」モードを使えます。頂点、辺、面を同時に選択できるようになります。

註記

モード選択ボタンはメッシュが編集モードに入っているときにだけ表示されます。

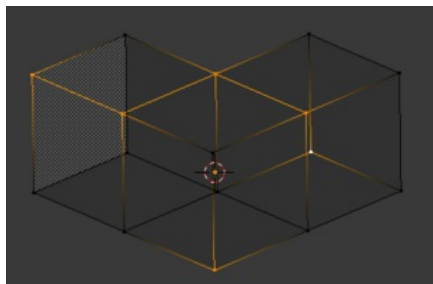
Selected elements after switching select mode

(選択モードを切り替えた後の選択要素)

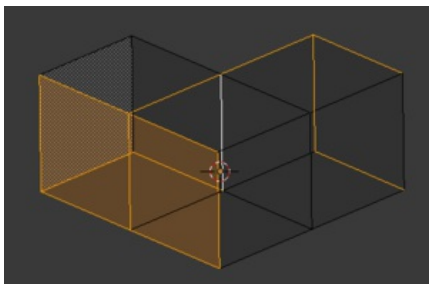
頂点 から 辺、辺 から 面 のように「上位」方式のモードに切り替えると(つまり単純なものから複雑なものにすると)、選択要素が新モードの1要素を形成できる場合には、その要素は選択されたままになります。

例えば、面の四辺を選択して 辺モード から 面モード に切り替えても、面は選択されたままです。新たなモードで要素を形成しない選択要素はすべて、選択が解除されます。したがって「下位」方式のモードに切り替えると(つまり複雑なものから単純なものに変えると)、「上位」要素を定義するすべての要素(例えば面)が選択されます(例えば四角形の4つの頂点または4つの辺)。

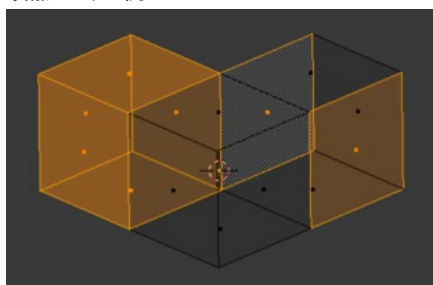
下図は各モードの例です。



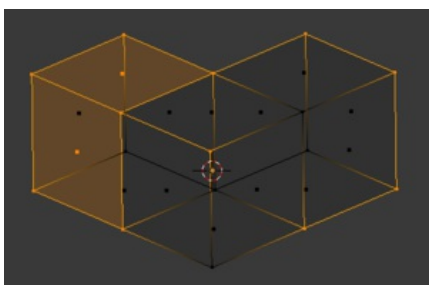
頂点モードの例



辺モードの例



面モードの例




混合モードの例

Basic Selection

(基本的な選択)


モード: 編集モード

ホットキー: RMB  および  Shift RMB 


要素の選択にもっともよく使われるのが、項目を RMB  クリックする方法で、これは既にある選択要素を新たな項目で置き換えます。

Adding to a Selection

(選択要素への追加)

既存の選択要素に追加するには、 Shift を押しながら右クリックします。選択した項目を再度右クリックすると、選択を解除します。

オブジェクトモードと同じで、より薄い色をした アクティブな (*active*) 要素が1つだけあります (通常は、最後に選択された要素です)。使っているツールによっては、この要素が非常に重要になるかもしれません!

なお、重なっている要素のどれを選ぶのかを決めるオプション(オブジェクトモードでAlt RMB  クリックをしたときのような)は用意されていません。ただし、ビューポートの陰影処理モードがソリッド/シェイデッド/テクスチャーのいずれかなら(パウンディングボックスとワイヤーフレーム以外なら)、キューブに似た4番目のボタンがヘッダーの選択モードボタンのすぐ右に現れます。

ボタンを有効化すると、見えている要素だけを選択できるように制限が入り(オブジェクトがソリッドであるかのように)、背面にあるか隠された項目を誤って選択、移動、削除することを防ぎます。

Selecting Elements in a Region

(範囲内の要素の選択)

モード: 編集モード

ホットキー: B、C および Ctrl LMB  クリックとドラッグ

範囲選択を使うと、3Dビュー内の2Dの範囲内にある要素の集まりを選択できます。この範囲は円形と四角形のどちらかになります。円形の範囲は編集モードでのみ使えます。四角形の範囲、別名「ボーダー選択」は、編集モードとオブジェクトモードの両方で使えます。


註記
この2つの選択ツールで選ばれたものは Solid Viewport Shading Mode (ソリッド陰影処理モード) の Limit Selection to visible (隠面処理) 機能(3Dビューポートで利用可能)の影響を受けます。

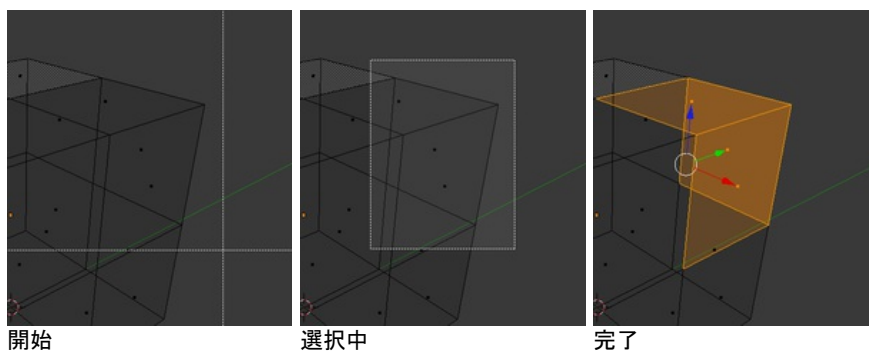
例えば、



1. ソリッドシェイディングモードと面選択モードでは、選択範囲の *内側* の面がすべて選択されます。
2. ワイヤーフレームモードと面選択モードでは、選択範囲の内側に選択ハンドルがある面だけが選択されます。

Rectangular region (Border select)

(矩形範囲(ボーダー選択))

ボーダー選択は編集モードとオブジェクトモードのどちらでも利用できます。ツールを開始するには B を使ってください。LMB  をドラッグして矩形を描き、オブジェクトのグループを選択します。ツールを使っている間、矩形に含まれるか触れたオブジェクトをすべて選択します。直前にアクティブだったオブジェクトがグループに含まれる場合、そのオブジェクトは選択され、アクティブになります。





上の図「開始」では、カーソルが点線の十字に変わっており、ボーダー選択が選ばれたことがわかります。図「選択中」では、*選択範囲*が LMB  で矩形を描くことで選ばれています。選択範囲は3つの面の選択ハンドルだけを覆っています。最終的に、LMB  を離すと選択は完了です。図「完了」をご覧ください。

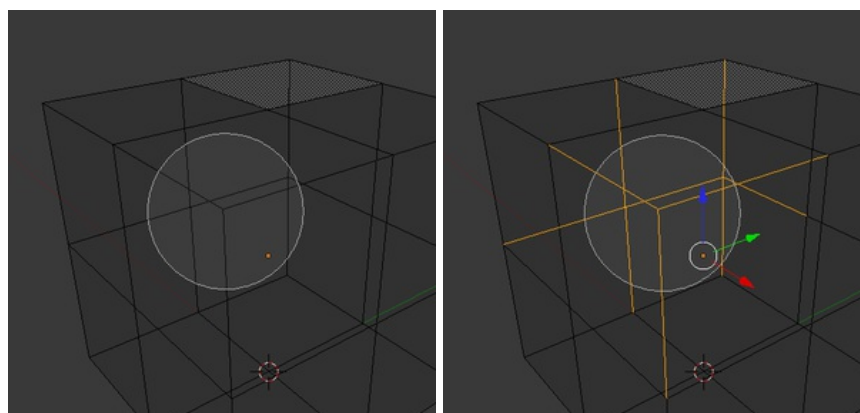
註記
ボーダー選択は直前の選択要素に付け足されます。したがって矩形範囲内にあるものだけを選びたい場合は、最初に全選択解除 A を行なってください。また、矩形範囲内のすべてのオブジェクトを非選択にするには、MMB  を押しながらボーダーを描きます。

Circular region

(円形範囲)


この選択ツールは 編集モードでのみ利用でき、C で開始します。このモードに入ると、2D の円で囲われた十字にカーソルが変わります。このツールは現在の選択モードが何であっていても使えます。LMB  でクリックまたはドラッグすると、円の中にある要素が選択要素に追加されます。



円形範囲の伸縮には + NumPad と - NumPad、または Wheel  を使います。



選択前
円形範囲の選択

選択後

上の図「円形範囲の選択」は 辺選択モードで辺を選択した例です。円と交差した辺はすぐに選択されます。ツールはインタラクティブで LMB  で円形範囲をドラッグしながら辺を選択できます。

選択を解除したいときは MMB  を押し続けるか Alt LMB  を押して再度クリックかドラッグします。


面選択モードでは、円形範囲が面の選択点 (通常小さな四角いピクセルで表現され、面の中央にあります) と交差する必要があります。


このツールから抜けるには RMB  クリックするか、Esc キーを押します。

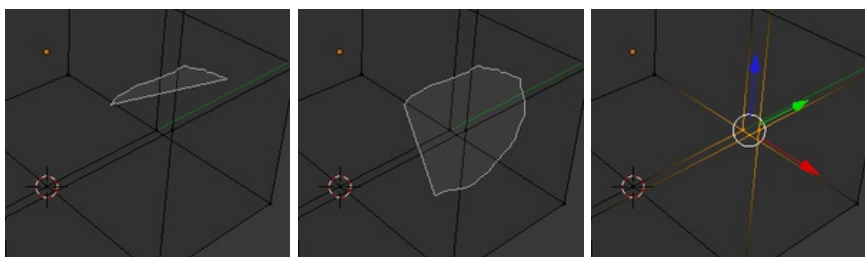
Lasso region

(投げ縄範囲)

投げ縄 (Lasso) 選択は範囲内のオブジェクトを選ぶ点では ボーダー選択と同じですが、投げ縄は手描きで定義され、通常は環状の丸い形状になります。投げ縄のようなものです。

投げ縄は編集モードとオブジェクトモードのどちらでも使えます。ツールを開始するには Ctrl LMB  ドラッグしてください。投げ縄と ボーダー 選択の違いは、オブジェクトモードでは、投げ縄ではオブジェクトの中心が投げ縄の範囲に交差したときだけ、そのオブジェクトを選択することです。

選択を解除するにはドラッグ中に Ctrl+ Shift LMB  を押します。



開始
投げ縄選択

選択中

完了

上図「投げ縄選択」は 投げ縄選択ツールを頂点選択モードで使った例です。

Additional Selection Tools

(さらなる選択ツール)

編集モードでは、選択メニューに要素を選択するためのさらなるツールがあります:

すべてを選択/選択解除 ((De)select All) A

メッシュ要素の全てを選択するか、何も選択しません。

選択の反転 (Invert Selection) CtrlI

非選択の要素をすべて選択し、選択されていた要素の選択を解除します。

より多く (More) CtrlNum+

選択されている要素に隣接する要素を選択要素に付け加えます。

より少なく (Less) CtrlNum-

現在の選択要素の境界線上の要素の選択を解除します。

Advanced Selection

(高度な選択)

編集モードの選択メニューにはさらに要素選択ツールがあります:

Mirror(ミラー)

鏡映位置にあるメッシュの構成項目を選びます

Linked(つながり/リンク) CtrlL

選択中要素につながった要素をすべて選びます

Select Random(ランダム)

パーセンテージ値に基いて、頂点、辺、面のランダムな集まりを選びます

Checker Deselect(チェッカー選択解除)

交互に面の選択を解除して、チェッカーのようなパターンを作ります

Select Every N Number of Vertices(N個おきに頂点を選択)

Nの倍数番目の頂点を選びます

Select Sharp Edges(辺の鋭さで選択)

二つの面の間にあつて指定角度より狭い角度を成す辺をすべて選びます。角度の入力パネルはツールの実行後に、ツールシェルフの最下段に現れます。狭くするほど鋭い辺だけが選ばれます。**180°**にするとすべての「多様体(後述)」の辺が選ばれます。

Linked Flat Faces(平らな面のつながり/リンクするフラットな面)(Ctrl⇧ ShiftAltF)

つながった面を、その間の角度のしきい値を基準に選びます。平らにつながる面の集まりを選ぶのに役立ちます

Select Non Manifold(非多様体を選択)(Ctrl⇧ ShiftAltM)

ジオメトリで完全に囲まれていない頂点を選びます。縁にある辺、浮いた辺、孤立した頂点などが該当します。頂点および辺選択モードでのみ利用できます

Interior Faces(埋もれた面)

属する面が二つより多い辺だけで作られた面を選びます

Side of Active(アクティブな側)

ひとつの軸方向にある、メッシュの構成項目をすべて選びます

Select Faces by Sides(頂点数で選択)

指定した辺の数(=頂点数)を持つすべての面を選びます

Loose(分離形状)

面の一部でない頂点や辺をすべて選びます

Select Similar

(類似選択)

モード: 編集モード

ホットキー: ⇧ ShiftG

メニュー: Select(選択) » Similar...(類似)

選択中のものに類似した属性を持つ要素を、ツールの実行後ツールシェルフに現れるパネル設定できるしきい値に基づいて選びます。選択モードによってオプションが変化します

頂点選択モード

Normal(法線)

選択中要素のものと似た法線の向きを持つ頂点をすべて選びます

Amount of Adjacent Faces(隣接面の数)

つながった面の数が似た頂点をすべて選びます

Vertex Groups(頂点グループ)

同じ [頂点グループ](#) にある頂点をすべて選びます

Amount of connecting edges(つながった辺の数)

つながっている辺の数が似た頂点をすべて選びます

辺選択モード

Length(長さ)

選択中のものと似た長さの辺をすべて選びます

Direction(方向)

選択中のものと似た向き(角度)を持つ辺をすべて選びます

Amount of Faces Around an Edge(辺周辺の面の数)

属する面数が似た辺を、すべて選びます

Face Angles(面の角度)

二つの面の間にある辺について、面の成す角度が選択中のものと似たものをすべて選びます

Crease(クリーズ)

選択中のものと同じ Crease 値をもつ辺をすべて選びます。Crease は [Subsurf モディファイア](#) で使われます

Bevel(ベベル)

選択中のものと同じ Bevel Weight を持つ辺をすべて選びます

Seam(シーム)

選択中のものと同じ Seam を持つ辺をすべて選びます。Seam はオン/オフのいずれかの値を持ち、[UV-texturing](#) で使われます

Sharpness(シャープ)

選択中のものと同じ Sharp 値を持つ辺をすべて選びます。Sharpはオン/オフのいずれかの値を持ち、[EdgeSplit モディファイア](#) で使われます

面選択モード

Material(マテリアル)

選択中のものと同じマテリアルを使う面をすべて選びます

Image(画像)

選択中のものと同じ UVテクスチャを使う面をすべて選びます ([UVテクスチャ](#) のページをご覧ください)

Area(面積)

選択中のものと似た面積を持つ面をすべて選びます

Polygon Sides(多角形の辺数)

選択中のものと同じ数の辺を持つ面を、すべて選びます

Perimeter(外周の長さ)

選択中のものと似た周の長さを持つ面をすべて選びます

Normal(法線)

選択中のものと似た法線を持つ面をすべて選びます。同じ方向(角度)を持つ面を選ぶことができます

Co-planar(同一面)

選択したものに近い平面にある面をすべて選びます

Selecting Loops


(ループの選択)

輪の形を成す要素を簡単に選ぶことができます:

Edge Loops and Vertex Loops

(ループ辺/ループ頂点)

モード: 編集モード → 頂点 か 辺選択モード

ホットキー: Alt RMB  または CtrlE → Edge Loop(ループ辺)

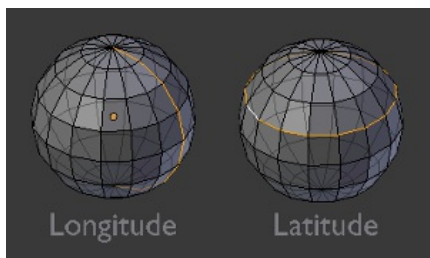
メニュー: Select(選択) » Edge Loop(ループ辺) または Mesh(メッシュ) » Edge(辺) » Edge Loop(ループ辺)

Altを押しながら辺を選択すると、ループ辺(Edge Loop)を選びます。ループ辺とは、クリックした辺を通して、端から端まで線状に並ぶ辺の集まりです。Alt+ Shiftを押しながらクリックすると、いま選択中の要素に追加します。

いま選択中の辺を基準にループ辺を選ぶこともできます。Select(選択) » Edge Loop(ループ辺) または Edge Specials(辺スペシャル)メニュー(CtrlE)のEdge Loop Select(ループ辺)を選びます。

頂点選択モード

頂点選択モードでも、同じホットキーを使って頂点ではなく「辺」を選ぶと、ループ辺を選ぶことができます



緯線と経線のループ辺

左の球では緯線方向の辺が選ばれています。ループの開き方に注目してください。これはループの判定が極にある頂点まで到達すると、この頂点が4つを越える辺につながっているため、終了するからです。一方、右の球では経線方向の辺が選ばれており、閉じたループになっています。これはループの判定が最初に始めた辺に戻ってこられたからです。

Face Loops

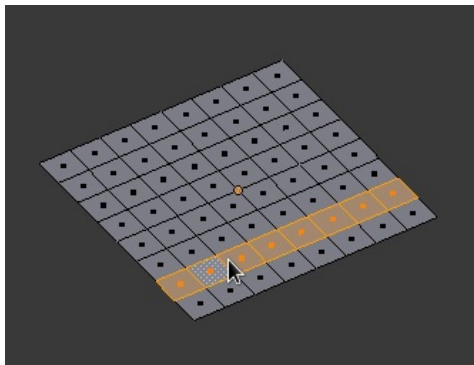
(ループ面)

モード: 編集モード → 面 または 頂点選択モード

ホットキー: Alt RMB

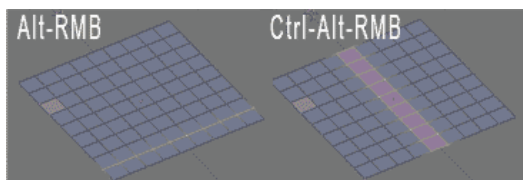
面選択モードで Alt を押しながら 辺 をクリックすると、端から端まで一本の線状に、対辺に沿ってつながる面を選びます。

頂点選択モードでは CtrlAlt を押しながら辺を選択すると暗にループ面を選択し、同じ結果になります。



選択されたループ面

このループ面は、面選択モードで Alt RMB で辺をクリックして選んでいます。ループは選択した辺から垂直方向に延びています。



頂点選択モードでの Alt と CtrlAlt の比較

ループ面を頂点選択モードで選ぶこともできます。技術的には CtrlAlt RMB でリング辺を選ぶと面の対辺が次々に選択され、頂点選択モードではこれは潜在的に面全体を選ぶことになり、結果としてループ面が選ばれます。

Edge Ring

(リング辺)

モード: 編集モード → 辺選択モード

ホットキー: CtrlAlt RMB または CtrlE → Selct(選択) » Edge Ring(リング辺)

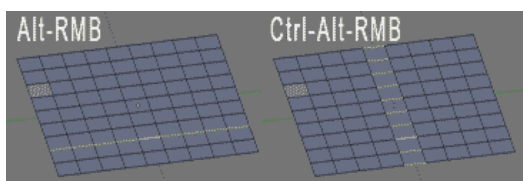
メニュー: Select(選択) » Edge Ring(リング辺) または Mesh(メッシュ) » Edges(辺) » Edge Ring(リング辺)

辺選択モードで CtrlAlt を押しながら辺を選択すると、その対辺が次々に [ループ面](#) 上で連続するような、つながっていない辺の集まりを選択します

ループ辺と同じく、Edge Specials (辺スペシャル) メニュー (CtrlE) にある Select(選択) » Edge Ring(リング辺) か Edge Ring Select(リング辺選択) オプションを使うと、現在の選択要素を基準にしてリング辺を選ぶことができます。

頂点選択モード

頂点選択モードで(頂点でなく)辺をクリックしたときも同じホットキーを使えますが、対応するループ面が直接選択されるでしょう




選択状態のループ辺とリング辺

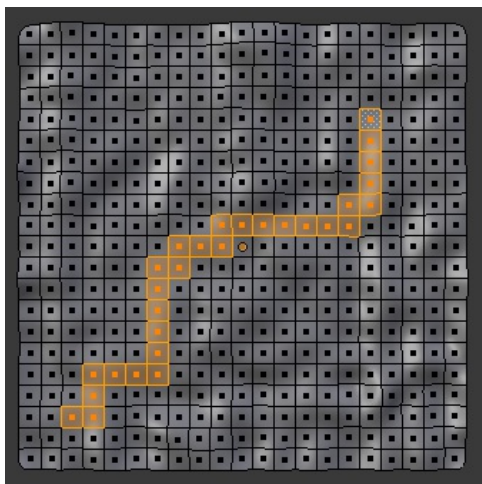
図「選択状態のループ辺とリング辺」では、左右どちらも同じ辺をクリックしていますが、別の命令が働いて別の「辺の集まり」が選ばれているところです。左は辺を基準に評価されており、右は面を基準にして評価されています。


Path Selection



(経路選択)

モード: 編集モード

ホットキー: Ctrl RMB 、および W → Select Vertex Path(最短パスを選択)



Ctrl RMB  を使った面や頂点経路の選択

辺および面選択モードで Ctrl RMB  すると、アクティブな辺/面からクリックした辺/面までの、最短経路上の辺/面をすべて選びます。頂点選択モードでは Ctrl RMB  は追加のオブジェクトの選択にすでに使われているので、頂点の経路を選ぶには Specials(スペシャル)メニュー-W または Vertex(頂点)メニュー-CtrlV → Select Vertex Path(最短パスを選択) を使う必要があります。

Loop Inner-Region

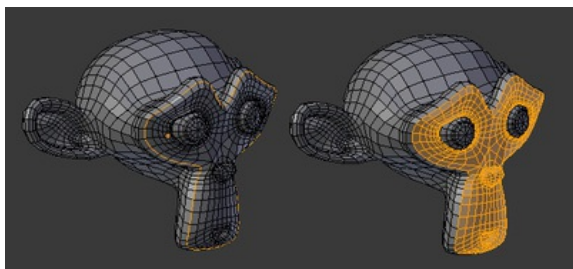
(ループの内側)

モード: 編集モード → 辺選択モード

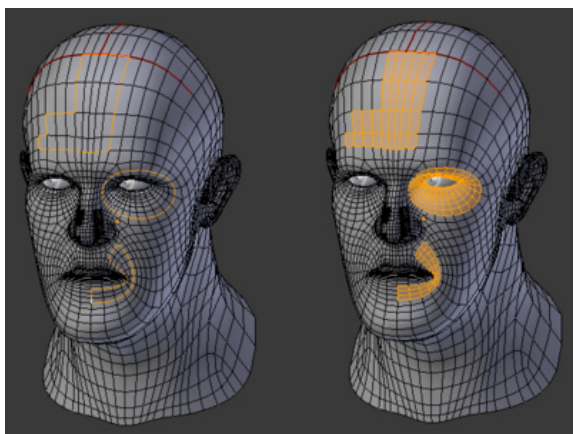
ホットキー: CtrlE → Select Loop Inner-Region(内側領域のループを選択)

メニュー: Select(選択) » Select Loop Inner-Region(内側領域のループを選択) または Mesh(メッシュ) » Edges(辺) » Select Loop Inner-Region(内側領域のループを選択)

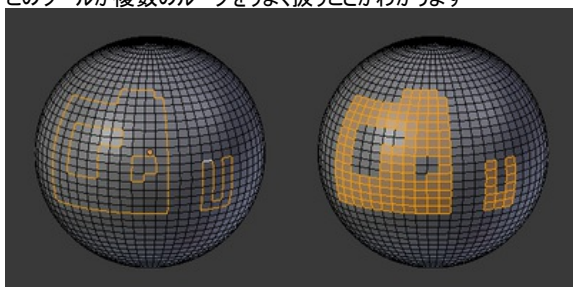
Select Loop Inner-Region(ループの内側/内側領域のループを選択)は閉じたループ辺の内側にある辺をすべて選びます。頂点選択モードや面選択モードでも使えますが、期待しない結果になるかもしれません。なお選んだループ辺が閉じていなければ、メッシュ上のつながった辺がすべて、ループ内にあるとみなされます。



Loop to Region(ループの内側)



このツールが複数のループをうまく扱うことがわかります



「穴」もうまく扱えます

Boundary Loop

(境界ループ)

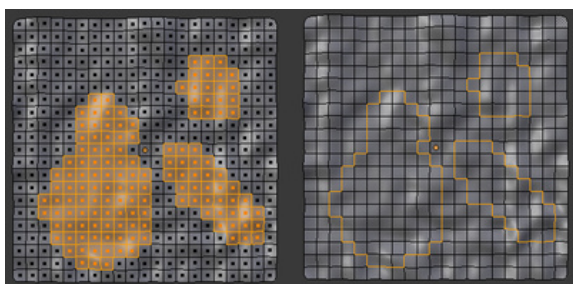
モード: 編集モード → 辺選択モード

ホットキー: CtrlE → Select Boundary Loop (境界ループを選択)

メニュー: Select (選択) » Select Boundary Loop (境界ループを選択) または Mesh (メッシュ) » Edges (辺) » Select Boundary Loop (境界ループを選択)

Select Boundary Loop (境界ループの選択) は Select Loop Inner-Region (ループの内側) の「論理的な反転」です。選択中の領域全体を見て、領域の境界線上の辺だけを選びます。どの選択モードでも使えますが、実行すると自動的に 辺選択モードに切り替わります。

実際に見ていただくとわかりやすいでしょう:



「境界ループの選択」は反対の操作で、強制的に辺選択モードに変わります

Selecting Edges

(辺の選択)



辺は頂点や面とほぼ同じ方法で選択できます。辺選択モードで右クリックです。◊ Shift を押しながらかlickすると、既存の選択への追加/削除を行います。

Edge Loops

(ループ辺)

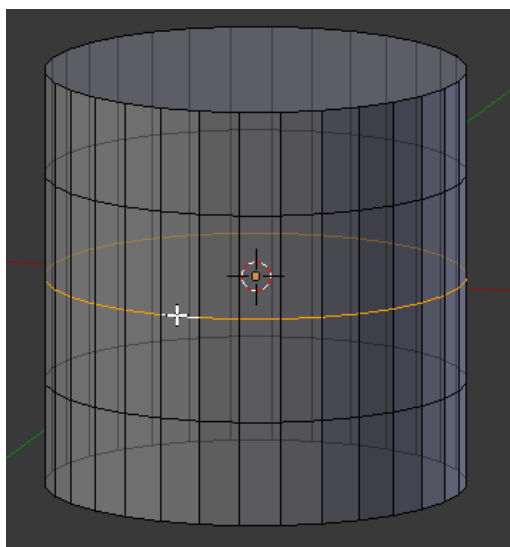
モード: 編集モード(メッシュ)

ホットキー: Alt RMB - または既存の選択要素の編集には ◊ ShiftAlt RMB

メニュー: Select(選択) » ループ辺(Edge Loop)

ループ辺の選択には、まず頂点か辺選択モードで辺を選び、次に Select(選択) » Edge Loop(ループ辺) を実行します。頂点か辺選択モードで辺に対してショートカット Alt RMB を使うと、同じ操作を素早く行えます。さらに、◊ Shift を押しながらかlickすれば、既存の選択要素に対してループの追加/削除が行えます。

なお、頂点選択モードでループを(追加で)選ぶ際、単に頂点を選ぶのに頂点を右クリックしたくなるかもしれませんが、ショートカットは辺に対して実行する必要があります。



Linux の Alt

Alt は Linux の一部の distribution では windows manager に使われています。上述のショートカットが動作しないときは、blender が Alt を正しく認識しているか確かめてください。

Edge Rings

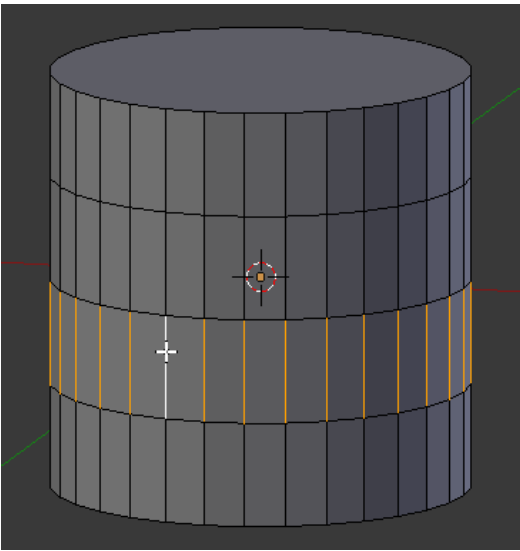
(リング辺)

モード: 編集モード(メッシュ)

ホットキー: AltCtrl RMB - または既存の選択要素の編集には ◊ ShiftAltCtrl RMB

メニュー: Select(選択) » Edge Ring(リング辺)

リング辺も同じように選べます。辺を選択して Select(選択) » Edge Ring(リング辺) を実行です。もしくは辺に AltCtrl RMB を使います。



選択対象を面全体に変える

辺選択モードでリング辺を選ぶと、面選択モードに切り替えたとき選択内容は解除されます。


これは、どの面も、面を構成するのに必要な四辺すべてが選ばれていなかった(二辺だけが選ばれています)ためです。

欠けている辺を手動で選んだり二度 ⇧ ShiftAlt RMB を使ったりするよりも、最初に頂点選択モードに入って選択要素を「あふれさせる」のが簡単です。それから面選択モードに入れば、正しく面を選択できるでしょう。

Selecting Faces

(面の選択)






メッシュの一部の面を使って選ぶには、面選択モードに切り替える必要があります。切り替えは上の画像にあるボタンをクリックするか **Ctrl⇨ Tab** を押してメニューを表示させて行います。選択はいつものように **RMB**  で行います。既存の選択への追加/削除は **⇨ Shift** を押しながらかlickしてください。

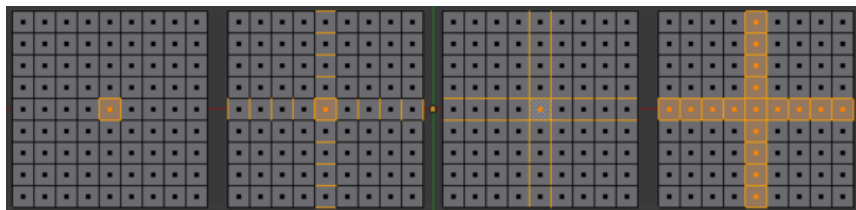
Face Loops



(ループ面)

モード: 編集モード(メッシュ)

ホットキー: **Alt RMB**  - または既存の選択要素の編集には **⇨ ShiftAlt RMB** 

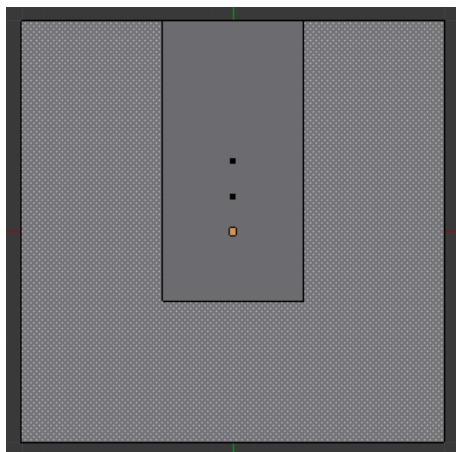
ループ面はリング辺に非常に似たものです。ループ面の選択を、面の選択から行うメニューは用意されていません。Select(選択) » Edge Ring(リング辺)を使うと、あらかじめ中央に選択しておいた面に「クロスする」ような選択が行われます。ループ面を選ぶのに辺選択モードに切り替えたくない場合には、ショートカット **Alt RMB**  を使ってください。



1. 面を選択したところ。
2. 面を選択して Select(選択) » Edge Ring(リング辺)を実行します。面選択モードにいるにも関わらず、Blenderの辺の選択の仕方を見てください。もしこれが目的の辺で、ここから作業したいのであれば、辺選択モードに切り替えてください。頂点選択モードに切り替えれば選択部分があふれ、さらに面選択モードに戻ると、4つ目の画像のような結果になるでしょう。
3. 面を選択して Select(選択) » Edge Loop(ループ辺)を実行します。上の例のように、Blenderは選択された面の4つの辺に基いて選択操作を行い、辺選択モードにいるかのように振舞います。
4. これは中央にある面の左辺を **Alt RMB**  クリックし、上辺を二度 **⇨ ShiftAlt RMB**  クリックして作られた選択結果です。二度繰り返すのは、初めのクリックで選択中のループ面を取り除くため(この例ではもともと選択されていた面だけです)、一方、二度目のクリックは縦方向に走るループをすべて選択要素に付け加え、十字を作ります。

Ngons in Face Select Mode

(面選択モードにおけるnゴン)



ご存知の通り、面の中央には小さな四角いドットがあります。nゴンがあると、このドットが表示を混乱させることがあります。この例では "U" の内側にある長方形の面の内側にある U字型の nゴンの中央ドットを示しています。どのドットがどの面のものなのか、答えるのは簡単ではありません(オレンジのドットはオブジェクトの中心です)。幸運なことに、さほど気にする必要はありません。面を選択するのに中央ドットをクリックする必要はなく、面そのものをクリックすればよいからです。



面の選択

面を選ぶには:

ドットでなく面をクリック!

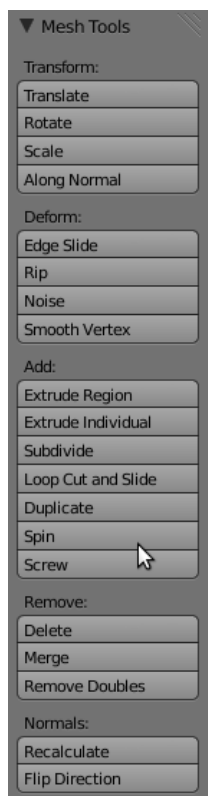
メッシュの編集

Blenderはメッシュを編集するためのさまざまなツールを提供します。これらは、Mesh Tools パレット、3Dビューのヘッダー内のメッシュメニュー、および3Dビューのコンテキストメニュー経由と同様に、個々のショートカットキーからも使用できます。

すべての“transform precision/snap” keys (Ctrl and/or ⇧ Shift) はすべてのこれらの高度な操作に対しても動作することを覚えておきましょう... ただし、それらのほとんどでは [axis locking](#) はできず、そしてそのうちのいくつかは、[pivot point](#) と [transform orientation](#) を考慮しません...

これらの変形ツールは、メニューバーのMesh menuの Transform セクションで利用できます。これらの一部は、曲線、サーフェス、格子のような他の編集可能なオブジェクトでも使用できることを覚えておきましょう。

ツールの種類



Mesh Tools

メッシュツールは様々な場所にあり、ショートカットを介しても同様に利用できます。

[Transform and Deform tools:](#)

- Translate
- Rotate
- Scale
- Mirror
- Shrink/Flatten/Along Normal
- Push/Pull
- To Sphere
- Shear
- Warp
- Edge Slide
- Noise
- Smooth Vertex
- Edge Flip
- Rotate Edge

[Add and Divide tools:](#)

- Make Edge/Face
- Fill
- Beauty Fill
- Solidify
- Quads to Tris
- Extrude Region
- Extrude Individual
- Subdivide
- Loop Cut/Slide
- Knife tool
- Duplicate
- Spin
- Screw

[Merge and Remove tools:](#)

- Delete
- Merge
- Remove Doubles
- Tris to Quads

[Separate tools:](#)

- Rip
- Split
- Separate

メッシュツールへのアクセス

メッシュツールパレット

メッシュを選択して、⇧ Tabで編集モードに切り替えると、Tool Shelfも Object Tools から Mesh Toolsに変わります。これらは、メッシュ編集ツール

のごく一部です。

メニュー

Meshは、ヘッダーバーに配置されています。メニューの一部はショートカットでアクセスすることができます:

CtrlFはFaceのツールメニューを表示します
CtrlEはエッジのツールメニューを表示します
CtrlVは頂点のツールメニューを表示します

基本的なメッシュ編集

この章ではメッシュの基本的な編集の仕方について説明します。

- [移動、回転、拡大縮小 \(en\)](#)
- [要素の追加 \(en\)](#)
- [要素の削除 \(en\)](#)
- [面と辺の作成 \(en\)](#)
- [ミラー編集 \(en\)](#)

移動、回転、拡大縮小

モード: 編集モード

パネル: メッシュツール (編集コンテキスト)

ホットキー: G/R/S

メニュー: メッシュ » トランスフォーム » 移動、回転、拡大縮小、...

1つ以上の要素を選択すると、Blender の他の多くのものと同じように移動(G)、回転(R)、拡大縮小(S)させることができます ([3D 空間での操作 \(en\)](#)を参照してください)。

これらの操作には 移動、回転、拡大縮小 ボタンを使うか、[トランスフォームマニピュレータ \(en\)](#)を使うか、または以下のショートカットを使います:

G、R、S

選択部分を動かしたあとは、ツールシェルフのオプションで変更の微調整や特定軸への効果の限定、プロポーショナル編集の切り替えなどを行えます。

特定種類の要素 (例: 辺) を動かしたときは、当然ながら、それに暗黙的に関係する他の種類の要素 (例: 頂点と面) も編集しています。

編集モードではこれらの基礎的な操作を行うときに、さらなるオプションが利用可能です: [プロポーショナル編集 \(en\)](#)

Deleting and Merging

(削除と結合)

これらのツールは要素を取り除くために使われます。

Delete

(削除)

Delete (削除) (X または Del)

選択した頂点、辺、面を削除します。この操作に次のような制限もできます:

Vertices (頂点)

現在の選択にあるすべての頂点を削除し、各頂点が繋がっていた面や辺もすべて取り除きます。

Edges (辺)

現在の選択にあるすべての辺を削除し、各辺と辺を共有していた面もすべて取り除きます。

Faces (面)

現在の選択にあるすべての面を取り除きます。

Only Edges & Faces (辺と面のみ)

選択した辺と隣接する面だけを削除対象にします。

Only Faces (面のみ)

選択した面を取り除きますが、その中にある辺は残します。

Edge Collapse (辺を束ねる)

複数の辺を一つの頂点に束ねます。面のループを取り除くときに使います。

Edge Loop (ループ辺)

ループ辺を削除します。現在の選択要素がループ辺でなければ、この操作は何もしません。

Dissolve

(溶解)

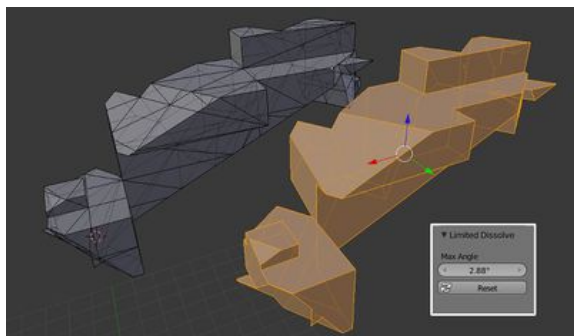
溶解操作も削除メニューからアクセスできます。ジオメトリを削除すると、埋め直さないといけない穴が開く可能性がありますが、その代わりに、溶解はジオメトリを削除し、取り囲むジオメトリを埋めます。

Dissolve (溶解)

選択したジオメトリを取り除きますが、サーフェスを閉じたまま、選択要素を効率的に単一の Nゴンに変えます。辺、面、頂点の選択生むによって溶解の振る舞いはわずかに変化します。必要な場所にディテールを追加し、必要がなければ簡単に取り除くことができます。

Limited Dissolve (限定的溶解)

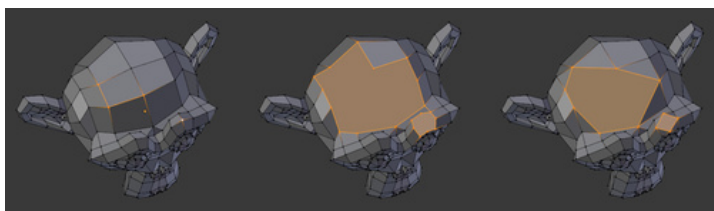
調整可能な角度のしきい値を基準にして、平面の細部や辺を減らします。



限定的溶解の利用例

Face Split (面分離)

頂点の溶解後にツールシェルフに現れるオプションです。周囲の面に頂点を溶解させたとき、段差のある非常に大きな nゴンになることがよくあります。面分離のオプションは、頂点につながった面のコーナーだけを使って溶解を行うようにします。



Dissolve 面分離 オプション。左:入力、中央:通常の溶解、右:面分離 が有効

Convert Triangles to Quads

(三角面を四角面に変換)

Tris to Quads(三角面を四角面に) AltJ

隣り合う三角面の共通辺を取り除いて四角面を作ります。複数の三角面が選択されているときに実行できます。

対象が三角面2つだけなら、選択してショートカット F で面を作っても同じ結果になります。

Unsubdivide

(分割の復元)

モード: 編集モード

ホットキー: CtrlE » Unsubdivide(分割の復元)

メニュー: Mesh(メッシュ) » Edges(辺) » Unsubdivide(分割の復元)

分割の復元は、細分化で作られた辺を削除することで、細分化操作の逆の操作を行います。細分化操作のあとで別の編集が行われていれば、期待しない結果になるかもしれません。

Iterations(反復)

取り除く分割の数です。

Merging(結合)

Merging Vertices

(頂点の結合)

モード: 編集モード

ホットキー: AltM

メニュー: Mesh(メッシュ) » Vertices(頂点) » Merge(結合)..., Specials(スペシャル) » Merge(結合) または Vertex Specials(頂点スペシャル) » Merge(結合)

選択頂点のうち、1つだけ頂点を残して他を削除することで結合します。残す頂点の決め方を、実行前のポップアップで選べます:

At First(最初に選択した頂点に)

頂点選択モードでのみ利用可能で、最初に選択した頂点に残りの選択頂点の位置をあわせませす。

At Last(最後に選択した頂点に)

頂点選択モードでのみ利用可能で、最後に選択した頂点(=アクティブ要素)に残りの選択頂点の位置をあわせませす。

At Center(中心に)

全ての選択モードで利用可能で、選択要素の中心にすべての選択頂点の位置をあわせませす。

At Cursor(カーソル位置に)

全ての選択モードで利用可能で、3Dカーソル位置にすべての選択頂点の位置をあわせませす。

Collapse(束ねる)

特別なオプションで、これを使うと「生き残る」頂点が2つ以上になる可能性があります。実際のところ、選択要素の「島(islands)」「繋がった頂点のグループ」の数だけ頂点を残します。残りの頂点の位置は、対応する「島」の中心にあわせませす。Mesh(メッシュ) » Edges(辺) » Collapse(束ねる) のメニュー操作からも実行できます。

頂点結合で辺や面が削除されることも当然ありますが、Blender は再結合に部分的にしか含まれない辺や面をなるべく保持しようとしています。

AutoMerge Editing

(重複頂点の自動結合)

モード: 編集モード

メニュー: Mesh(メッシュ) » AutoMerge Editing(重複頂点の自動結合)

重複頂点の自動結合の切り替えは、メッシュメニューで行えます。有効化すると、頂点の移動中に Limit(しきい値) の設定(メッシュツールパネルから行います。下記参照)よりも近くに他の頂点があるとすぐに、自動的に結合します。

Remove Doubles

(重複削除)

モード: 編集モード

パネル: 編集コンテキスト → メッシュツール

ホットキー: W » 4 または CtrlV » Remove Doubles(重複削除)

メニュー: Mesh(メッシュ) » Vertices(頂点) » Remove Doubles(重複削除)、Specials(スペシャル) » Remove Doubles(重複削除)、Vertex Specials(頂点スペシャル) » Remove Doubles(重複削除)

重複削除は、指定距離よりも近い頂点を結合して、メッシュを単純化するとき役立つツールです。代替手段として、[ポリゴン数削減\(Decimate\)モディファイア](#)を使ってメッシュを単純化する方法があります。

Merge Distance(結合距離)

結合距離のしきい値を Blender Unit で指定します。

Unselected(非選択部)

選択要素内の頂点が選択していない頂点と結合可能にします。無効化すると、選択頂点は選択されている別の頂点とだけ結合します。

Make Edge/Face

(辺と面の作成)

モード: 編集モード

ホットキー: F

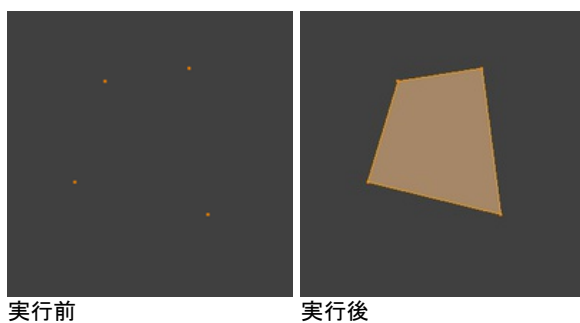
メニュー: Mesh(メッシュ) » Faces(面) » Make Face/Edge(面/辺の作成)

このツールは状況に応じて挙動が変わるツールで、選択部分を埋めることでジオメトリを作ります。2頂点だけが選択されていれば辺を作り、それ以外では面を作ります。

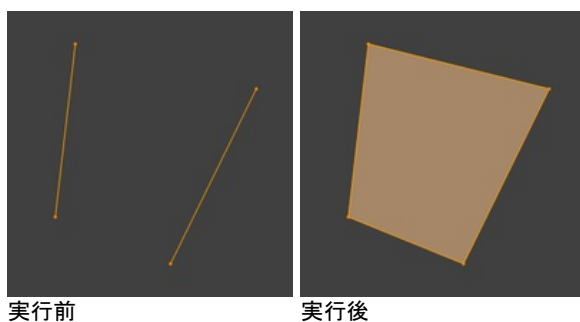
典型的な使い方は、頂点複数を選んで F を押すことですが、Blender はさまざまな選択状態から面を作ることができ、ジオメトリの素早い構築を支援します。

次の方法が状況に応じて自動的に使われます。

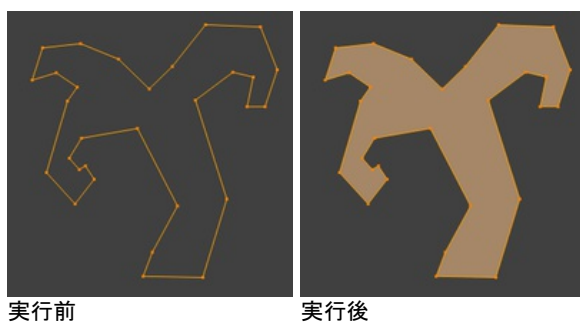
孤立した頂点。



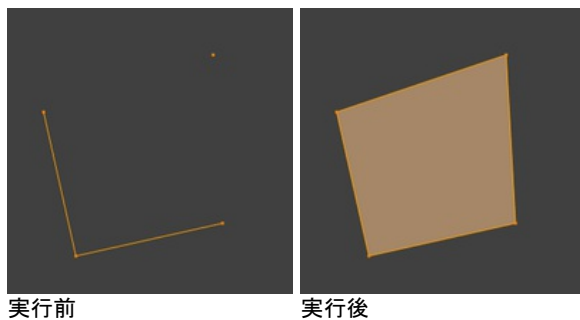
孤立した辺。



辺の集まりからの N-ゴン: 辺がたくさんあると Blender は nゴンを作ります。なお穴はサポートされません。穴を開けるには [Fill\(面張り\)](#) (AltF) を使う必要があります。



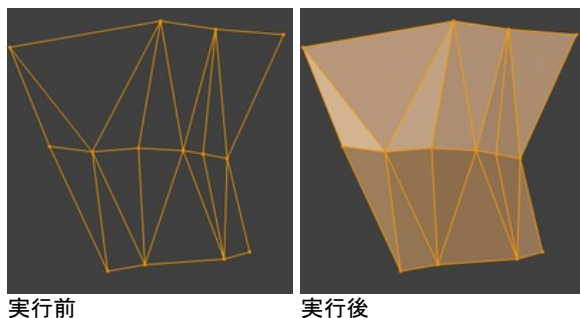
頂点と辺の混合: 余分な頂点だけでなくすでにある辺も面を作るのに使われます



実行前

実行後

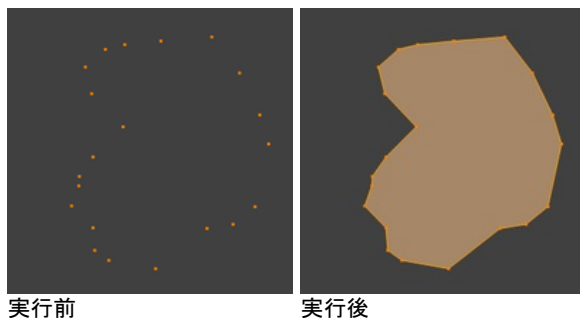
網状の辺: 内部に面を持たないたくさんの接続した辺ができることがあります



実行前

実行後

点の集まり: 孤立した頂点がたくさんあると Blender は nゴン用の辺を算出します



実行前

実行後

単独頂点の選択: 境界線上の単独頂点を選ぶと、面は境界線に沿って作られ、他の2つの頂点を自分で選択する手間が省けます。なおこのツールは面を作り続けるために複数回使えます



Further Reading

(参考)

その他の面の作成方法は次の記事をご覧ください:

- [Fill](#)
- [Grid Fill](#)
- [Bridge Edge Loops](#)

Mirror Editing

(ミラー編集)

X-Mirror

(Xミラー)

モード: 編集モード

パネル: Mesh Options(メッシュオプション) » X Mirror(Xミラー)

メッシュオプション パネルの X Mirror オプションを使うと、メッシュの「両側」を1回の操作で編集できます。ある要素(頂点/辺/面)をトランスフォームしたとき、(ローカル空間の)X軸を基準にちょうど左右対称位置に要素があれば、その要素も対称性を守りつつトランスフォームされます。

Topology Mirror

(トポロジーミラー)

モード: 編集モード

パネル: Mesh Options(メッシュオプション) » Topology Mirror(トポロジー反転)

Topology Mirror が効果を発するには、X Mirror オプションが有効にされていなければなりません。

X Mirror オプションを使って鏡映されたメッシュジオメトリに効果を発揮させるには、鏡映頂点をぴったり設置する必要があります。鏡映位置に正確に設置しなければ、X Mirror はその頂点を鏡映されたものとして扱いません。位置のずれた頂点はほんのわずかしかずれていないことが大半なので、いろいろするかもしれません。

Topology Mirror は、頂点位置だけでなく、メッシュジオメトリ内での他頂点との関係の仕方を見て、どの頂点が鏡映されたものであるかを検知し、この問題を解決しようとしています。このツールはある頂点を鏡映として扱うかどうかを調べるために、メッシュジオメトリのトポロジー全体を見ます。この結果、X Mirror と Topology Mirror の両方が有効化されているときは、鏡映頂点が左右対称位置になくても、依然として鏡映として扱われます。

なお、Topology Mirror の機能は、きめ細かなメッシュジオメトリであるほどうまく働きます。たとえば立方体や UV スフィアのような非常に簡単なメッシュジオメトリを使う場合は、Topology Mirror は働かないことが多くなるでしょう。

以下は Topology Mirror の使い方の例です。

1. Blender で新たなシーンを開始し、デフォルトの立方体を削除して、Monkey オブジェクトを 3D 描画域に追加します。
2. TAB キーを押して、Monkey オブジェクトの 編集モード に入ってください。
3. X Mirror オプションが無効な状態で、Monkey オブジェクトの頂点をひとつ、少し動かしてください。
4. 次に X Mirror オプションを有効にしますが、Topology Mirror は無効のままにします。
5. 同じ頂点をいま再び動かすと X Mirror は作動せず、鏡映にあたる頂点は変化しません。
6. ここで Topology Mirror を有効にして再度同じ頂点を動かすと、X Mirror はもう一方の頂点が完全に正しい位置になくても、まだ操作を鏡映するはずで。

Mirror Modifier

(ミラーモディファイア)

Xミラーが効果を発するための条件は非常に厳しいので、使いづらいかもしれません。メッシュ(の半分)の正確な鏡映を作るには、[ミラーモディファイア \(Mirror modifier\)](#) を使うほうが簡単でわかりやすいでしょう。

Snap to Symmetry

(対称にスナップ)

モード: 編集モード

メニュー: Mesh(メッシュ) » Snap to Symmetry(対称にスナップ)

The Snap to Symmetry ツールは、Blender が鏡映を検知できないくらい動かされた頂点を含む、ほぼ左右対称なメッシュに効果があります(たとえば X Mirror オプションが有効であるとき)。

こうした状況は、誤って X Mirror を有効にせず編集したときに発生します。他のアプリケーションからインポートされたモデルも、Mirror が失敗するほど非対称なことがあります。

Direction(方向)

スナップする軸と方向を指定します。任意の3つの軸と、正から負、負から正のいずれかを選べます。

Threshold(しきい値)

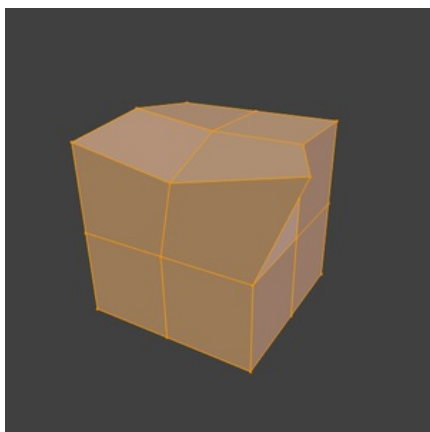
対応頂点を探す際に使う検索範囲を指定します。

Factor(強さ)

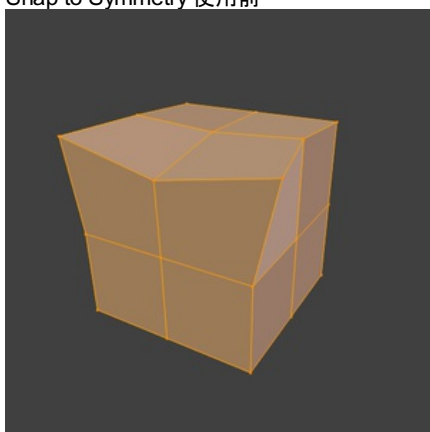
片側から反対側に、鏡映位置を混ぜ合わせるためのサポートです(0.5 は均等に両者を混ぜ合わせます)。

Center(中心)

中心軸にある頂点をゼロにスナップします。



Snap to Symmetry 使用前



Snap to Symmetry 使用後

Symmetrize Mesh

(対称化)

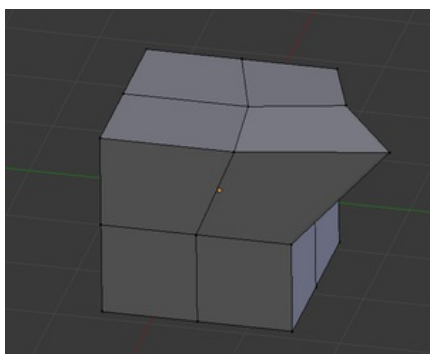
モード: 編集モード

メニュー: Mesh(メッシュ) » Symmetrize(対称化)

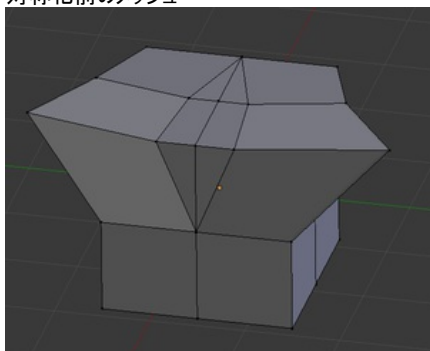
対称化 ツールは左右対称のメッシュを作る手っ取り早い方法です。このツールはオブジェクトのピボットでメッシュを切り、指定軸を中心にしてジオメトリを鏡映コピーし、二つを結合します(二つが繋がっている場合)。

方向(Direction)

効果の軸と方向を指定します。軸には3軸どれでも選べます。方向は正から負、または負から正のいずれかです。



対称化前のメッシュ



対称化後のメッシュ

Mirroring Geometry

(ジオメトリのミラー)

ミラーツールを使うと、ある軸を中心にジオメトリを反転させることができます。詳細は [Mirror](#) をご覧ください。

Vertex Tools

(頂点ツール)

このページでは Mesh(メッシュ) » Vertices(頂点)メニューにあるたくさんのツールを扱います。主に選択した頂点に対して働くツールですが、選択した辺や面に働くものもあります。

Merging

(結合)

Merge Vertices

(頂点結合)

モード: 編集モード

ホットキー: AltM

メニュー: Mesh(メッシュ) » Vertices(頂点) » Merge(結合)..., Specials(スペシャル) » Merge(結合) か Vertex Specials(頂点スペシャル) » Merge(結合)

選択している頂点のうち一つを残して他を削除することで、結合します。残る頂点の位置を、実行前のポップアップメニューから選ぶことができます。

最初に選択した頂点に (At First)

頂点選択モードでのみ利用でき、残る頂点を最初に選択した頂点の位置に置きます。

最後に選択した頂点に (At Last)

選択モードでのみ利用でき、残る頂点を最後に選択した頂点 (=アクティブなもの) の位置に置きます。

中心に (At Center)

すべての選択モードで利用でき、残る頂点を選択部分の中央に置きます。

カーソル位置に (At Cursor)

すべての選択モードで利用でき、残る頂点を 3Dカーソルにあわせます。

束ねる (Collapse)

残る頂点が1つ以上になる可能性のある、特殊なオプションです。実際には選択要素の「島 (island、繋がった選択頂点のグループ)」の数だけ頂点が残ります。残る頂点の位置はそれぞれの「島」の中央になります。Mesh(メッシュ) » Edges(辺) » Collapse(束ねる) メニューからも利用可能です...

頂点結合はもちろん、辺や面の一部も削除します。ただ、Blender は、再結合に部分的にしか関わらない辺や面をできる限り維持しようとしています。

AutoMerge Editing

(自動結合)

モード: 編集モード

メニュー: Mesh(メッシュ) » AutoMerge Editing(重複頂点の自動結合)

メッシュメニューにある、有効/無効の切り替えができるオプションです。有効にすると、頂点を動かす際に 重複のしきい値 (Double Threshold) 設定 (メッシュツールパネルにあります。下記参照) より近くに他頂点が近づくと自動的に結合されます。

Remove Doubles

(重複削除)

モード: 編集モード

パネル: Mesh Tools(メッシュツール)

ホットキー: W » 4 または CtrlV » Remove Doubles(重複頂点を削除)

メニュー: Mesh(メッシュ) » Vertices(頂点) » Remove Doubles(重複頂点を削除) または Specials(スペシャル) か Vertex Specials(頂点スペシャル) にある同名メニューから

重複頂点を削除はメッシュの単純化に役立つツールで、指定距離より近い頂点同士を結合します。メッシュの単純化方法としては [ポリゴン数削減モディファイア\(Decimate\)](#) を使うこともできます。

結合距離 (Merge Distance)

結合される距離のしきい値を決めます。単位は Blender Unit です。

非選択部 (Unselected)

選択頂点が非選択の頂点と結合できるようにします。無効化すると、選択頂点は他の選択頂点とだけ結合します。

Separating

(分離)

Rip

(切り裂き)

モード: 編集モード

ホットキー: V

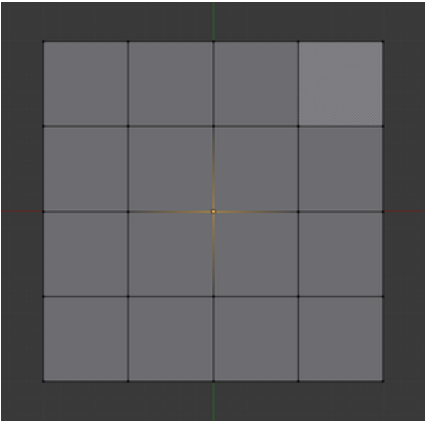
メニュー: Mesh(メッシュ) » Vertices(頂点) » Rip(切り裂き)

切り裂きツールは選択した頂点/辺を、選択していない隣の頂点と繋げたまま複製します。新しい辺は裂け目の一方の側の境界線に、古い辺はもう一方の側の境界線になり、メッシュには穴が開きます。

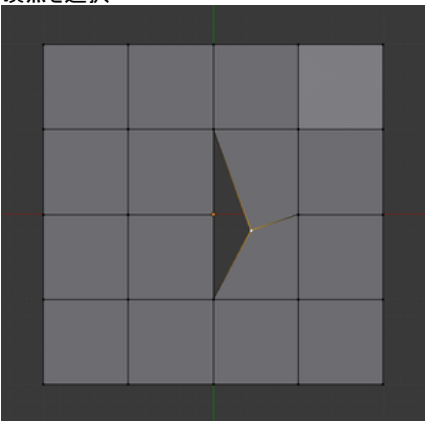
Blender 2.63 以降では複数の切り裂きの一斉実行、扇型の中心頂点の分離ができるようになりました。

Examples

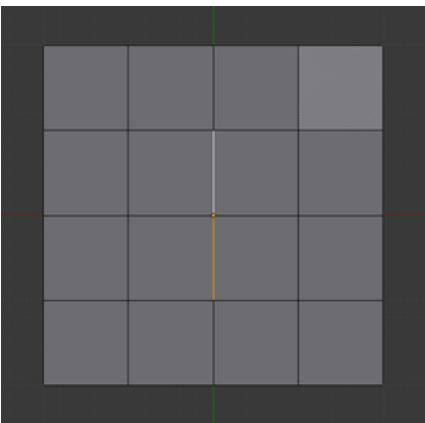
(例)



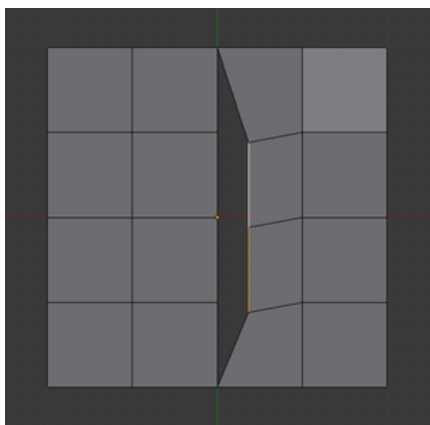
頂点を選択



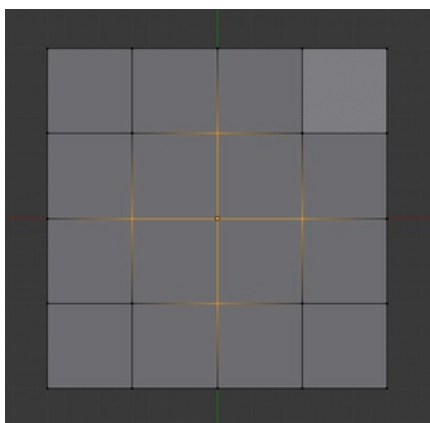
頂点を切り裂いたあとにできた穴



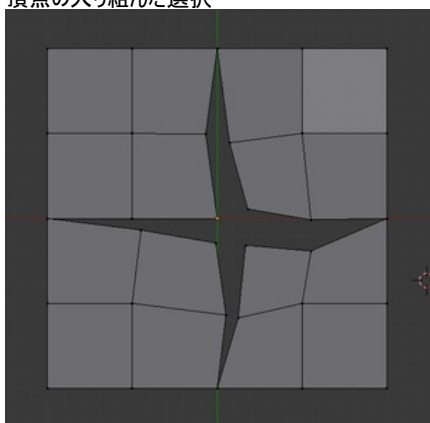
辺を選択



辺の選択をもとに切り裂いた結果



頂点の入り組んだ選択



切り裂き処理の結果

Limitations

(制限事項)

切り裂きツールは辺/頂点が選択されていなければ機能しません。(潜在的なものも含めて) 選択要素に面が含まれるとエラーメッセージ“Can't perform ripping with faces selected this way”が返ります。2つの面の間にない(manifold) 辺や頂点を選択していると失敗し、エラーメッセージ“no proper selection or faces include”が返ります。

Rip Fill

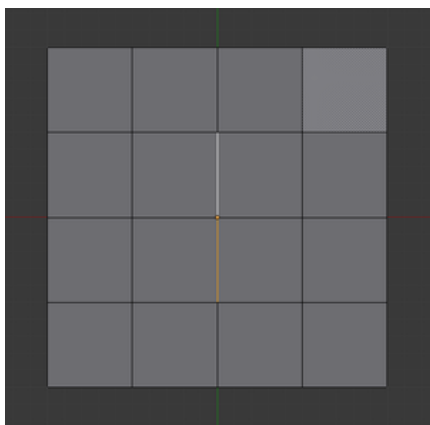
(切り裂きフィル)

モード: 編集モード

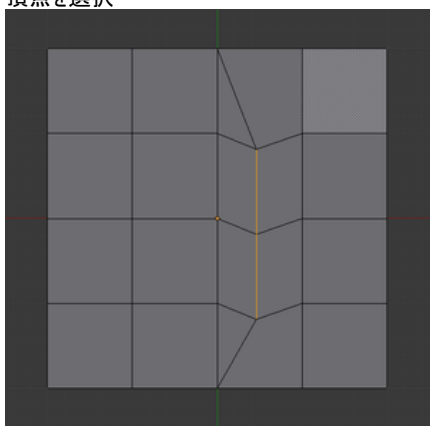
ホットキー: AltV

メニュー: Mesh(メッシュ) » Vertices(頂点) » Rip Fill(切り裂きフィル)

切り裂きフィルは上述の切り裂きと同じように動作しますが、穴を開けたままにせず、ジオメトリで埋めます。



頂点を選択



切り裂きフィルの結果

Split

(分離)

モード: 編集モード

ホットキー: Y

メニュー: Mesh(メッシュ) » Vertices(頂点) » Split(分離)

極めて特殊なツールで、選択要素の複製を作り、もし非選択要素がどれも元のデータを使っていなければ、元のデータを削除します。つまりメッシュから辺を分離したとき、元の辺はそれが他と一切繋がっていない場合を除いて、残されるということです。面を分離すると、元の面そのものは削除されますが、その辺や頂点は変更されずに残される、といった具合です。

複製が元の辺とまったく同じ場所に残されることに注意してください。はっきり見るためには移動させる (G) 必要があるでしょう。

Separate

(別オブジェクトに分離)

モード: 編集モード

ホットキー: P

メニュー: Mesh(メッシュ) » Vertices(頂点) » Separate(別オブジェクトに分離)

選択要素を別のメッシュオブジェクトに分けます。詳しくは [こちらで説明されています](#)。

Vertex Connect

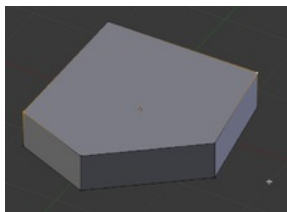
(頂点の連結)

モード: 編集モード

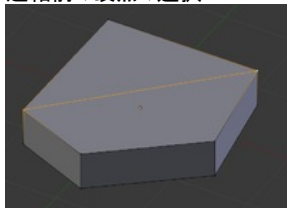
ホットキー: J

メニュー: Mesh(メッシュ) » Vertices(頂点) » Vertex Connect(頂点の連結) または CtrlV » Vertex Connect(頂点の連結)

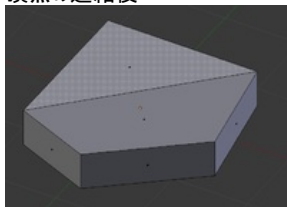
頂点の連結は、面内の2つの頂点を辺で繋ぎます。



連結前の頂点の選択



頂点の連結後



頂点結合操作でできた2つの面

Vertex Slide



(頂点スライド)

モード: 編集モード

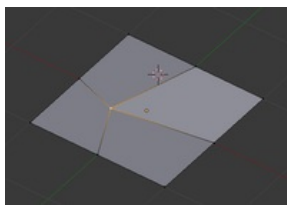
パネル: Mesh Tools (メッシュツール)

ホットキー: ⇧ ShiftV » Vertex Slide (頂点スライド) (Blender 2.66 以降は G → [Template:Shortcut](#) も利用できます)

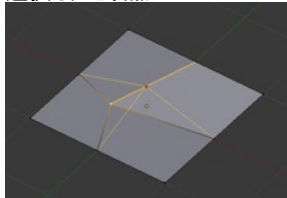
メニュー: Mesh (メッシュ) » Vertices (頂点) » Vertex Slide (頂点スライド) または CtrlV » Vertex Slide (頂点スライド)

頂点を、頂点に繋がった辺に沿って滑らすように移動します。ツールを開始するには ⇧ ShiftV を押してください。Blener 2.66 以降ではトランスフォームツールの一つとして G → [Template:Shortcut](#) でも利用できます。マウスを動かして目的の辺をハイライトし、LMB  で確定します。その辺から生じた直線に沿ってカーソルをドラッグし、位置を指定します。再度 LMB  を押して頂点の移動を完了します。次のキーを押し続けると、3つのオプションが利用できます:

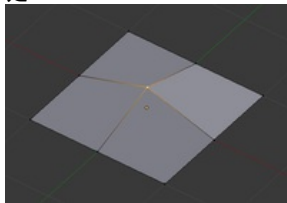
- 中点にスナップ ⇧ Shift
- 終点にスナップ Alt
- 終点にスナップして結合 Control



選択された頂点



インタラクティブな頂点の位置指定



再配置された頂点

Smooth

(スムーズ)

モード: 編集モード

パネル: Mesh Tools(メッシュツール)

ホットキー: CtrlV » Smooth(頂点スムーズ)

メニュー: Mesh(メッシュ) » Vertices(頂点) » Smooth Vertex(頂点スムーズ) または Specials(スペシャル) か Vertex Specials(頂点スペシャル) の同名メニュー

[スムーズツール](#) を一回適用します。

Make Vertex Parent

(頂点ペアレント作成)

モード: 編集モード

ホットキー: CtrlP

メニュー: Mesh(メッシュ) » Vertices(頂点) » Make Vertex Parent(頂点ペアレント作成)

他の選択オブジェクトを、選択頂点/辺/面の子要素にします。詳しくは [こちらをご覧ください](#)。

Add Hook

(フック)

モード: 編集モード

ホットキー: CtrlH

メニュー: Mesh(メッシュ) » Vertices(頂点) » Add Hook(フック)

選択要素にリンクした [フックモディファイア\(Hook\)](#) を追加します(新しいエンプティが現在選択しているオブジェクトを使います)。なお、この操作は履歴(history)メニューに現れるにも関わらず、編集モードでは取り消しできません – おそらく他オブジェクトが絡んでいるためです...

Blend From Shape/Propagate Shapes

(シェイプから混合/シェイプの伝播)

モード: 編集モード

ホットキー: W » AltBlend from shape(任意のシェイプキーを選択部に合成) または CtrlV » Blend From Shape(任意のシェイプキーを選択部に合成)、および W » AltShape propagate(形状伝播) または CtrlV » Shape Propagate(形状伝播)

メニュー: (Vertex) Specials((頂点)スペシャル) » Blend From Shape(任意のシェイプキーを選択部に合成) および Vertex Specials(頂点スペシャル) » Shape Propagate(形状伝播)

[シェイプキー \(eng\)](#) に関するオプションです。

Make Edge/Face

(辺/面の作成)

モード: 編集モード

ホットキー: F

メニュー: Mesh(メッシュ) » Edges(辺) » Make Edge/Face(辺/面作成)

選択要素に応じて辺か面を作ります。このツールは [編集の基礎のページ](#) で説明されています。

Set Edge Attributes

(辺の属性指定)

他のいくつかのツールがメッシュにどのように影響するかを決める、様々な属性を持つことができます。

Mark Seam and Clear Seam

(シームをつける/クリア)

モード: 編集モード(頂点 か 辺 選択モード)

ホットキー: CtrlE1 NumPad および CtrlE2 NumPad

メニュー: Mesh(メッシュ) » Edges(辺) » Mark Seam(シームを付ける)/Clear Seam(シームのクリア) (または Edge Specials(辺スペシャル) メニューの同じオプション)

シーム(Seam)とは UVマップ上に裂け目「島(island)」を作るための方法です。詳細は [UVテクスチャの章](#) をご覧ください。このコマンドは選択した辺にフラグをつけたり消したりします。

Mark Sharp and Clear Sharp

(シャープをつける/クリア)

モード: 編集モード(頂点 か 辺 選択モード)

ホットキー: CtrlE1 NumPad および CtrlE2 NumPad

メニュー: Mesh(メッシュ) » Edges(辺) » Mark Sharp(シャープを付ける)/Clear Sharp(シャープのクリア) (または Edge Specials(辺スペシャル) メニューの同じオプション)

シャープ(Sharp) フラグは、円滑化技法の一部である [辺分離\(EdgeSplit\)モディファイア](#) で使われます。シームと同じく辺のプロパティです。このコマンドは選択した辺のフラグをつけたり消したりします。

Adjust Bevel Weight

(ベベルウェイトの調整)

モード: 編集モード(頂点 か 辺 選択モード)

ホットキー: Ctrl⇧ ShiftE

メニュー: Mesh(メッシュ) » Edges(辺) » Adjust Bevel Weight(ベベルウェイト)

0.0 から **1.0** までの値をとる、辺のプロパティです。辺のベベルの強度を制御するために [ベベル\(Bevel\)モディファイア](#) で使われます。ツールを実行するとインタラクティブモードになり(トランスフォームツールに少し似ています)マウスを動かすと(またはキーボードから値を入力すると)選択した辺のベベルの(平均)強度を決められます。

Crease SubSurf

(クリース)

モード: 編集モード(頂点 か 辺 選択モード)

ホットキー: ⇧ ShiftE

メニュー: Mesh(メッシュ) » Edges(辺) » Crease SubSurf(クリース)

0.0 から **1.0** までの値をとる、辺のプロパティです。細分化されたメッシュで辺の鋭さを制御するために、[細分化\(Subsurf\)モディファイア](#) で使われます。ツールを実行するとインタラクティブモードになり(トランスフォームツールに少し似ています)マウスを動かすと(またはキーボードから値を入力すると)選択した辺のクリースの(平均)値を決められます。

Mark Freestyle Edge/Clear Freestyle Edge

(Freestyleの辺を付ける/クリア)

モード: 編集モード(頂点 か 辺 選択モード)

ホットキー: Ctrl⇧ ShiftE

メニュー: Mesh(メッシュ) » Edges(辺) » Mark Freestyle Edge (Freestyle辺を付ける)/Clear Freestyle Edge (Freestyle辺をクリア)

Freestyleの辺(Freestyle Edge) フラグは、Freestyle を使ったレンダリングで、レンダリング対象とする辺の指定に使われます。詳細は [Freestyleのラインセット > 辺マーク](#) のページをご覧ください。

Edge Slide

(辺をスライド)

モード: 編集モード (頂点 か 辺選択モード)


ホットキー: CtrlE » 6 NumPad (Blender 2.66以降ではG → G も利用できます)

メニュー: Mesh(メッシュ) » Edges(辺) » Slide Edge (辺をスライド) (または Edge Specials (辺スペシャル) メニューにある同じオプション)

隣接する面の間を横切るように、1つ以上の辺をスライドします。

Blender 2.66でトランスフォームモードの一つとして書きなおされ、G → G でスライドできるようになりました。Blender の古いバージョンでは、操作対象にできる辺に若干の制限があります(下記参照)。

均一 (Even) E


ループ辺の形状を、隣接するループ辺に合わせます。F を使うと反転して反対の頂点を合わせます。制御対象の辺を変えるには Alt Wheel  を使います。

反転 (Flip) F

均一モードが有効なら、ループ辺の形状を反対側の隣接ループ辺に合わせます。

LMB  は変更を確定し、RMB  か Esc でキャンセルします。

3Dビューのフッターおよび ツールシェルフ(確定後)には、このツールの係数が表示されます。精密制御するために -1 から 1 までの数値で入力できます。

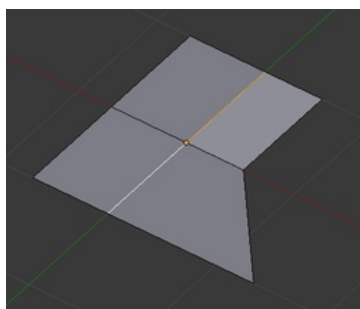
プロポーションナル (Proportional) モードでは、Wheel  や ← と → でプロポーションの計算に使う選択辺を変えられます。

Ctrl か ⇧ Shift を押し続けるとスライドを精密操作します。Ctrl は移動量の 10% の刻み幅でスナップし、⇧ Shift は 1% の刻み幅でスナップします。デフォルトは 5% の刻み幅です。

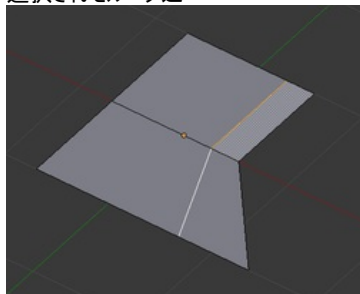
Usage

(使い方)

デフォルトでは、ループ辺上の頂点は、辺の長さに関わらず、元の位置から隣にあるループ辺までの距離に比例して動きます。



選択されたループ辺



位置を決め直したループ辺


Even mode

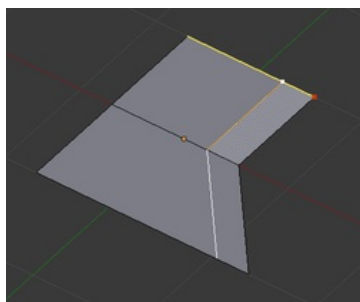
(均等モード)

均等モードでは、直交する各辺に対して一定比率ずつスライドさせるのではなく、選択されたループ辺が隣接するループ辺と同じ形状になるようにスライドします。

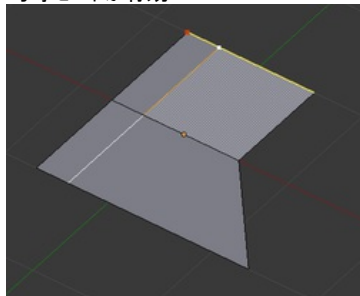
均等モードでは、大きな赤いマーカーで描かれる頂点から、現在選択中の黄色の辺の長さ方向に位置を示します。スライド中のループ辺の動きはこ

の長さに制限されます。マウスを動かすと、ヘッダーにはこの辺上の位置が表示されます。

ループ辺の位置計測の基準になる辺を別のものにするには、Alt Wheel  をスクロールします。



均等モードが有効



反転が有効化された均等モード

マウスを動かすと選択中の辺ループを開始頂点に近づけたり離したりしますが、ループ線は境界を接するループ辺のひとつと形を一致させるため、現在選択中の辺の長さまでしか動きません。

Limitations & Workarounds

(制限事項と回避策)

※ Blender 2.66 以降ではこれらの条件に当てはまってもエラーにならず、行おうとした操作が自動解釈されます
 選択した辺が下記の条件に当てはまる場合、操作は無効です:

それ自身に交差するループ

選択した辺の隣に、ふさわしい面が見つからなかった場合。例えば、同じ面を共有する2つの辺を選んだ場合です。面はそれ自身に隣接できません。

複数のループ辺

選択された辺が同じループ辺に属していない場合、つまり共通の辺を持たない場合です。このエラーを最小限に抑えるには、常に端から端まで辺を選択するか、ループを「チェーンで」選択します。複数の辺を選択するときは、それらが繋がっていることを確かめてください。これで延々とエラーが出る可能性を減らせるでしょう。

境界にある辺

片面のオブジェクトの辺が一つ選択された場合。面が1つしかないので、ループ辺を見つけられません。ループ辺とは、2つ以上の面に広がったループであるからです。

一般に、複数の辺を選択するときは、端から端まで、切れ目のない一連の流れを作るようにつなぐべきです。文字通りこれは一般的なルールで、一連の流れになってはいても、その中の一部の辺が別のループ辺に属している場合、不正な辺を選んでしまう可能性がまだあります。

Rotate Edge

(辺を回転)

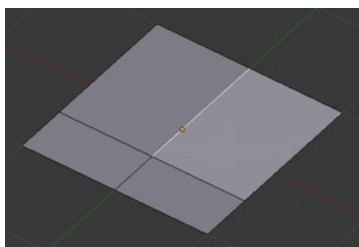
モード: 編集モード (頂点 か 辺選択モード)

ホットキー: CtrlE » Rotate Edge CW(辺を時計回りに回転) および CtrlE » Rotate Edge CCW(辺を反時計回りに回転)

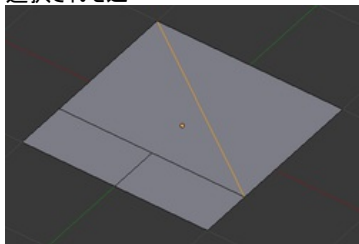
メニュー: メッシュ » 辺 » 辺を時計回りに回転 (Rotate Edge CW)/辺を反時計回りに回転 (Rotate Edge CCW)

「辺を時計回り/反時計回りに回転」は、二面間にある辺の頂点位置を、面の頂点位置を順番に巡るように回します。これはメッシュのトポロジーを再構築する際とても役立ちます。このツールは、選択状態の1辺、選択状態の2つの頂点、または間にある辺を共有している、選択状態の2つの面だけに実行できます。

(訳注: Blender 2.63 で、複数の辺の同時回転ができるようになりました)



選択された辺



時計回りに回転した辺

Using Face Selection

(面の選択を使う)

面の選択をもとにして辺を回転するには、「隣接した2つの面」を選択する必要があります。そうしないと次のエラーメッセージが出ます: "ERROR: Could not find any select edges that can be rotated(選択された辺には回転できるものが見つけれませんでした)". 2つの面の選択から 辺を時計回りに回転 と 辺を反時計回りに回転 のどちらを使うにしても、2つの面の共通の辺を選択したのとまったく同じ結果を作ることができます。

Delete Edge Loop

(ループ辺の削除)

モード: 編集モード (頂点 か 辺選択モード)

ホットキー: X/Del » 7 NumPad

メニュー: Mesh(メッシュ) » Edges(辺) » Delete Edge Loop(ループ辺の削除) (または 削除 メニューのEdge Loop(ループ辺) オプション)

ループ辺の削除を使うと、選択しているループ辺が他の2つのループ辺の間にあれば、削除できます。2つ存在していたループ面が一つのループ面に置き換わることになります。

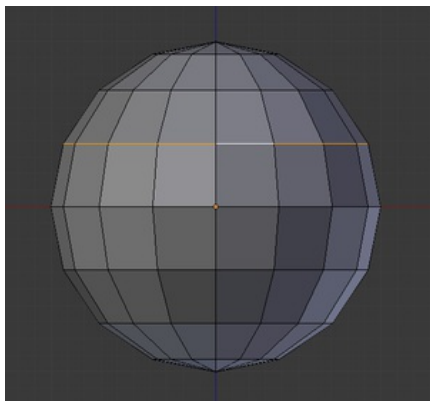
註記

仮にループ辺に似た辺を削除するのだとしても、ループ辺の削除機能を使うのとは大きな違いがあります。ループ辺を削除したときは、メッシュ表面を保持しようとして、周囲の面がお互いにつながります。一連の辺を削除したときは、辺が取り除かれ、周囲の面も同様に削除されます。この結果、メッシュの、面のあった場所は穴があいたままになります。

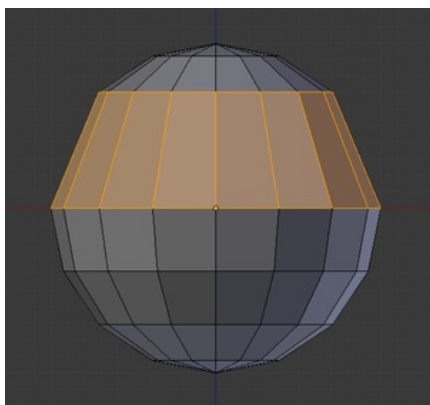
Example

(例)

UVスフィア上のループ辺を選択して削除したところ、ループ辺を挟んでいた辺が結合されて面を作りました。もし 削除 メニューから 辺 を選んで辺を削除していたら、代わりに面が削除されて、球を一周する帯状の空白ができていたでしょう。



選択されたループ辺



削除されたループ辺

Collapse

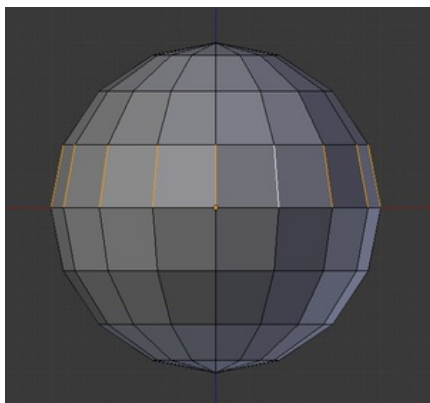
(辺を束ねる)

モード: 編集モード

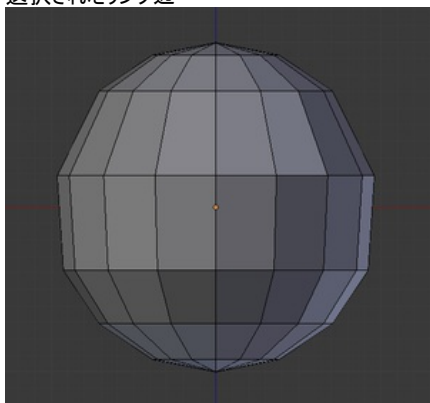
ホットキー: AltM » 3 NumPad

メニュー: Mesh(メッシュ) » Delete(削除) » Edge Collapse(辺を束ねる)

選択した辺それぞれに対して、辺の2頂点を結合して1つにします。リング辺を選択して束ねると、そこを通るループ面を削除できるので便利です。



選択されたリング辺



束ねられたリング辺

Edge Split

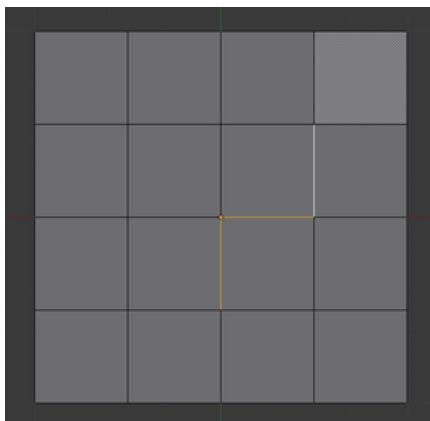
(辺の分離)

モード: 編集モード

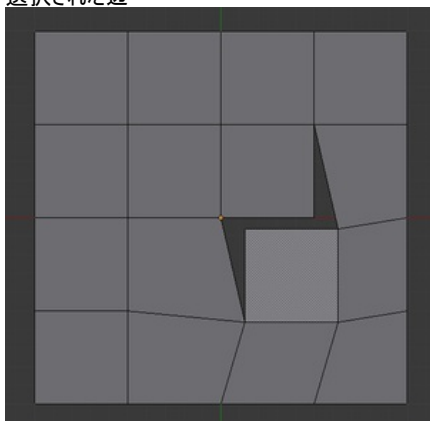
ホットキー: CtrlE » 辺の分離 (Edge Split)

メニュー: Mesh(メッシュ) » Edges(辺) » Edge Split(辺の分離)

辺の分離は切り裂き(rip)ツールに似ています。メッシュ内部の辺を2つ以上連続して選ぶか、端にある辺を選んだとき、辺の分離を使うと穴ができ、穴の境界線を形作るように、選択していた辺が複製されます。



選択された辺



分離して隣の面を動かすと左に穴が開きます

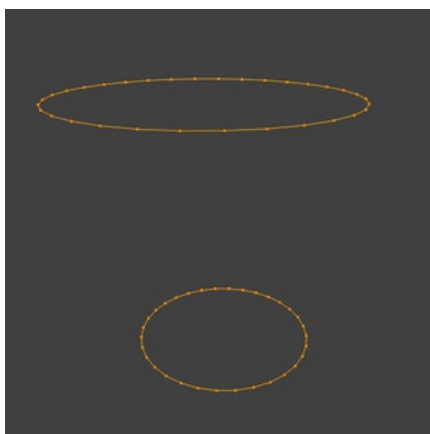
Bridge Edge Loops

(辺ループのブリッジ)

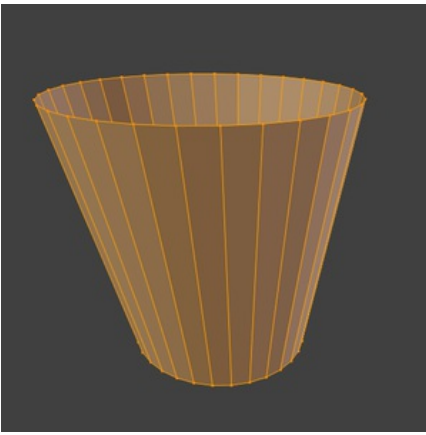
モード: 編集モード

メニュー: Mesh(メッシュ) » Edges(辺) » Bridge Edge Loops(辺ループのブリッジ)

Bridge Edge Loops は面を使って複数の辺ループをつなげます。
二つの閉じた辺ループによる簡単な例。

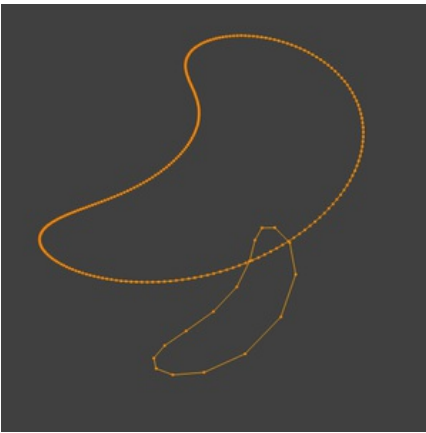


入力

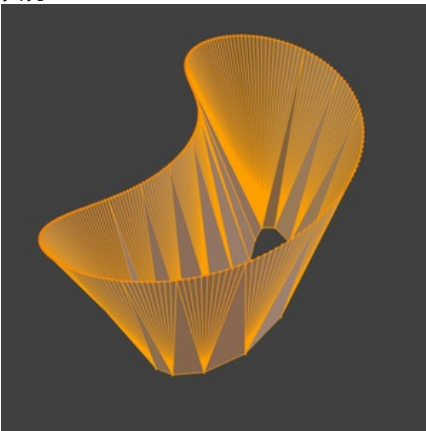


Bridge の結果

頂点数の異なる辺ループ間のブリッジ例。

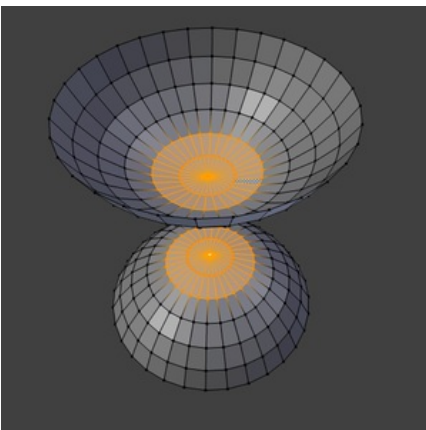


入力

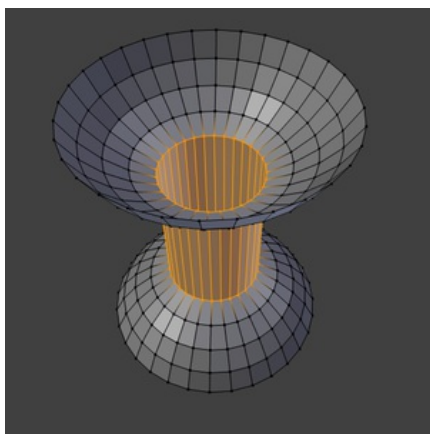


Bridge の結果

ブリッジツールで選択面に穴を空け、つなげる例。

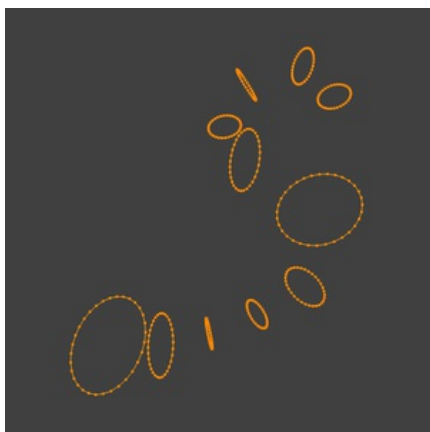


入力

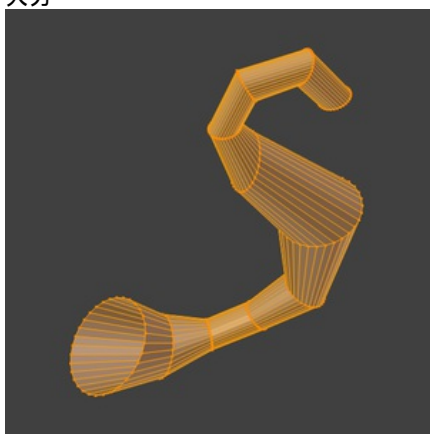


Bridge の結果

ブリッジツールが1ステップで複数のループを検知し、loft する様子を示す例。

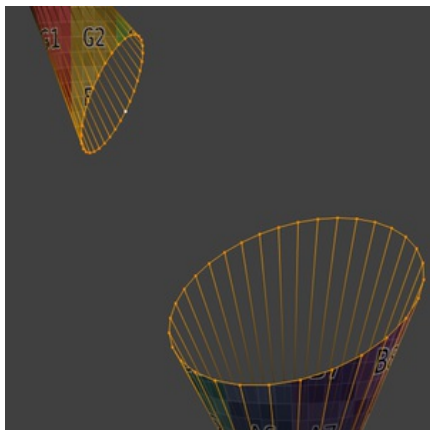


入力

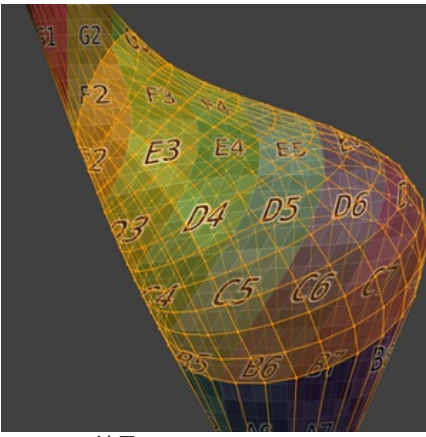


Bridge の結果

細分化 (subdivision) オプションおよびUVの混ざったサーフェスの例。



入力



Bridge の結果

Face Tools

(面ツール)

面を操作するツールです。

Creating Faces

(面の作成)

Make Edge/Face

(辺/面の作成)

モード: 編集モード

ホットキー: F

メニュー: Mesh(メッシュ) » Faces(面) » Make Edge/Face(辺/面作成)

選択要素に応じて辺や面を作ります。[基本的な編集](#)ページに詳細があります。

Fill

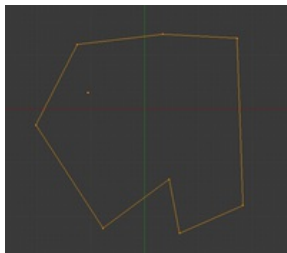
(面張り)

モード: 編集モード

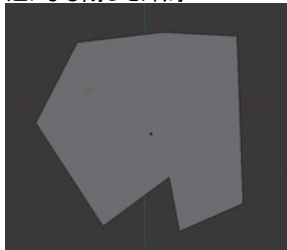
ホットキー: AltF

メニュー: Mesh(メッシュ) » Faces(面) » Fill/Beautify Fill(面を張る)

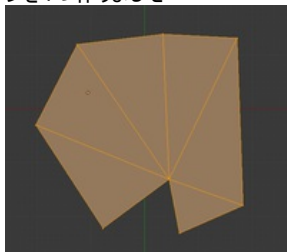
選択された辺か頂点のグループのいずれかが 1つ以上の完全な外周を作る場合にのみ、そのグループから 三角形の面を作ります。



辺による閉じた外周

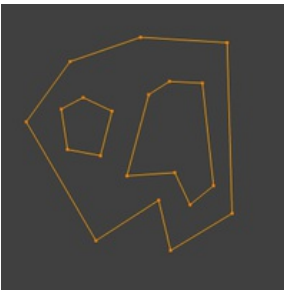


ショートカットFによる面張り。n-ゴンを1つ作りました



AltFによる面張り

なお、n-ゴンの作成と違い、Fill は穴をサポートします。



穴のある閉じた外周



fillAltF の実行後

Beauty Fill

(面を最適化)

モード: 編集モード

ホットキー: Alt⇧ ShiftF

メニュー: Mesh(メッシュ) » Faces(面) » Fill/Beautify Fill(面を最適化)

面を最適化は、既存の面を選択したときにだけ働きます。選択された複数の三角形をもっと「バランスのとれた」ものに(長細い三角形が少なくなるように)並べ替えます。



メッシュに変換されたテキスト



面を最適化Alt⇧ ShiftFの結果

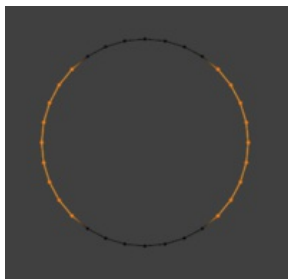
Grid Fill

(グリッド埋め)

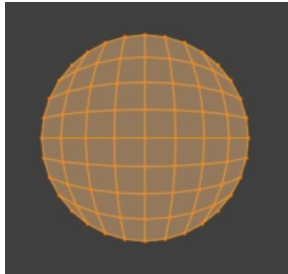
モード: 編集モード

メニュー: Mesh(メッシュ) » Faces(面) » Fill/Grid Fill(グリッドフィル)

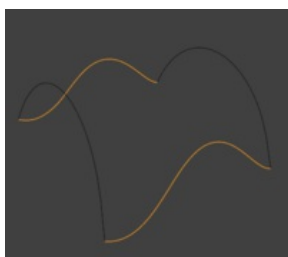
Grid Fill は、繋がった1組の辺ループを使って、周囲のジオメトリにならうグリッドで穴埋めします。



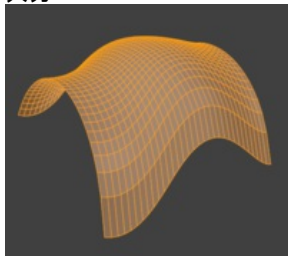
入力



Grid fill の結果



入力



Grid fill の結果

Convert Quads to Triangles

(四角面を三角面に)

モード: 編集モード

ホットキー: CtrlT

メニュー: Mesh(メッシュ) » Faces(面) » Convert Quads to Triangles(四角面を三角面に) または Face Specials(面スペシャル) » Triangulate(面を三角化)

名前の通り、選択された四角面をそれぞれ、2つの三角面に変換します。四角面は2つの三角面のセットであることを思い出してください。

Convert Triangles to Quads

(三角面を四角面に)

モード: 編集モード

パネル: Mesh Tools(メッシュツール)

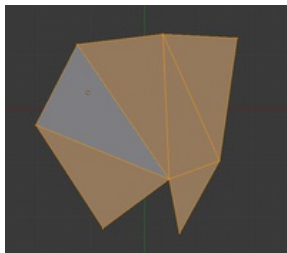
ホットキー: AltJ

メニュー: Mesh(メッシュ) » Faces(面) » Convert Triangles to Quads(三角面を四角面に)

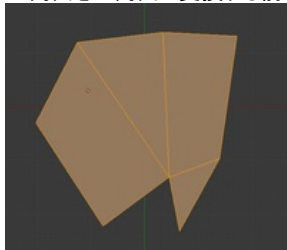
選択された三角面を四角面に変換します。具体的には、しきい値に基づき、隣り合う三角面の共通の辺を取り除いて四角面に変換します。このツールは複数の三角面を選択したときにだけ実行できます。

2つの三角面にこのツールを実行した結果は、その2つの三角面を選択してショートカット F で面を作るか、共通の辺を選んでショートカット X » 溶解(Dissolve) で溶け合わせた結果と同じです。

このツールで四角面を作るには、少なくとも2つの隣接した三角面が必要です。選択した三角面の数が偶数のときは、結果が四角面だけにならない可能性もあります。実際のところ、このツールは指定した三角面から「正方形になるべく近い」四角面を作ろうとします。つまり三角面が残される可能性があるということです。



三角面を四角面に変換する前



最大角度 30 を指定して三角面を四角面に変換した後

メニューとホットキーはすべて、メッシュツール パネルにある設定を使います:

最大角度 (Max Angle)

0 から 180 までの値で、隣接する三角面にこのツールが作用するしきい値を調整します。しきい値が **0.0** なら、完全な長方形(斜辺を共有する2つの直角三角形)を作る隣接する三角面だけを繋げます。他の二辺に比べて短い辺を共有している三角面には、大きな値が必要です。

UVを比較 (Compare UVs)

有効にすると、三角面がアクティブな UVマップでも隣接していないと、結合されません。なお、動作しているオプションはこれだけのようです...

頂点色を比較 (Compare Vcol)

有効にすると、頂点色が一致しない三角面は結合されません。このオプションがどうやったら動作するのか不明です - 実際に動作するのかどうかも...

シャープを比較 (Compare Sharp)

有効にすると、「sharp」な辺を共有する三角面は結合されません。このオプションが動作しているのかどうかも、「sharp」領域が何を指すのかも不明です。Sharp フラグも、2つの三角面がなす角度も、ここでは影響をしていないようです...

マテリアルを比較 (Compare Materials)

有効にすると、同じマテリアル番号を使っていない三角面は結合されません。このオプションも動作していないようです...

Solidify

(厚みづけ)

モード: 編集モード

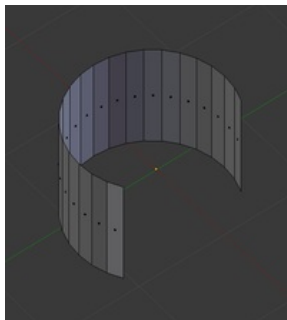
ホットキー: CtrlF » Solidify(厚みづけ)

メニュー: Mesh(メッシュ) » Faces(面) » Solidify(厚みづけ)

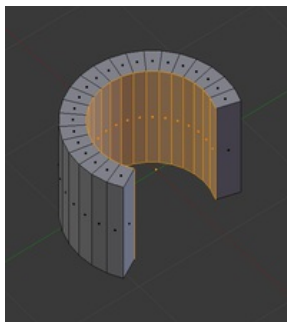
選択された面を均一に押し出し、非多様体の表面(a non-manifold surface)にボリュームを与えることで、厚みを与えます。[モディファイア](#)を使っても同じことができます。ツールの使用后、ツールパレットで厚みを設定できます。

厚み (Thickness)

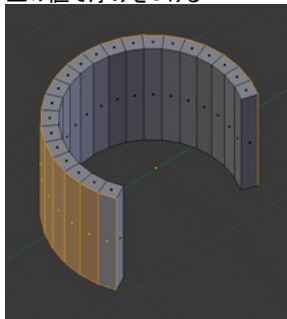
新たに作られた表面をずらす量です。法線を基準に、正の値はサーフェスを内側にずらします。負の値は外側にずらします。



厚みをつける前のメッシュ



正の値で厚みをつける



負の値で厚みをつける

Rotate Edges

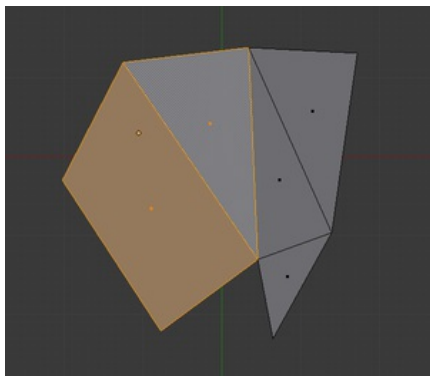
(辺を時計回りに回転)

モード: 編集モード

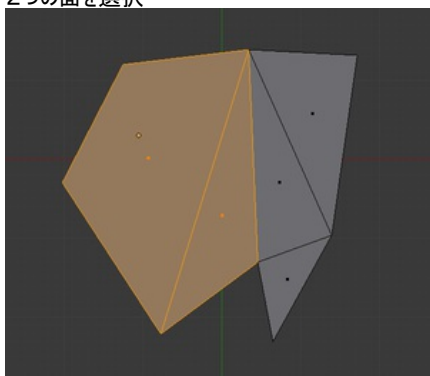
メニュー: Mesh(メッシュ) » Faces(面) » Rotate Edge CW(辺を時計回りに回転)

辺モードの辺の回転と同じ機能です。

2つの面が共有する辺に働き、辺が選択されていれば、その辺を回転させます。



2つの面を選択



辺の回転後

さらなる情報は [辺を時計回り/反時計回りに回転](#) をご覧ください。

Normals

(法線)

法線は主に面の「副産物」なので、そのオプションもここで述べます。

面の法線を扱うためのさらなる情報は [円滑化](#) をご覧ください。

Flip Direction

(方向の反転)

モード: 編集モード

ホットキー: W » Flip Normals (面を反転)}

メニュー: Mesh(メッシュ) » Normals(法線) » Flip(反転) または Specials(スペシャル) » Flip Normals(面を反転)

選択されたすべての面の法線の向きを単に反転します。なお、選択したものだけを反転するので、法線の向き(常に面に直交する「方向」とは違います)を正確に制御できます。

Recalculate Normals

(法線の再計算)

モード: 編集モード

ホットキー: CtrlN および ctrl

メニュー: Mesh(メッシュ) » 法線(Normals) » 面の向きを外側に揃える(Recalculate Outside) および Mesh(メッシュ) » Normals(法線) » Recalculate Inside(面の向きを内側に揃える)

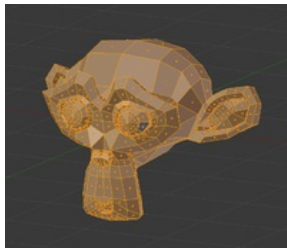
選択した面の法線を再計算して、面の属するボリュームの外側(または内側)を指すようにします。ボリュームは閉じている必要はありません。というより、このツールは少なくとも1つ、同じ平面上にない面と隣接している面にしか働きません。例えば グリッド(Grid) プリミティブの法線の向きは、面の向きを外側に揃えると 面の向きを内側に揃える のどちらを使っても変わらないでしょう。

ジオメトリの変形

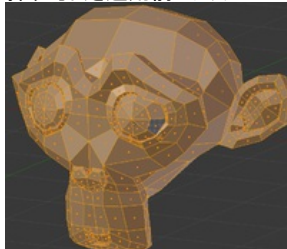
押す/引く

収縮/膨張と同じく、選択要素と選択要素の平均位置を結ぶ線上に沿って、すべての選択要素を動かします。移動量はどれも同じ値で、マウスでコントロールできます。拡大縮小にどこか似ていますが、もっと大きな変形です。

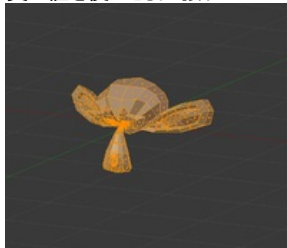
先述したものと違って、変形を軸に [ロック](#) できます (ただ、役立つのは「面ロック」だけでしょう...)



押す/引くを適用前のメッシュ



負の値を使った引っ張り



正の値を使った押し込み

湾曲 (Warp)

モード: 編集モード

ホットキー: \diamond ShiftW

メニュー: メッシュ/曲線/サーフェス » トランスフォーム » 湾曲

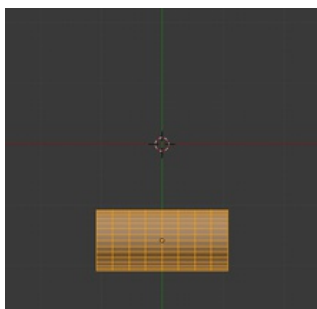
湾曲変形は非常に特定された状況で役立ちます。選択要素を 3Dカーソルに (ピボットの設定には関係しません) 巻きつけます。ここで、動作はビューに依存します。カーソルと縦方向に並ぶ点は同位置に留まります。各点のカーソルの横方向の位置との距離は、ツール実行後の角度に対応します。点はカーソルに巻きつけられ、360 を指定すると完全な円状に巻きつき、左右に一番離れていた2点とカーソル位置が直線上に並びます。

このツールを使うには、まずビューの円の中心にしたい箇所にカーソルを置いてください。ツールを実行したら、カーソルを動かすか、メッシュが巻きつく角度を入力します。

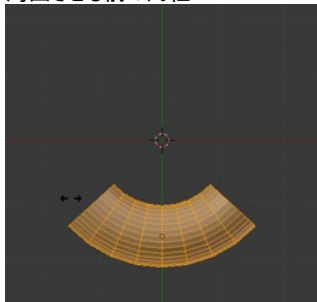
例

円柱を環状に湾曲します。

- 上面図に切り替えてメッシュを 3Dカーソルから離します。この距離は湾曲の半径を定義します。
- メッシュの編集モード (\Leftrightarrow Tab) に入り、A を押して全頂点を選択します。 \diamond ShiftW を押して湾曲変形ツールを実行します。マウスを左右に動かして、結果をプレビューしながら湾曲の度合いを決めます。



湾曲させる前の円柱



小さな角度で湾曲させた円柱



大きな角度で湾曲



せん断 (Shear)

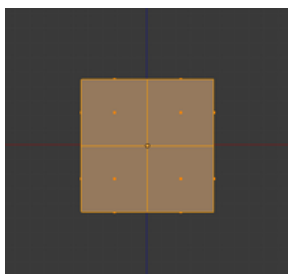
モード: 編集モード

ホットキー: CtrlAlt⇧ ShiftS

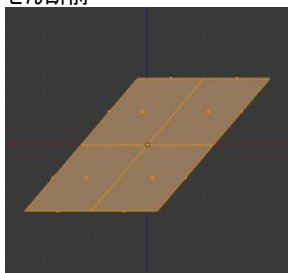
メニュー: オブジェクト/メッシュ/曲線/サーフェス » トランスフォーム » せん断

せん断変形は選択要素(編集モードでは、頂点/辺/制御点/...)にせん断を行います。他の変形ツールと同じように、ビュー空間を使い、ピボット位置を中央にして行われます。せん断はビューのピボットを通る x 軸に沿って行われます。この軸より「上」にあるもの(正の y 軸座標を持つもの)はすべてマウスポイントと同じ方向に(ただし常に x 軸と平行に)動かされます。この x 軸より「下」にあるものはすべて、反対の方向に動かされます。x 軸から離れた要素ほど多く動きます。

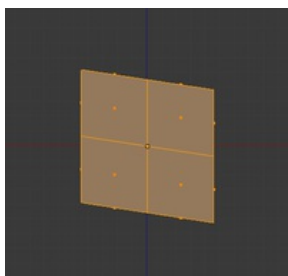
ツールを実行したら、マウスを左右に動かして結果プレビューを見ながらせん断を調整します。横方向ではなく縦方向に効果が現れるようにするには、MMB  クリックしてからマウスを上下に動かすか、または 0 以上の数値を入力します。ツールの変更を確定するには、LMB  を押します。



せん断前



横方向のせん断



縦方向のせん断

球へ変形 (To Sphere)

モード: 編集モード

パネル: メッシュツール (編集コンテキスト)

ホットキー: ⇧ ShiftAltS

メニュー: メッシュ/曲線/サーフェス » トランスフォーム » 球へ変形

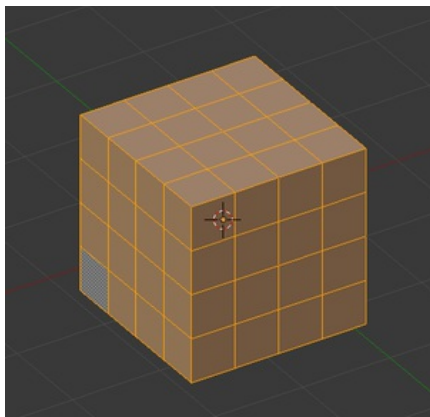
このコマンドは選択したメッシュ要素を「球状化」します。まず要素の平均位置が計算され、この点からの平均距離に向けて、選択要素を動かしていきます。値1を使うとすべての頂点がこの点から等間隔に置かれ、球形状になります。

ツールをアクティブにしたら、マウスを左右にドラッグして効果を確認しながら調整するか、0から1の範囲で値を入力して調整します。

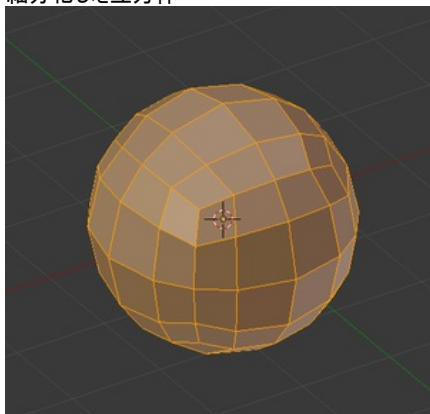
例

まずは [立方体](#) から初めましょう。

- ⇔ Tabを押して編集モードに切り替えます。
- Aを2度押して立方体の全頂点選ばれていることを確認します。それからF9を押して編集コンテキストに入ります。メッシュツールパネルが見えるようになったはずです。
- メッシュツールパネルで細分化ボタンを押すか、W » 細分化を使って立方体を細分化します。これを好きな回数行います。細分化するほど、作られる球は滑らかになります。
- ⇧ ShiftAltSを押してマウスを左右に動かし、結果プレビューを見ながら「球状化」の形を決めます(もしくは後述のように"1.000"のような値を直接入力しても同じ結果が得られます) – なるべく中点ピボットを使って!
- もしくは、球へ変形ボタン(メッシュツールパネルにあります)を使うこともできます。球を作るなら"100"を選んで球を作ってください。この場合は、3Dカーソルは動かさないほうがよいでしょう。動かすと球ではなく球の一部になります。



細分化した立方体



細分化した立方体、湾曲後

ミラー

モード: 編集モード


ホットキー: CtrlM

メニュー: メッシュ » ミラー » 希望の座標軸

ミラーツールは、選択要素を選択した軸に対してミラー反転します。

編集モードのミラーツールは[オブジェクトモードのミラー \(en\)](#)と同じです。これは頂点/辺/面を、選択したピボットを中心に指定軸方向に -1 だけ拡大縮小するのとまったく同じですが、素早く手軽に使えます。

ツールを実行したら、選択要素をミラー反転する軸を x、y、z のいずれかを入力して指定します。

MMB  を押し続けてミラー反転を行いたい方向にドラッグして、結果プレビューを見ながらジオメトリをミラー反転することもできます。

対称軸

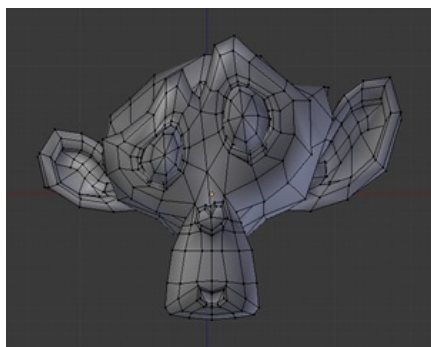
座標系ごとに、ミラー反転の軸を選ぶことができます。

ピボットはミラー反転の軸にしたい任意の場所に配置でき、軸に沿う座標系も選ぶことができます。可能性は無限で完全に自由です。

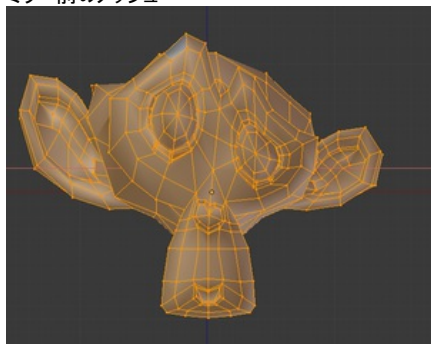
ピボット (Pivot point)

[ピボット](#) は先に設定されていなければなりません。ピボットは対象の中心になります。ミラーツールが使われている間、常にピボットの場所が表示されます。

図 ([中点を対象にミラー](#)) (訳注: ここにない画像かもしれませんが) では、ピボットに編集モードのデフォルトの [中点](#) が選ばれています。[ピボット](#) で述べられるように、これは編集モードの特殊なケースです。

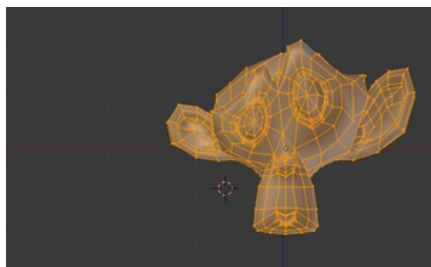


ミラー前のメッシュ

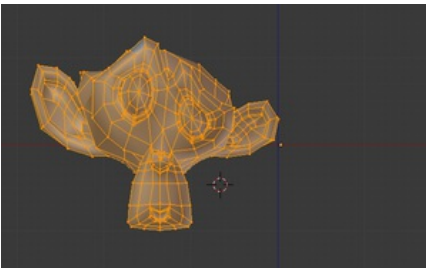


X軸に沿ってミラー適用後

図 ([X軸に沿い、3Dカーソルをピボットにしてミラー適用後](#)) では、ピボットに3Dカーソル、座標系にローカル (= オブジェクト空間) が選ばれ、変形の軸は X 軸です。



ミラー前のメッシュ



X軸に沿い、3Dカーソルをピボットにしてミラー適用後

座標系 (Transform Orientation)

[座標系 \(en\)](#) は、3D 領域のヘッダーのマニピュレーター ボタンの隣にあります。ミラーが従う座標系を決めます。

収縮/膨張 (Shrink/Fatten Along Normals)

モード: 編集モード

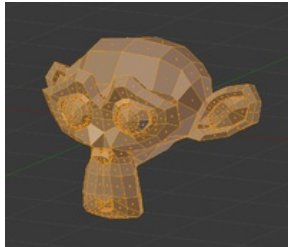
パネル: メッシュツール (編集コンテキスト)

ホットキー: AltS

メニュー: メッシュ » トランスフォーム » 収縮/膨張

このツールは選択した頂点/辺/面をそれ自身の法線(面に対する垂線)に沿って移動します。「標準的な法線のメッシュ」は収縮/膨張します。

このツールはピボットやトランスフォーム方向には関係しません。



収縮/膨張前のメッシュ



正の値を使って膨らませたところ



負の値を使って縮ませたところ

スムーズ(Smooth)

モード: 編集モード

パネル: メッシュツール (編集コンテキスト, F9)

ホットキー: CtrlV » 頂点スムーズ

メニュー: メッシュ » 頂点 » 頂点スムーズ

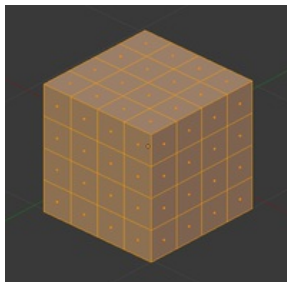
このツールは面同士の角度を平均化して、選択要素を滑らかにします。ツールの使用后、オプションがツールシェルフに現れます:

スムーズする回数 (Number of times to smooth)

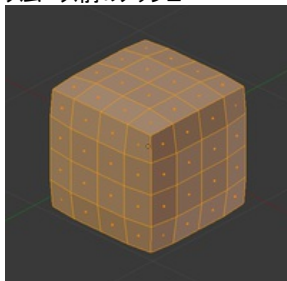
スムーズを繰り返す回数です。

軸 (Axes)

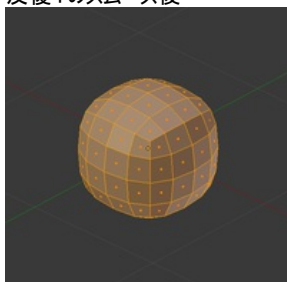
効果を特定軸に制限します。



スムーズ前のメッシュ



反復1のスムーズ後



反復10のスムーズ後

ラプラシアンスムーズ (Laplacian Smooth)

モード: 編集モード

ホットキー: W » ラプラシアンスムーズ

詳細は [ラプラシアンスムーズモディファイア \(Laplacian Smooth Modifier\) \(en\)](#) をご覧ください。

ラプラシアンスムーズはメッシュ形状全体をうまく保つような新しい円滑化アルゴリズムを使います。メッシュ操作として実行するか、非破壊のモディファイアを利用することができます。

註記

[スムーズモディファイア \(Smooth modifier\)](#) はスムーズツールの非破壊な代替に使えます。効果範囲を頂点グループに制限できます。

実際のスムーズとシェーディングのスムーズ

このツールと、[このページ \(en\)](#) に書かれているシェーディングスムーズオプションを間違えないようにしてください。別の動作をします！ このツールはメッシュ自身を編集し、鋭角を取り除きます。一方、Set Smooth/AutoSmooth はメッシュのシェーディング方法を決めて滑らかになる「魔法」をかけるだけで、メッシュは一切編集しません...

ノイズ

モード: 編集モード

パネル: メッシュツールパネル (編集コンテキスト)

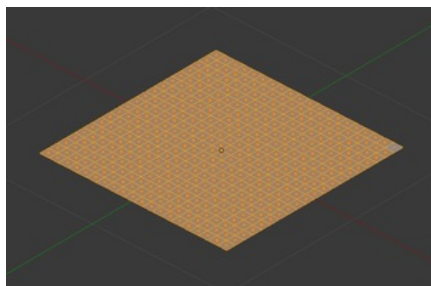
註釈
ノイズは古い機能です。[置換\(Displace\)モディファイア](#)はノイズツールの非破壊版の代替になり、この種の効果を理解するにはもつとすなおな方法です。モディファイアの優れた点は、いつでもキャンセルができること、ずれの量や方向を正確に制御できること、などなど…。ANT Landscape [アドオン](#)もご覧ください。

ノイズ機能を使うと、メッシュのマテリアルの、最初のスロットにあるテクスチャの灰色の値に基いて、メッシュにある頂点をずらすことができます。

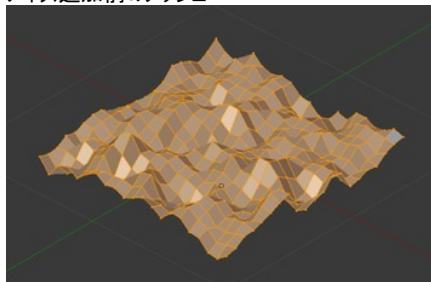
この機能が動作するためには、メッシュにマテリアルとテクスチャが割り当てられていなければなりません。テクスチャがマテリアルのプロパティに影響することを避けるには、テクスチャメニューで無効化することができます。

ノイズ機能はオブジェクトの±Z軸方向にのみ頂点をずらしします。

ノイズはマテリアルテクスチャに従って、永久的にメッシュを変更します。凹凸は現在のメッシュに追加されます。レンダリング時に効果を出すには、テクスチャをディスプレイメントにマップします。オブジェクト/編集モードではオブジェクトは通常通りに見えますが、レンダリングすると変形します。変形はマッピングパネルやテクスチャ自身のパネル(例:クラウド、マーブルなど)を編集して制御できます。



ノイズ追加前のメッシュ



基本的なクラウドテクスチャによるノイズ追加後のメッシュ

メッシュの複製ツール

この章では、既存のジオメトリを複製して新たにジオメトリを追加するツールを扱います。

- [ジオメトリの複製](#)
- [押し出し](#)
- [スピン \(en\)](#)
- [スクリュー \(en\)](#)

複数のビューポート

メッシュツールパネルにある複製ツールのどれかを使うと、複数ビューを開いている時は、Blender はどのビューで作業をすべきなのか判断できません。こうしたツールには対象ビューが重要なので、ツールを使い始めるとカーソルがクエスチョンマークのような形に変化します。ツールを使いたいウィンドウをクリックしてください。

複製 (Duplicate)

モード: 編集モード

ホットキー: ◊ ShiftD

メニュー: メッシュ » 複製

選択された要素を、メッシュの残りの要素には一切繋げずに(例えば押し出しとは違う動作です)複製し、オリジナルと同じ場所に置きます。複製が終われば、*新たに複製された要素だけが選択され*、自動的に移動モードに入るので、複製を好きな場所に移動できます。

ツールシェルフでは ベクトルオフセット、プロポーション編集、複製のモード(機能していない?)、軸の制限の設定ができます。

複製された要素はオリジナルと同じ[頂点グループ](#)に属します。同じ事が [マテリアルのインデックス](#)、辺の シャープやシームといったフラグ、それにその他の頂点/辺/面のプロパティにもおそらく言えるでしょう。

Extrude

(押し出し)

Extrude Region

(領域の押し出し)

モード: 編集モード

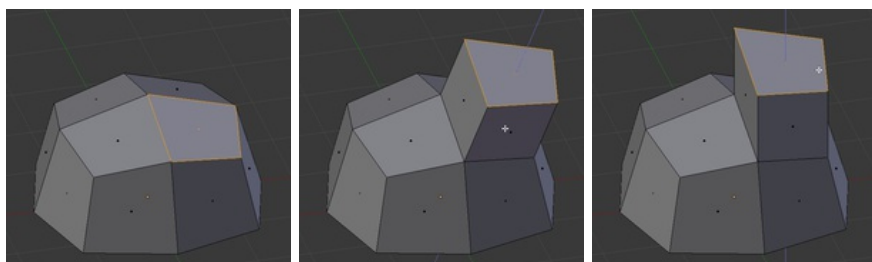
パネル: Mesh Tools (メッシュツール) » Extrude (押し出し)

ホットキー: E または AltE

メニュー: Mesh (メッシュ) » Extrude Region (領域の押し出し)

メッシュを扱うために最大級に重要なツールの一つが押し出し (Extrude) です。四辺形からは六面体を、円からは円柱を作ることができ、同様に簡単に樹の枝のようなものも作れます。押し出し (Extrude) は Blender で最も頻繁に使われるモデリングツールです。単純でわかりやすく、簡単に使えますが、非常に強力です。

選択された面は、選択された面すべての共通法線に沿って押し出されます。押し出しを単一の軸方向に制限することもできます。例えば押し出し中に X を使うと X 軸方向に制限でき、ShiftX を使うと YZ 平面に制限できます。面の法線に沿って押し出すとき、グローバルな Z 軸方向に制限するには、Z を二度押す必要があります。一回目で面の法線の Z 軸の制限を解除し、二回目でグローバルの Z 軸の制限を有効にします。



選択された面

押し出し中

Z 軸方向に制限

処理は非常に直感的ですが、以下に述べるように押し出しの裏にはかなり精巧な仕組みがあります:

- このアルゴリズムはまず、押し出しの「辺ループ」の外部がどこからなのかを決めます。押し出しの辺ループとは、選択した辺と辺の間にあり、押し出されて面が変わるものです。このアルゴリズムはデフォルトでは (後述) 2つ以上の選択面に属する辺は内部にあるので、ループの一部ではないとみなします
- 次に、辺ループ内の辺は面に変えられます
- 辺ループ内の辺がメッシュ全体のただ一つの面にだけ属していれば、選択したすべての面が複製され、新たに作られた面に繋がられます。例えば、矩形はこの段階で六面体になります。
- その他の場合には、選択した面は新たに作られた面に繋がられますが、複製はされません。これは、最終的にメッシュの「内部に」面を残したくないからです。これにより、押し出しを使うと常に論理的に閉じたポリウムが確実に作られるので、複製の有無の判断は非常に大切です。
- 完全に閉じたポリウム (例えば6面すべてを持つ立方体など) を押し出すとポリウムが複製されるので、単に元のポリウムと一切繋がっていない複製ができます。
- 選択した面に属せず、「開いた」辺ループにある辺は複製され、新たな辺と元の辺との間に新たな面が作られます。
- 選択した辺に属せず、単独で選択されている頂点は複製され、新たな辺がその間に作られます。

Extrude Individual

(個々に押し出し)

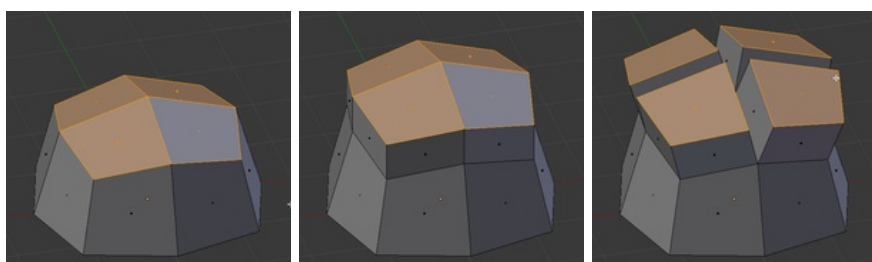
モード: 編集モード

パネル: Mesh Tools (メッシュツール) » Extrude Individual (個々に押し出し)

ホットキー: AltE

メニュー: Mesh (メッシュ) » Extrude Individual (個々に押し出し)

個々に押し出し (Extrude Individual) は複数の面を選択しているとき、領域としてではなく、面を個別に押し出します。面はそれぞれ、全体の平均の法線ではなく、その面の法線に沿って押し出されます。ここから言えることがいくつかあります。まず言えるのは、「内部の」辺 (選択した面同士の境界にある辺) はもう削除されないということです (元の面は削除されます)



複数の面の選択

領域の押し出し

個々に押し出し

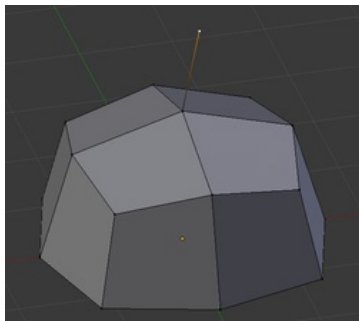
Extrude Edges and Vertices Only

(辺と頂点のみ押し出し)

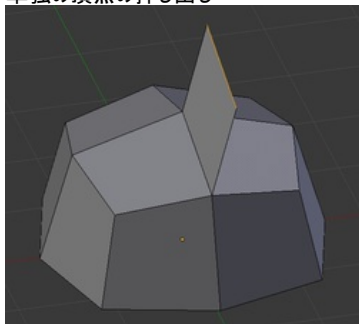
モード: 編集モード、頂点と辺

ホットキー: AltE

押し出し中に、辺や面の一部ではない頂点を選択されていると、この頂点も期待通り押し出され、非多様体の辺を作ります。同様に、面の一部ではない辺が選択されていれば、押し出されて面を作ります。



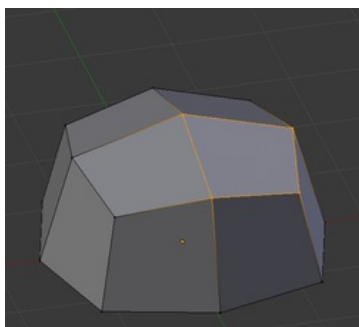
単独の頂点の押し出し



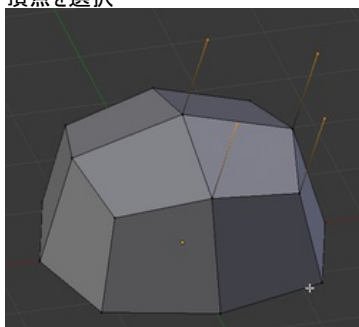
単独の辺の押し出し

選択した頂点が辺や面の形状になるときは、辺が選択されているかのように押し出されます。選択した辺が面の形状になるときも同様です。

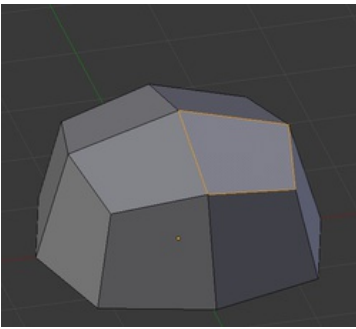
頂点や辺を選択して、頂点またはや辺だけを押し出したい場合は、AltEを使って 辺のみの押し出し (Edges Only) や 頂点のみの押し出し (Vertices Only) を呼び出してください。



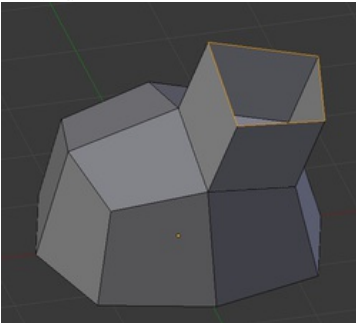
頂点を選択



頂点のみ押し出し



辺を選択



辺のみ押し出し

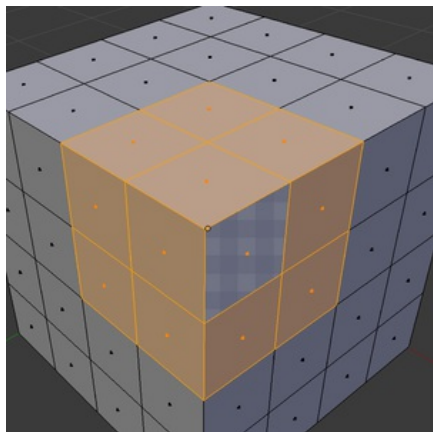
面の差し込み (Inset)

モード: 編集モード

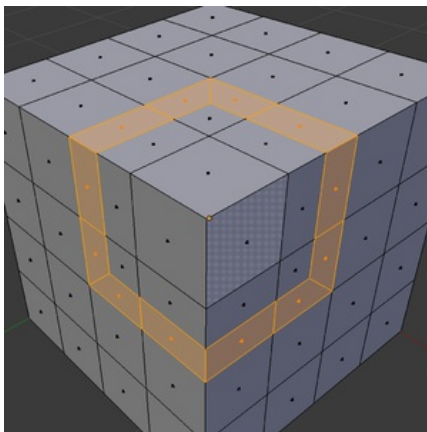
ホットキー: I

メニュー: メッシュ » 面 » 面の差し込み または CtrlF » 面の差し込み

現在選択されている面の内側に、厚さと深さを調整可能な面 (inset) を作ります。ツールを実行するとモーダルになり、マウスを動かして厚さを調整可能です。同様にCtrlを押しながら操作すると、差し込む面の深さを調整できます。

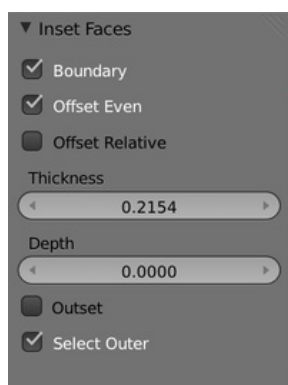


面を差し込みたい面を選択



差し込んだ面が選ばれている状態

オプション



面の差し込み操作の設定

境界 (Boundary)

開いた辺の内側に面を差し込むかどうかを指定します。

均一オフセット (Offset Even)

均一の厚さになるよう拡大縮小します。

相対的なオフセット (Offset Relative)

周囲のジオメトリに相対的に拡大縮小します。

厚さ (Thickness)

差し込む面のサイズを決めます。

深度 (Depth)

新たに内側に差し込む面を持ち上げるか押し下げる深さです。

外側に挿入 (Outset)

内側ではなく外側に面を差し込みます。

より外のものを選択 (Select Outer)

内側に差し込んだ面のどちら側を操作後に選択するか決めます。

Spin

(スピントール)

モード: Edit モード

パネル: Mesh Tools (メッシュツール)

Spin ツールは、ろくろで作るようなオブジェクトを生み出すのに使ってください(このため「ろくろツール」や「掃引ツール」と呼ぶ文献もあります)。具体的には、3Dカーソル位置を通してビューに垂直な軸を中心にして、選択要素を円を描くように押し出す処理が行われます。

- 押し出しの回転軸は視点で決まります
- 3Dカーソルの位置は回転の中心になります

Screw モディファイア

Screw 値を 0 にした [Screw モディファイア](#) を、Spin と同じ目的に使うことができます。

ツールの実行後、ツールシェルフの最下部に次の設定が現れます:

Steps (ステップ)

「掃引」をいくつに分けて行うか指定します

Dupli (スピン複製)

有効にすると、もともと選択していた要素をメッシュ内に切り離されたまま残します (複製間が辺や面でつながりません)

Angle (角度)

「掃引」角度を指定します。単位は度で、範囲は -360 から 360 です (例: 半周させるには **180°** にします)。

Center (中心)

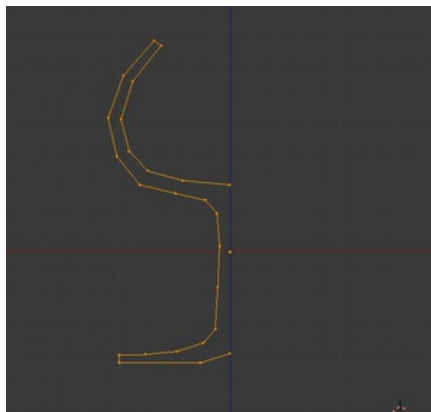
スピンの中心を指定します。デフォルトではカーソル位置を使います

Axis (軸)

スピンの軸をベクトルとして指定します。デフォルトではビューに垂直な軸を使います

Example

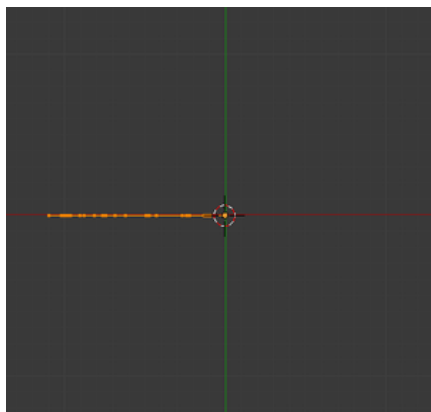
(例)



グラスの断面図

まずはオブジェクトの断面図を表すメッシュを作ります。空洞になったものをモデリングするつもりなら、輪郭線を厚くしたほうがいいかもしれません。図「グラスの断面図」は、これからモデリングするワイングラスの断面図を示しています。

上面図に表示したカーソルを中心に回転させたいので、7 NumPad で上面図にします。



上面図で見たスピン直前のグラスの断面

中心にある頂点を選択し、**ShiftS** » **Cursor -> Selection** で断面図の中心にカーソルを置きます (図「グラスの断面図」)。図「上面図で見たスピン

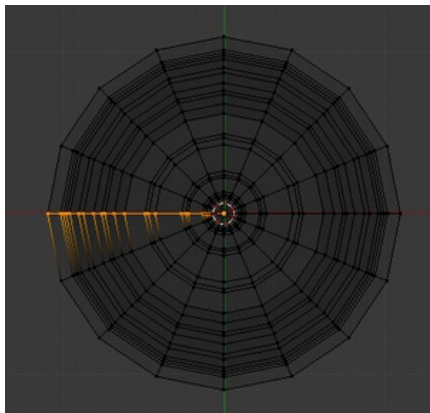
直前のガラスの断面」はワイングラスの断面(にあたる頂点)を上面図で見たところで、カーソルが正しい位置に置かれています。

Spin ボタンをクリックします。結果はすぐに表示されます(図「スピンした断面図」)。ツールシェルフ(T)にはスピン操作に関するオプションが現れています

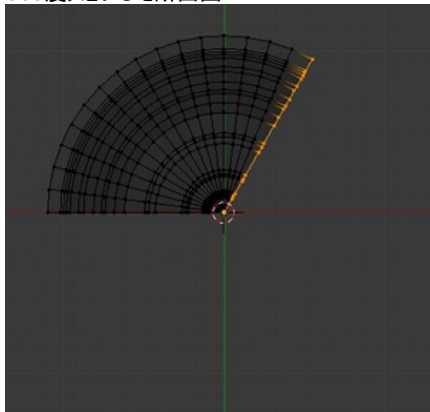
Angle

(角度)

ガラスの形をつくるため Angle を(なっていないければ)360度にします。



360度スピンした断面図

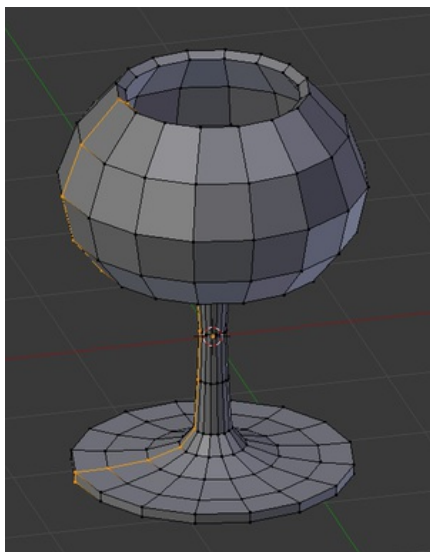


120度スピンした断面図

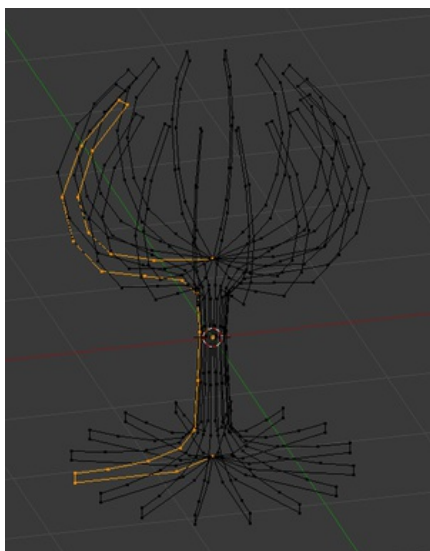
Dupli

(複製)

今回は Dupli をオフにします。



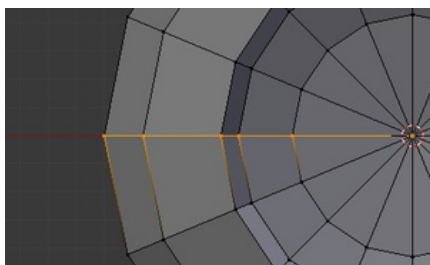
スピン操作の結果



Dupli 有効時の結果

Merge Duplicates

(重複頂点の結合)



複製された頂点

スピン操作は断面図上に重複した頂点を置いたままにします。この継ぎ目の頂点を矩形選択 (B) ですべて選択して (図「複製された頂点」)、Remove Doubles (重複削除) を行うと削除できます。



重複削除後の頂点数 (Blender 2.4 の UI です)

Remove Doubles 操作の前後で、選択頂点の数の変化に注目してください (図「重複削除後の頂点数」)。すべてうまくいけば、最終的な頂点数 (この例では 38 です) はもとの断面の頂点数と一致するはずですが、一致していなければ、削除し損ねた頂点があり、手動で結合する必要があります。もしくは、もっと悪いパターンでは、余分な頂点まで結合しています。

二頂点の結合

二頂点を Merge (Weld) するには、**Shift RMB** で両方を選択して S で拡大縮小を開始し、Ctrl を押しながら X、Y、Z 軸方向に 0 になるまで縮小します。LMB を押して拡大縮小を完了させ、ツールシェルフ (T) にある Remove Doubles ボタンを押すか、W » Remove Doubles) を選びます。別の方法として、同じ Specials メニューの W » Merge (結合) を使うこともできます (または AltM)。ポップアップメニューが開いたら、結合した頂点の配置先を選びます。この例では Center がよいでしょう！

Recalculate Normals

(法線の再計算)

残された作業は、法線を再計算して外側に向ける処理だけです。全頂点を選択して CtrlN を押すか Mesh (メッシュ) » Normal (ノーマル) » Recalc Normals Outside (面の向きを外側に揃える) を選びます。

メッシュ分割ツール(Mesh Subdividing Tools)

分割(Subdividing)は既存の面や辺を切断して細かくし、解像度を増やす操作です。分割するためのツールは複数あります:

[分割\(Subdivide\) \(en\)](#)

解像度を追加して面や辺をより細かくします

[ループ分割\(Loop Subdivide\) \(en\)](#)

既存の辺ループの間に、辺ループを挿入します

[ナイフ分割\(Knife Subdivide\) \(en\)](#)

辺や面を対話的に切ります

[ベベル\(Bevel\)](#)

辺や頂点を分割して面取りや角の丸めを行います

細分化 (Subdivide)

モード: 編集モード

パネル: メッシュツール (編集コンテキスト)

ホットキー: W » 1 NumPad/2 NumPad

メニュー: メッシュ » 辺 » 細分化 (Subdivide), Specials » 細分化 (Subdivide)/滑らかに細分化 (Subdivide Smooth)

このツールは、選択した辺や面を半分以上に切り分けて、必要に応じて頂点を追加し、それにもなつて面を分割します。細分化は次のルールと設定に従って行われます:

- 面の1つの辺だけが選択されているときは (三角モードのとき)、三角形は2つの三角形に分けられ、四角形は3つの三角形に分けられます。
- 面の2つの辺が選択されているときは:
 - この面が三角形なら、新しい2つの頂点の間に新たな辺を作って、三角形を三角形と四角形に分割します。
 - この面が四角形で、選択した辺が隣り合う場合は、四角の角タイプ (Corner Cut Type) (メッシュツールパネルの細分化ボタンの隣にあるドロップダウンメニュー) の設定によって3つの挙動が考えられます。詳細は下記を参照してください。
 - この面が四角形で選択した辺が対岸にある場合には、四角形は新たな2つの頂点を結ぶ辺によって、単に2つの四角形に分けられます。
- 面の3つの辺が選択されているときは:
 - この面が三角形なら (面全体が選択されていることになり)、4つの小さな三角形に分割されます。
 - この面が四角形なら、選択した辺のうち対岸にある2つの辺が上述のように分割されます。続いて「中央の」辺が分割され、この結果上述のような新たな「サブ四角形」を1つの辺に対してだけ作ります。
- 面 (四角形) の4つの辺が選択されていれば、面は4つのより小さな四角形に分割されます。

オプション

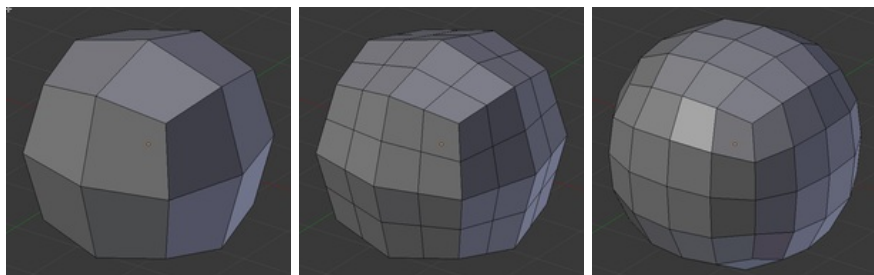
オプションはツールの実行後、ツールパネルから利用できます。

分割数 (Number of Cuts)

1つの辺の分割数を指定します。デフォルトは1で、辺をちょうど半分に分けます。値が2なら3つに分ける、といった具合です。

スムーズ (Smoothness)

おおよその曲率を維持するように分割位置をずらします。この効果は subdivision モディファイアがメッシュを変形する方法に似ています。



細分化前のメッシュ

スムーズを使わない細分化

値1のスムーズを使った細分化

四角/三角モード (Quad/Tri Mode)

細分化の結果にnゴンを使わず三角形を作ります。旧式の挙動をシミュレートしています (下の例をご覧ください)

角タイプ (Corner Cut Type)

四角形の隣接する2つの辺だけが選択されているときの細分化方法を選びます

扇型 (Fan)

四角形は、選択した2辺の反対側に共通の頂点を持つ、4つの三角形の扇型に分けられます

内部頂点 (Innervert)

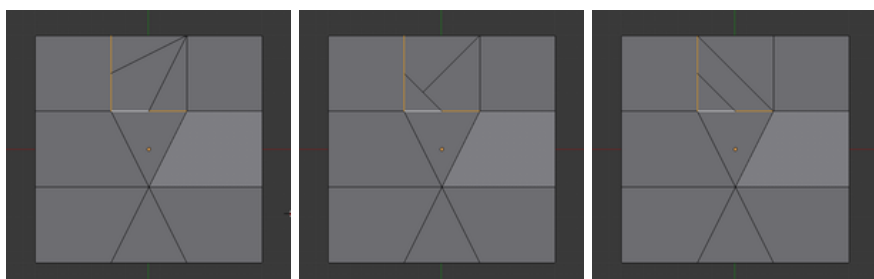
選択した辺はサブ分割され、新たな2頂点の間に辺が作られて小さな三角形ができます。最終的に1つの三角形と2つの四角形ができます。

パス (Path)

まず、選択した2辺の対向する端点間を結ぶ辺が作られ、四角形を2つの三角形にわけます。それから、この三角形が上述のやり方でわけられます。

直線 (Straight Cut)

現在機能していません...



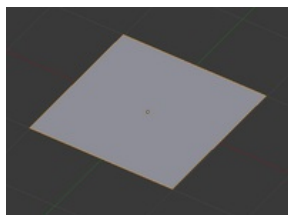
扇型

内部頂点

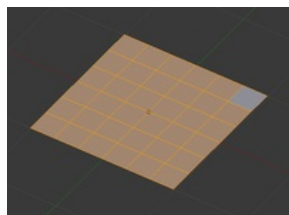
パス

フラクタル (Fractal)

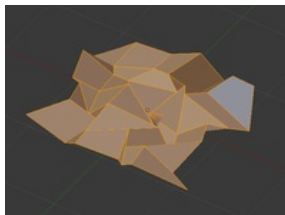
メッシュの細分化のあとで、頂点をランダムな方向にずらします。



細分化前の平面



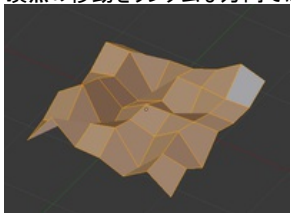
標準的な細分化



同じメッシュにフラクタルを使った場合

法線に沿う (Along Normal)

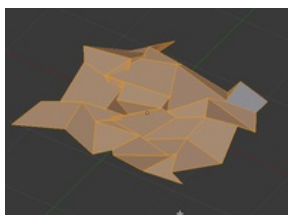
頂点の移動をランダムな方向ではなく、その法線に沿うようにします



法線に沿うを 1 にセット

ランダムシード (Random Seed)

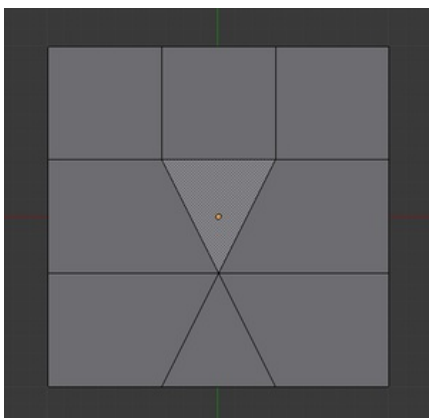
ノイズ発生関数の乱数のシードを変えます。シード値ごとに結果が変わります。



シード値の違う同じメッシュ

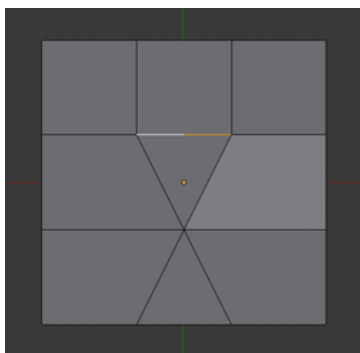
例

以下は細分化とSubdivide Multi ツールでできる様々な例を表すものです。細分化後の選択部分に注意してください。

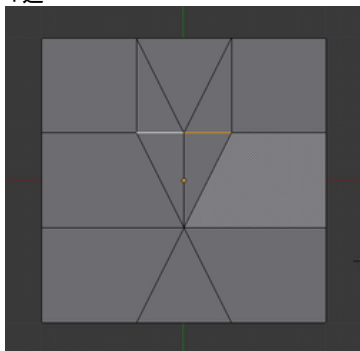


サンプルメッシュ

1 辺

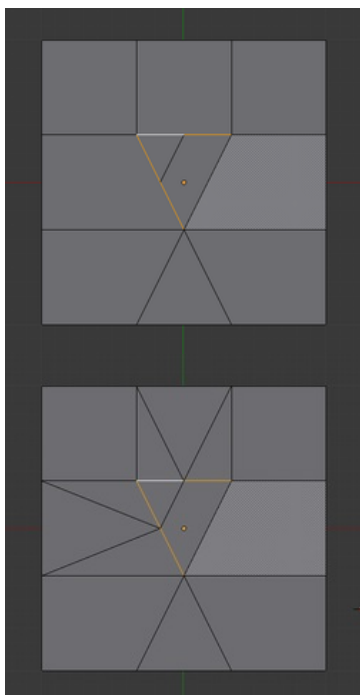


1 辺



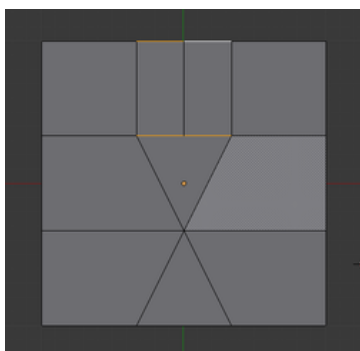
四角/三角モード

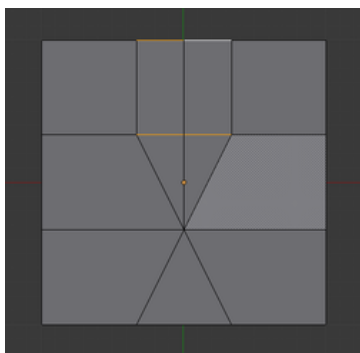
三角形の2 辺



四角/三角モード

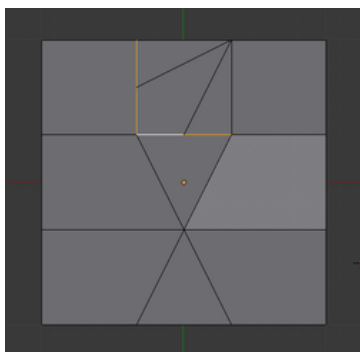
四角形の対抗する2 辺



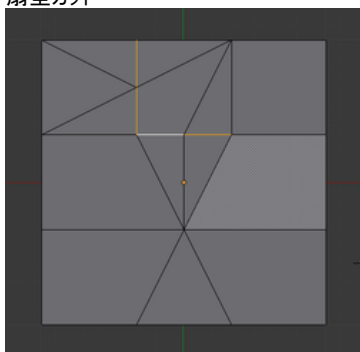


四角/三角モード

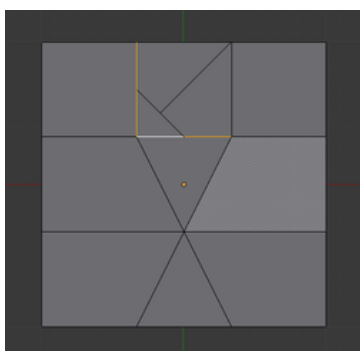
四角形の隣接2辺



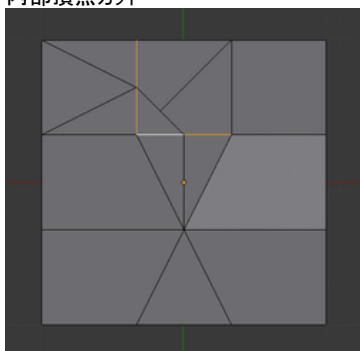
扇型カット



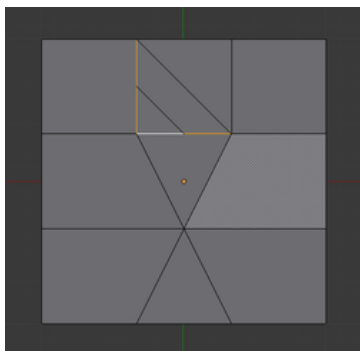
四角/三角モード



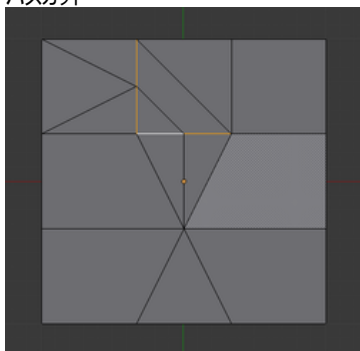
内部頂点カット



四角/三角モード

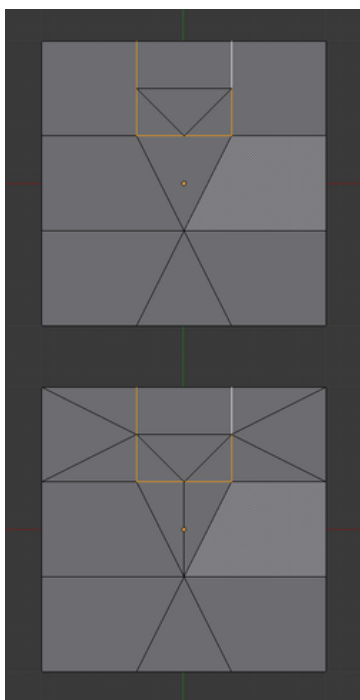


パスカット



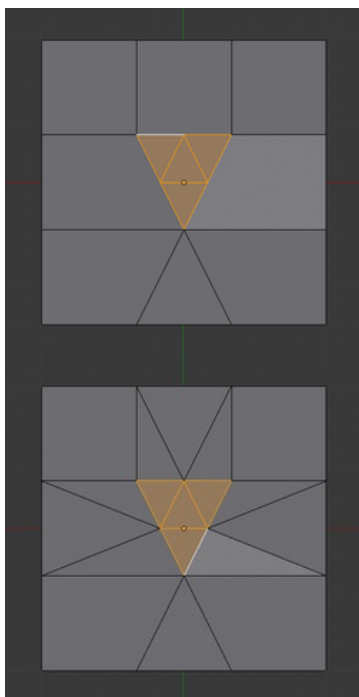
四角/三角モード

3辺



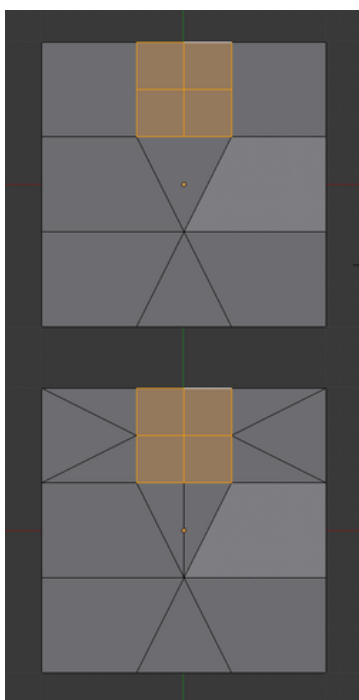
四角/三角モード

三角形



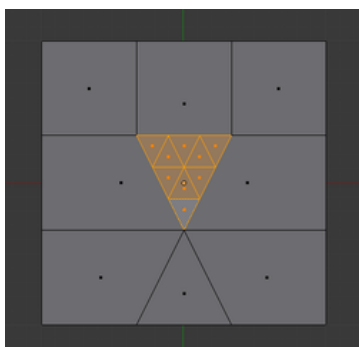
四角/三角モード

四角形/4辺

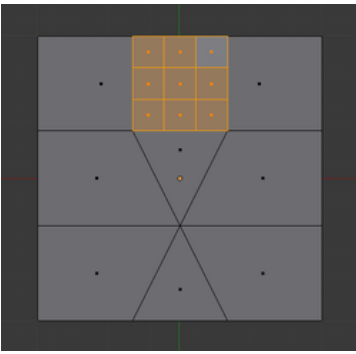


四角/三角モード

多分割



三角形、分割数2



四角形、分割数2

ループ分割 (Loop Subdivide)

モード: 編集モード

パネル: 編集コンテキストで → メッシュツール

ホットキー: CtrlR




ループカット (Loop Cut) は、選択した辺に交差するように新たな辺ループを挿入して、面にあるループを分割します。このツールは対話的で、2つの手順を踏みます:

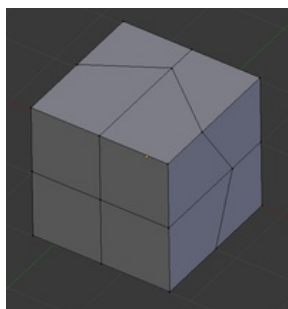
使い方

切断のプレビュー

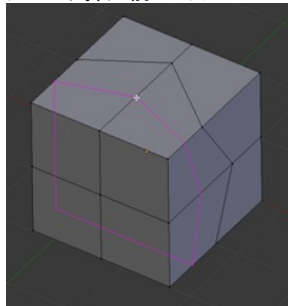
ツールを実行したら、目的の辺にカーソルを動かします。マウスを辺の上に動かす間、マゼンタ色の線で切り口が示されます。新たに作られる辺ループは既存の面ループの終わりの両極 (三角面とnゴン) で止まります (訳注: 原文は "The to-be-created edge loop stops at the poles(tris and ngons) where the existing face loop terminates.")。

新たな辺ループのスライド

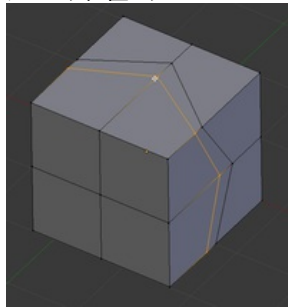
LMB  で辺が一旦選ばれると、マウスを辺に沿って動かして、新たな辺ループをどこに作るか決めることができます。これは [辺スライドツール \(Edge Slide tool\)](#) ([en](#)) と全く同じです。再度 LMB  クリックすると、プレビューしていた位置で切断を行います。または RMB  をクリックして正確に 50% に切断できます。この手順は複数の辺ループを使っていると省かれます (下記参照)



辺ループ挿入前のメッシュ



辺ループ位置のプレビュー



対話的な辺ループの隣接ループ間への配置

オプション

オプションはツールの使用中にだけ利用可能で、3D ビューのヘッダーに表示されます


Even E

単独の辺ループにのみ使えます。辺ループの形状を隣接の辺ループの一つにあわせませす (詳細は [辺スライドツール \(Edge Slide tool\)](#) ([en](#)) をご覧ください)

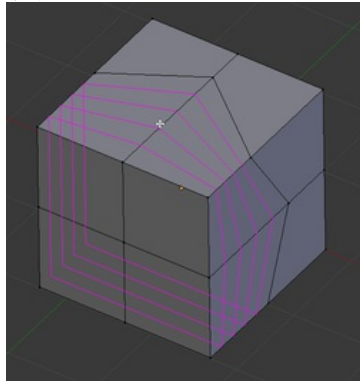
Flip F

Evenが有効のとき、形状をあわせる辺ループを逆にします (詳細は [辺スライドツール \(Edge Slide tool\)](#) ([en](#)) をご覧ください)

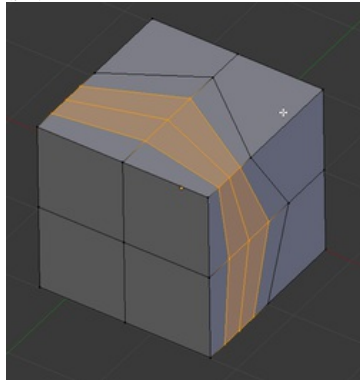
切断数 (Number of Cuts) Wheel または + NumPad/- NumPad

ツールの実行後、最初のループ位置の確認をするまでの間、キーボードで数値を入力するか、Wheel  をスクロールするか、または + NumPad と - NumPad を使って、切断数を増減できます。

複数のループを作ると、元の面ループを均等に分割した箇所で切断され、個々の位置を操作することはできません。



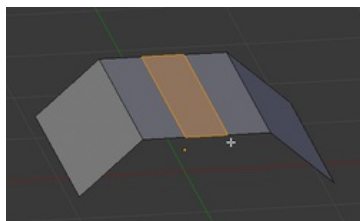
複数の辺ループのプレビュー



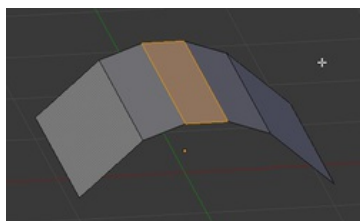
複数切断の結果

円滑化 (Smoothing) Alt Wheel

辺ループの位置を、追加する面を基準に外側か内側に指定したパーセンテージ分動かします。滑らかに分割 (Subdivide Smooth) コマンドに似た動作です。円滑化を使わない場合、新しい辺ループの新しい頂点は、正確に既存の辺上に置かれます。この結果切断された面は平らなままですが、特に [細分化 \(Subdivision Surfaces\) モディファイア](#) を使っていると、ジオメトリが歪む可能性があります。円滑化は一旦サーフェスが分割されたあとの、曲率の維持に役立ちます。



円滑化を使わずに辺ループを追加した場合



同じ辺ループだが円滑化値を使った場合

Knife

(ナイフツール)

モード: 編集モード

パネル: Mesh Tools (メッシュツール)

ホットキー: K または ⇧ ShiftK

ナイフツールは Blender 2.6 で改良されました。このツールは、ユーザーの描いた「ナイフ」の線と交差した線や面を切断します。完全に対話的に操作でき、辺、切れ目、頂点にスナップし、辺を複数回切ることができます。

例えば、球の前面に穴を開けたいとき、前にある辺だけを選んで、その上にマウスで線を引きます。ツールは対話的で、全選択か、矩形選択や ⇧ Shift RMB クリックにより選ばれた辺を対象にします。

⇧ ShiftK やツールパネルにある選択ツールを使って、ナイフツールが選択部分にのみ、貫通 (cut-through) モード (後述) で作用するようにします。

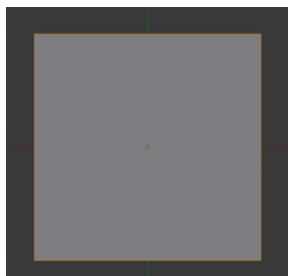
Usage

(使い方)

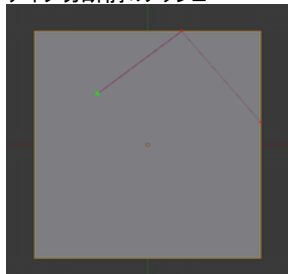
K (または ⇧ ShiftK) を押すと、対話的な実行が開始されます。

切り口を描く

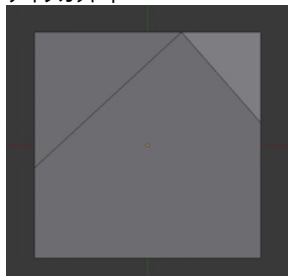
ナイフ分割中はカーソルがメスのアイコンに変わり、ヘッダーがツール用のオプションを表示するように変化します。直線を LMB クリックで繋げて描いていきます。



ナイフ切断前のメッシュ



ナイフカット中



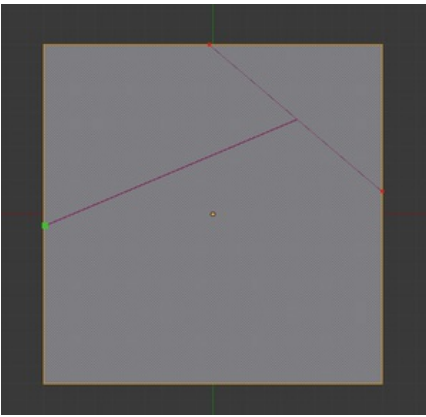
ナイフカット完了

Options

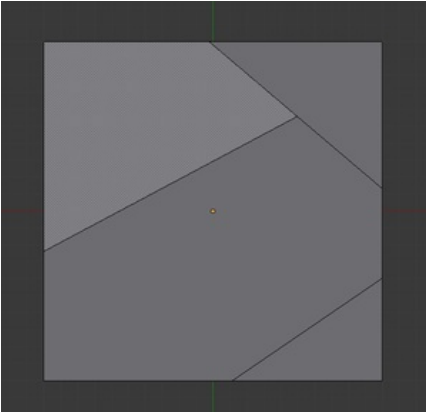
(オプション)

新規カット (New cut) E

(それまでとは別の) 新たな切り口の指定に入ります。複数の、別の切り口を作れます。切り口が複数あっても、すべてがスナップ対象になります。



複数の切り口を作る



新たな切り口を増やした結果

中点にスナップ (Midpoint snap) Ctrl

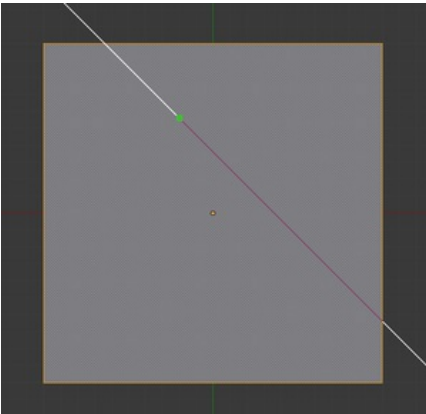
押し続ける間、辺の中点にカーソルをスナップします

スナップ無視 (Ignore snap) ⇧ Shift

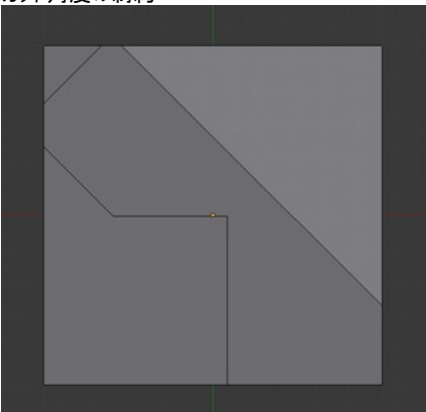
押し続ける間、スナップ設定を無視します

角度制約 (Angle constrain) C

カット方向をビュー上の45度ずつに制限します



カット角度の制約




カット角度制約の結果

透過 (貫通、Cut through) Z

見えている面だけでなく、隠面まで貫通して切断します。

Confirming and selection

(実行の確定と選択)

実行のキャンセルはいつでも、Esc を押すか RMB  クリックで行えます。切断を確定するには \leftarrow Enter を押します。このとき次のオプションがあります:

\leftarrow Enter は、切断でできた新たな辺以外の辺を選択したままにします。

Limitations

(制限事項)

どの辺にも触れない切り口を面の中央に作ろうとすると、この切り口は無視されます。現在のところ、これは Blender で作れるジオメトリの制限です。

閉じた円は面の中央で穴を開けるように切ることができますが、同様のモデリング制限により、この穴はそれを囲むジオメトリと2つの辺で結ばれます。'貫通 (cut through)' モードでは、面を完全に横切る切り口だけが切られます。

Optimizations

(最適化)

大きなメッシュでは、分割したい辺に属する頂点だけを選んで選択頂点の数を減らすと、ナイフの軌跡の交差チェックの時間を節約できるため、より高速動作します。

Knife Project

(ナイフ投影)

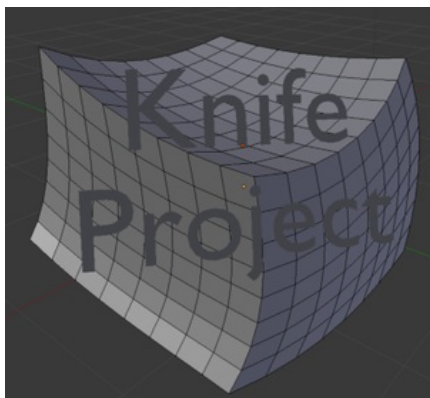
ナイフ投影は、切り口を手描きで決めず、別のオブジェクトをクッキー型のように使って決める、非対話的なツールです。

まず、切り口に使う輪郭線を持つオブジェクトと、切りたいオブジェクトをオブジェクトモードで選択します。編集モードに入りこのツールを実行すると、選択した他のオブジェクトの輪郭線を使って、メッシュに切り口が入ります。実行後は、ジオメトリ内の切り抜いた輪郭線の内部が選択状態になります。

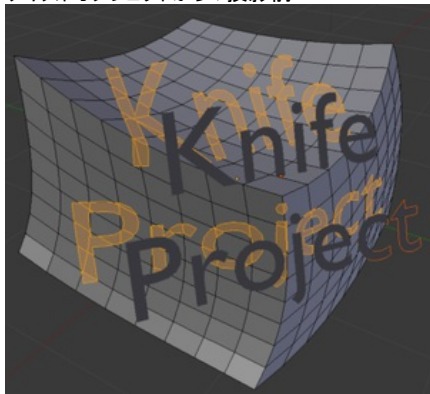
ワイヤーか境界辺を輪郭線に使えます。

Examples

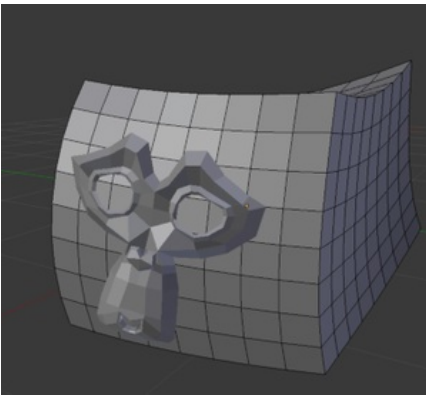
(例)



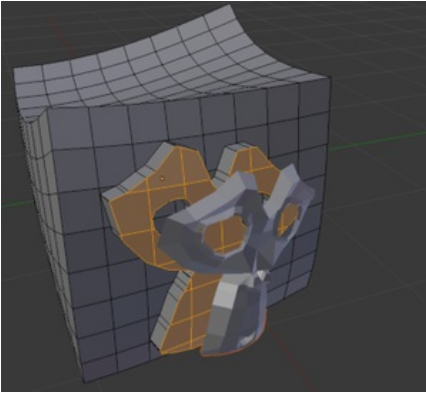
テキストオブジェクトからの投影前



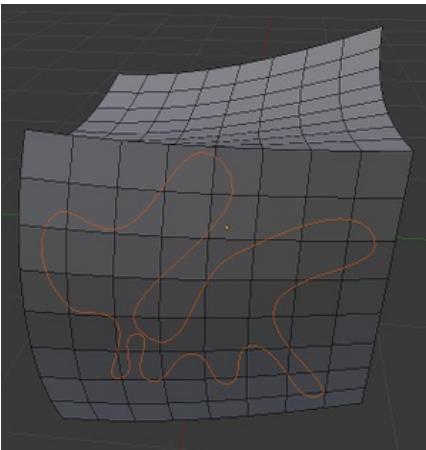
ナイフ投影の結果



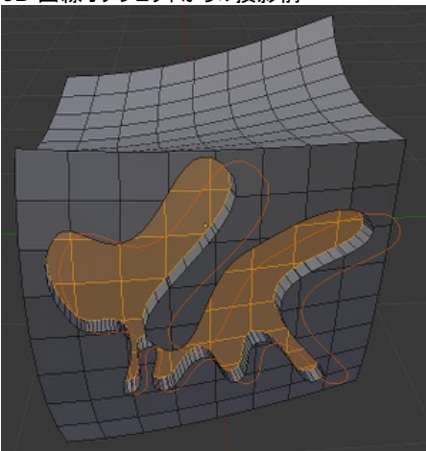
メッシュオブジェクトからの投影前



ナイフ投影の結果 (実行後押し出し)



3D 曲線オブジェクトからの投影前



ナイフ投影の結果 (実行後押し出し)

Known Issues

(既知の問題)

単一の面に穴を開けるような切断は失敗する可能性があります。これは通常のナイフツールと同じ制限ですが、特にテキストを使う場合は注意すべきです。より細かく分割されたジオメトリに投影することで、回避可能です。

Mesh Bisect

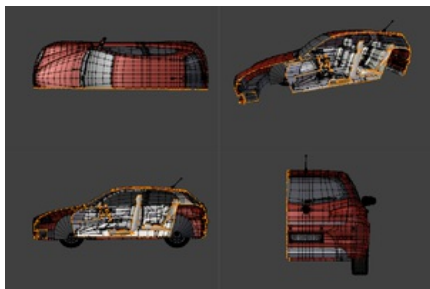
モード: 編集モード

メニュー: Mesh(メッシュ) » Bisect(二等分)

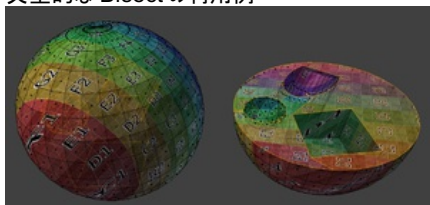
Bisect(二等分)ツールは、メッシュを任意の平面に沿って真っ二つに切る、手軽な手段です。

このツールとナイフツールには、三つの大きな違いがあります。

- 平面を操作パネル上で正確に、数値で調整できます
- 切断により、片側のジオメトリ(=メッシュ表面の形状)が取り除かれるかもしれません
- オプションとして、切断によりできた穴を、周囲のジオメトリの持つマテリアルや UV、頂点色を使って埋めることもできます。つまり、Bisect ツールはメッシュの一部を、一切穴をあけずに切り取れるということです



典型的な Bisect の利用例



穴埋めオプションを使った Bisect の例

Vertex Connect

(頂点の結合)

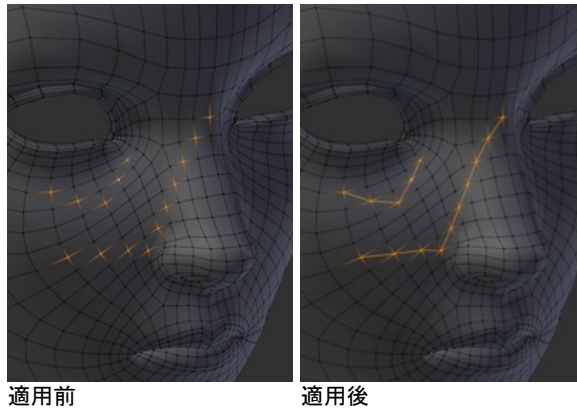
モード: 編集モード

ホットキー: J

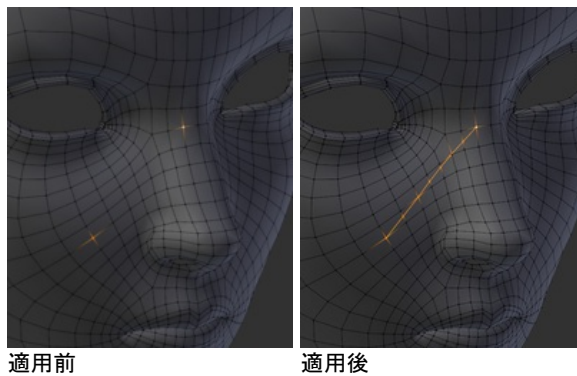
メニュー: Mesh(メッシュ) → Vertices(頂点) → Connect(頂点の結合)

このツールは選択頂点を辺で結びます。辺の作成との主な違いは、新たに結合された頂点により、面が分割されることです。

たくさんの頂点を選ぶと各面は、その面にある、選択されている頂点で分割されます。



ふたつの頂点のみが選択されていれば、ナイフツールに似て、非選択の面を横切るように切断されます。ただしこれは、つながった面を横切る、まっすぐな切断に限られます。



ベベル(Bevel)

モード: 編集モード

ホットキー: CtrlB または W » Bevel

メニュー: Mesh » Edges » Bevel または CtrlE » Bevel



ベベル使用後と使用前

ベベルツールはジオメトリに面取り、あるいは角の丸めを行います。ベベルとは辺や角を滑らかにする効果です。現実世界にはまっすぐに切り立った角はほとんどありません。ナイフの刃でさえも完璧な鋭さを持っていません。ほとんどの辺には、物理的かつ現実的な理由で意図的に面取りがされています。


ベベルは無機的なモデルに現実感を与えるのに役立ちます。現実世界では、物の辺の丸まりは光を受けるため、辺の周辺の陰影を変化させます。ベベルを行わないオブジェクトの見た目が完璧すぎるのとは違って、立体的で現実的に見えます。

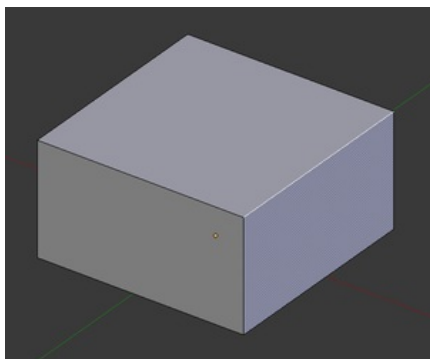
ベベルモディファイア

[ベベルモディファイア](#)はベベルツールの非破壊的な代替手段です。同じオプションを持ち、頂点ウェイトによる制御などの追加機能があり、モディファイアの一般的な強み(非破壊であることなど)も備えています。ベベルモディファイアは再帰的なオプションを持たないことに注意してください。モディファイアを複数追加して効果を重ねると、同等の機能を得られます。

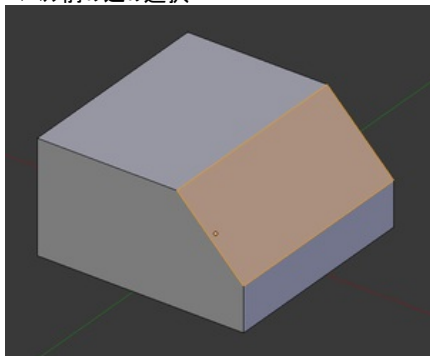
使い方

ベベルツールは選択した辺にのみ働きます。選択された頂点や面に含まれる辺も同様に認識し、辺を明示的に選択したのと同じようにベベルが実行されます。ベベルツールは辺と「コーナー(頂点)」を、指定回数「分割」して滑らかにします(ベベルの詳細なアルゴリズムについては、下記のオプションをご覧ください)。

CtrlB または上の方法を使ってツールを実行します。マウスを動かさず対話的にベベルの量を決めことができ、Wheel  をスクロールすると線分の数を増減できます(後述)。

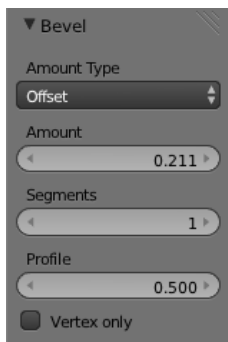


ベベル前の辺の選択





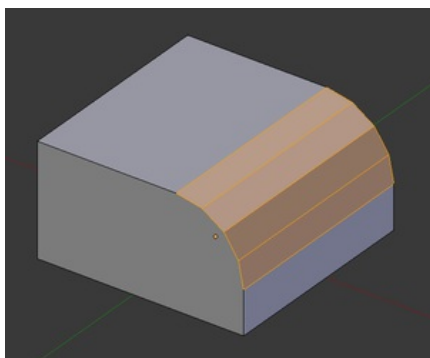
ベベルの結果

オプション



Offset

マウスをオブジェクトから離したり近づけたりして、トランスフォームツールに似た要領でベベル幅を変えることができます。いつもと同じように、⇧ Shift を押したままスケールすると 0.001 ステップ刻みで細かくスケールできます。LMB  は操作を完了させ、RMB  か Esc で実行を中止します。



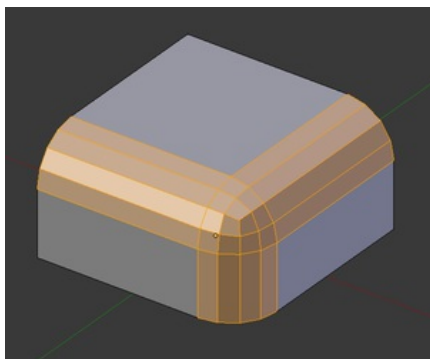
4つの線分によるベベル

Segments

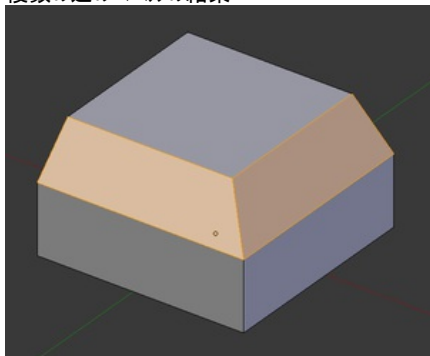
ベベルに含まれる線分の数はマウスのホイールで増減できます。数が大きいほどベベルは滑らかになります。

替わりに、手動でスケール値を入力することもできます。ツール使用中に数値を入力するか、ツールの使用後にメッシュツールオプションから指定します。

例



複数の辺のベベルの結果



複数辺ベベルの別の例

その他の編集ツール

要素のソート(sort)

Specials、Vertices、Edges と Faces の各メニューにあるツールで、選択したメッシュ内のマッチする要素の順序を、次のような様々な方法で並べ替えます。Specials メニューから呼び出した場合は、アクティブな選択モードと同じものが対象になることに注意してください。

View Z Axis(ビューZ軸)

アクティブなビューのZ軸に沿って、デフォルトでは遠い側から近い側に並べます。順序を逆にする場合は Reverse を使います。

View X Axis(ビューX軸)

アクティブなビューのX軸に沿って、デフォルトでは左から右に並べます。これも反転用の Reverse オプションがあります。

Cursor Distance (カーソル距離)

3D カーソルの位置に近い側から遠い側に向けて並べます。反転用の (Reverse オプションがあります)。

Material(マテリアル)

面のみに使えます！ マテリアルのインデックスが低いものから高いものの順に並べます。各「マテリアルグループ」内での面の順番は変更されません。Reverse オプションはマテリアルの順序だけを反転し、面の順序は反転しないことに注意してください。

Selected(選択部位)

選択中のすべての要素を最初の位置に (Reverse を使うと最後の位置に)、相対的な順序を変えずに移動します。**警告: このオプションは非選択の要素のインデックスにも影響します！**

Randomize(ランダム化)

選択要素のインデックスを乱数で決めます (非選択のものには *影響しません*)。seed オプションを使うと別の乱数を使えます。同じ seed と同じメッシュや選択要素の集合を使う限り、結果が常に同じになります！

Reverse(反転)

選択要素の順序を単に逆順にします。

Retopologizing

註記

Blender 2.5 で、Retopo tool は改良されたメッシュのスナップ機能に置き換えられました。このページの記述は Blender の将来のバージョンで retopology tools が更新されれば変化します。

Retopology は一般的なモデリング工程の一つです。モデルは形状や細部に重点をおいて作られることがよくありますが、topology または辺の流線が理想的なものではないか、メッシュがひどく密集していて、無駄が多くなっています。モデル製作者は元のメッシュに形状がマッチした、低解像度のメッシュを新たに作ることができます。

メッシュへのスナップ

面へのスナップを有効にすると、メッシュの頂点がビューのZ軸でビューポート上の一番近いサーフェスに移動します。

形状を気にせず自由なモデリングができるようになり、topology に集中できます。

[スナップ](#) をご覧ください。

収縮包装モディファイア (Shrinkwrap)

[収縮包装モディファイア](#) は面のスナップと組み合わせると便利です。スナップを無効にして作られた新しいメッシュを編集すると、このモディファイアはラップを巻きつけるように、新しいメッシュを古いメッシュに張り付けます。

Overview

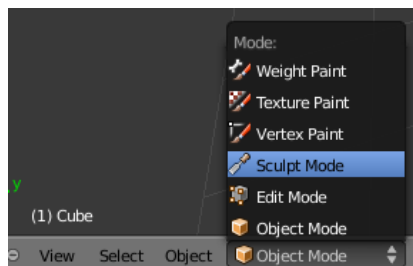
(概観)

Sculpt モードはモデルの形状を変えるために使われるという点では 編集 モードと似ていますが、大きく異なる作業工程を使います。個別の要素(頂点、辺、面)を扱う代わりに、モデルの一定範囲をブラシを使って変更します。言い換えれば Sculpt モードはブラシの場所を基準に頂点を自動選択し、編集するのです。

Sculpt Mode

Sculpt モードは 3D ビュー ヘッダーのメニューから選べます。

いったん sculpt モードに入ると 3D ビュー の ツールシェルフ が sculpt モード専用のパネルに変わります。ツールシェルフにあるパネルは Brush(ブラシ)、Texture(テクスチャ)、Tool(ツール)、Symmetry(対称)、Stroke(ストローク)、Curve(カーブ)、Appearance(外観) と Options(オプション) です。また、3D ビューのカーソル位置に赤い円が追従ようになります。



Sculpt モード ドロップダウン項目。



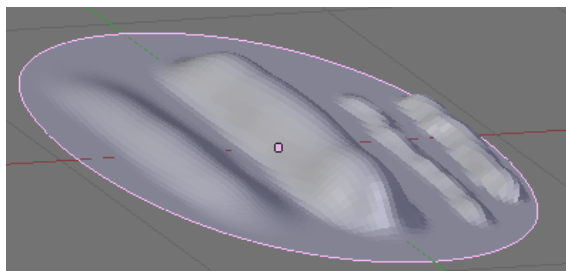
Sculpt モードのカーソル

Sculpt Brushes

(スカルプトブラシ)

あらかじめ用意されたブラシが複数あります。ブラシは stroke(ストローク)、texture(テクスチャ)やオプションなどのツールの組み合わせです。

Sculpt モードにはブラシが 16種類あり、それぞれ違った方法でモデルを制御します。多くは加算と減算の効果を切り替えることができます。選択は Tool メニューで行います。以下はブラシの現在のプリセットです:



様々な大きさと強さでの描画

Blob(プロブ)

メッシュを外側か内側に球状に押し込みます。球のエッジでつまむ量を調整する設定があります。

Clay(クレイ)

Draw ブラシに似ていますが、ブラシの動作する平面の調整項目を含みます。

Clay Strips(クレイストリップ)

Clay ブラシに似ていますが、ブラシの効果範囲を球ではなく立方体で決めます。

Crease(クリーズ)

複数の頂点をまとめてつまみながら、メッシュを押すと鋭い切れ込み、引くと鋭い隆起を作ります。

Draw(ドロー)

ブラシストロークの範囲にある、頂点の法線の平均にもとづいて頂点を内側か外側に動かします。

Fill(面を張る)

Flatten ブラシと同じように動作しますが、上向きブラシ平面より低い頂点だけを動かします。Fill と逆の動作をするのが Deepen で、下向きの平面より低い頂点を押し込みます。

Flatten(平面化)

デフォルトではブラシ領域内の頂点の上下方向の高さの平均位置にある 'area plane' を見つけて、頂点をこの平面に向かって引っ張ります。

Flatten ブラシの逆の動作をするのが Contrast ブラシで、ブラシ平面より上下に離れている頂点を押し込みます。

Grab (ひっぱる)

周辺の点の集まりを動かすのに使います。他のブラシと違い、Grab はブラシがドラッグした点以外を編集しません。代わりにマウスダウンで頂点の集まりが選ばれ、マウスに追従して引っ張られます。この効果は 編集 モードでプロポーション編集を有効にして頂点の集まりを動かすのに似ていますが、Grab では他の Sculpt モードのオプションを使える点が違います (例えばテクスチャやシンメトリー)。

Inflate (膨らます)

Draw に似ていますが、Inflate モードの頂点は個々の法線方向に変形される点が違います。

Layer (レイヤー)

Draw ブラシに似ていますが、ずらす高さに上限がある点が違います。この結果、立体的な厚い層のような外観が描かれます。このブラシは (1 ストローク内では) 重ね塗りができません。ブラシストロークはそれ自身に交差します。マウスボタンを離すと新たなストロークが始まり、深さをリセットして直前のストロークの一番上塗ります。

Nudge (ナッジ)

ブラシストロークの向きに頂点を動かします。

Pinch (つまむ)

ブラシの中心に向けて頂点を引っ張ります。逆の設定が Magnify で、ブラシの中心から頂点を押し出します。

Rotate (回転)

ブラシの当たる頂点をカーソルを動かした向きに回します。

Scrape (こすり)

Flatten ブラシに似ていますが、下向きの平面より上の頂点だけを動かします。Scrape ブラシの逆の動作をするのが Peak で、平面より上に離れた頂点だけを押し込みます。

Smooth (スムーズ)

名前が示すように、ブラシの効果範囲内のメッシュの不規則な部分を、頂点位置を滑らかにしてなくします

Snake Hook (スネークフック)

ブラシの動きにあわせて頂点を引っ張り、長い蛇のような形状を作ります

Thumb (指)

Nudge ブラシに似ていますが、ブラシ領域のメッシュを、ブラシストロークの向きに動かす間、平らにします。

Sculpting with the Multires Modifier

(Sculpting で Multires Modifier を使う) ...

Sculpt Properties Panel

このパネルは 3D 描画域の左側にあるツールシェルフにあります。

Brush Menu

(ブラシパネル)

Radius (半径)

ブラシの半径をピクセル単位で制御します。3D ビューで F を使うと、マウスドラッグと左クリックでインタラクティブにブラシの大きさを変えられます (円の内側にブラシのテクスチャが見えるはずです)。F で大きさを決めているときに、数字を入力して Enter キーを押せば大きさの数値指定ができます。対応タブレットを使っていれば、筆圧感知アイコンを有効化すると筆圧がブラシの大きさに影響します。

Strength (強さ)

Strength はブラシ1回のモデルへの影響量を制御します。例えば、Draw ブラシでは値が高いほどモデルを深く掘るのが速くなります。Smooth ブラシではモデルの円滑化が速くなります。この設定は Grab、Snake Hook、Rotate では利用できません。

強さの範囲がモデルにあわないような場合は (例えば、一番弱い値でもモデルに変化を与えすぎる場合)、モデルを拡大縮小できます (オブジェクトモードではなく 編集 モードで)。サイズが大きいほどブラシの効果は弱まり、逆も同じです。3D ビューで \diamond ShiftF を押してからマウスを動かし、左クリックするとブラシ強度を変えることができます。 \diamond ShiftF で大きさを変えているときに数値入力も使えます。対応タブレットを使っていれば、筆圧感知アイコンを有効化すると筆圧が強さに影響します。

Autosmooth (自動スムーズ)

各ストロークに使われる円滑化の量を決めます。

Sculpt Plane (スカルプト平面)

sculpt が行われる平面を設定するメニューです。

Plane Offset

ブラシが実行される平面と観察者との距離を調整します。

Trim

Distance 設定により sculpt 平面をトリミングできるようにします。

Front Faces Only (前面のみ)

有効にするとブラシは観察者に向いた頂点だけに影響します。

Accumulate

ストロークが上塗りを繰り返すようになります。

Stroke Menu

(ストロークパネル)

Stroke Method (ストローク方法)

ブラシストロークのメッシュへの適用方法を決めます。

Dots (ドット)

標準的なブラシストロークです。

Drag Dot (ドラッグドット)

ブラシ形状内で1度だけ動きを起こします。クリックしてドラッグすると適用位置を決められます。マウスボタンを離すと確定します。

Space (空間)

ストロークを点の連なりにします。点の間隔は Spacing (間隔) 設定で決めます。Spacing はブラシの直径に対するパーセンテージです。

Anchored (アンカー)

ブラシ位置で1度だけ動きを起こします。クリックしてドラッグするとブラシの直径を変えられます。Edge to Edge (辺から辺へ) が有効ならば、ブラシの位置と方向は、最初にクリックした位置を第1点、最初と反対側にドラッグした位置を第2点としたときの2点円で決まります。

Airbrush (エアブラシ)

マウスを押し続ける間、Rate 設定によるブラシ処理が継続します。無効にすると、ブラシの位置が変わったときだけモデルは編集されます。Grab ブラシには利用できません。

Dots、Space と Airbrush のストロークでは次のパラメータが利用可能です:

Smooth stroke

ブラシをマウスの動きよりも遅らせて、経路を滑らかにします。有効にすると、次の設定ができます:

Radius

ストロークが繋がる最小距離を指定します。

Factor

円滑化の程度を指定します。

Jitter

描画中にブラシ位置をランダムに細かく揺らします。

Curve Menu

(カーブパネル)

Curve 欄では曲線を編集してブラシの強度を制御することができます。曲線の左端がブラシの中央で、右側に向かうほど外縁の強度を表しています。

Texture Menu

(テクスチャパネル)

テクスチャもブラシ効果の強さを決めるのに使われます。既存のテクスチャをテクスチャボックスから選ぶか、New ボタンで新規作成します。

Brush Mapping

ブラシのストロークにテクスチャを対応付ける方法を設定します:

Fixed

有効ならテクスチャがマウスを追随し、テクスチャがモデルにドラッグされているように見えます。

Tiled

画面にタイル状にテクスチャを並べ、ブラシがテクスチャと切り離されて動いて見えます。プロシージャルテクスチャよりもタイリング可能な画像で使うほうが役立ちます。

3D

ブラシにプロシージャルテクスチャを最大限利用します。テクスチャのどの部分を使うか決めるのに、ブラシ位置の代わりに頂点座標を使います。

Angle

テクスチャブラシの回転角度です。3D ビューで CtrlF を押すとマウスを動かして決めることができます。このとき、回転値をキーボードから数字で入力することもできます。設定項目は以下の通りです:

User

角度を直接指定します。

Rake

角度がブラシのストローク方向に従います。3D テクスチャには利用できません。

Random

角度がランダムになります。

Offset

テクスチャマップの x、y、z 軸上の位置を精密に調整します。

Size

テクスチャのスケーリング率を変えます。Drag テクスチャには利用できません。

Sample Bias

テクスチャのサンプル値に追加される値です。

Overlay

有効なら描画領域に Alpha 値によって決まる通りにテクスチャを表示します。

Symmetry Menu

(対称/ロックパネル)

ブラシストロークを、選択したローカル軸に対して鏡映します。軸の方向を変えるときは モデルを オブジェクト モードではなく 編集 モードで回転しなければならぬことに注意してください。

Radial(放射)

指定角度で放射相称を作ります。数値は回転軸の周囲360度内でストロークが何回繰り返されるかを表します。

Feather(フェザー)

鏡面側の平面に重なったストロークの、強度を抑えます。

Lock(固定)

スカルプト中のモデルに対する、選択したローカルの座標軸方向の編集や変形をすべて遮断します。

Options Menu

(オプションパネル)

Threaded Sculpt(スレッドスカルプト)

マルチ CPU のパフォーマンスを向上させます。

Fast Navigation(操作を高速化)

Multires モデル用です。描画域の操作をする間は低解像度で表示します。

Unified Settings(共通設定)

Size(サイズ)

指定したブラシの大きさをブラシ間で共有します

Strength

指定したブラシの強さをブラシ間で共有します

Appearance Menu

(外観パネル)

加算と減算のモードの違いに応じて、ブラシの色を変えることができます。

画像ファイルからブラシのアイコンをセットすることもできます。

Show Brush(ブラシを表示)

描画領域でブラシ形状を表示します。

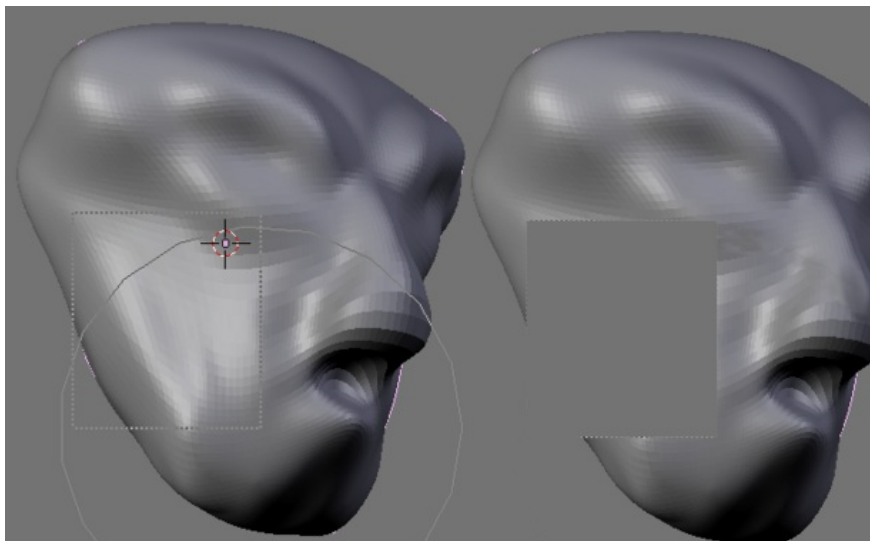
Tool Menu

使うブラシプリセットをここで選べます。Reset Brush(ブラシをリセット) はブラシ設定をデフォルトに戻します。現在のブラシを Vertex Paint(頂点ペイント)、Weight Paint(ウェイトペイント)、Texture Paint(テクスチャペイント) の各モードで使うように設定することもできます。ボタンを使って切り替えます。

Hiding and Revealing Mesh

(メッシュの非表示と再表示)

sculpt を使いたい部分のメッシュを隔離させると便利があります。メッシュの一部を隠すには H を押して、隠したい部分をクリックしてドラッグします。隠した部分の再表示には \diamond ShiftH を押して再表示したい部分をクリックしてドラッグします。隠した部分をすべて表示する場合は単に AltH を押してください。









隠す前と隠した後

Keyboard Shortcuts

(キーボードショートカット) (訳注:バージョンによって異なります)

ショートカットは次の場所でカスタマイズできます: File(ファイル) » User preferences(ユーザー設定) » Input(入力) » 3D View(3Dビュー) » Sculpt Mode(スカルプト)

操作	ショートカット
選択したメッシュを隠す	H のあとクリック & ドラッグ
選択したメッシュを再表示	⇧ ShiftH then click & drag
メッシュ全体を表示	AltH
ブラシサイズをインタラクティブに設定	F
ブラシサイズの増減	[と]
ブラシの強さをインタラクティブに設定	⇧ ShiftF
ブラシテクスチャをインタラクティブに回転	CtrlF
ブラシの向きの切り替え (Add/Sub)	スカルプ中に Ctrl を押し続ける
ストローク方式の設定(エアブラシ、アンカード、...)	A
スムーズストロークの切り替え	⇧ ShiftS
Draw ブラシ	D
Smooth ブラシ	S
Pinch/Magnify ブラシ	P
Inflate/Deflate ブラシ	I
Grab ブラシ	G
Layer ブラシ	L
Flatten/Contrast ブラシ	⇧ ShiftT
Clay ブラシ	C
Crease ブラシ	⇧ ShiftC
Snake Hook ブラシ	K
Mask ブラシ	M
マスクの消去	AltM
マスクの反転	Ctrl
番号でブラシを設定	0 - 9 と ⇧ Shift0 - ⇧ Shift9
スカルプトオプションパネル切り替え	T
多重解像度のレベルを1つ上げる	Page up
多重解像度のレベルを1つ下げる	Page down
多重解像度のレベル設定	Ctrl0 - Ctrl5
動的トポロジーの切り替え	CtrlD
テクスチャ角度のタイプを設定	R
ステンシルテクスチャの移動/拡大/回転	RMB  , ⇧ Shift RMB  , Ctrl RMB 
ステンシルマスクの移動/拡大/回転	Alt RMB  , Alt⇧ Shift RMB  , AltCtrl RMB 

Page status ([reviewing guidelines](#))

Page reviewed and in good shape

Removed from Blender 2.5

Blender 2.5 ではもうこの機能を利用できません。[Multires モディファイア](#) をご覧ください。

Vertex Groups

(頂点グループ)

[- 頂点グループの作成と削除](#)

作成と管理の概要

[- ウェイトの編集 \(eng\)](#)

頂点グループのウェイトの編集

[- ウェイトペイント \(eng\)](#)

ウェイトペイントモード

[- ウェイトペイントツール \(eng\)](#)

ウェイトペイントツール

Vertex Groups

(頂点グループ)

頂点グループは主に、メッシュオブジェクトやラティスの部品を構成する、頂点の識別に使われます。椅子の足やドアのヒンジ、あるいはキャラクターの手、腕、脚、頭、足などを想像してください。さらに、別々のウェイト(weight)値を([0.0,1.0]の範囲で)頂点グループ内の頂点に割り当てることができます(このため頂点グループはウェイトグループと呼ばれることがあります)。

頂点グループはアーマチュア([Skinning Mesh Objects \(eng\)](#))で最もよく使われますが、Blenderの他の多くの場所でも使われます。例えば次のものがあります:

- シェイプキー
- モディファイア
- パーティクル生成
- 物理シミュレーション

他にも多くの用途が考えられます。頂点グループは実際、意味のあるものには何であれ使うことができます。ある場面では、頂点グループを自動生成することができます(たとえばリグづけされたオブジェクト)。しかしながらこの章では、手動で作られた(ユーザー定義の)頂点グループに焦点を当てます。

頂点グループはメッシュとラティスオブジェクトだけに使えます
他のオブジェクト型はどれも頂点を持たず、頂点グループを持つことができません。

Typical usage scenarios for Vertex groups

(よくある頂点グループの用途)

- アーマチュアのスキニング

メッシュをアニメーションさせて動かしたいとき、多数のボーンで構成されたアーマチュアを定義します。頂点グループはメッシュの一部をアーマチュアのボーンに関連付けるのに使われます。頂点グループの各頂点には、[0.0 ... 1.0]の範囲で影響力(weight(ウェイト))を指定できます。

- モディファイアへの働きかけ

モディファイアの多くは、モディファイアの影響力を頂点別に制御することができます。これは頂点グループと頂点に関連付けられたウェイトを通じて行われます。

- メッシュの一部の素早い選択/編集/非表示

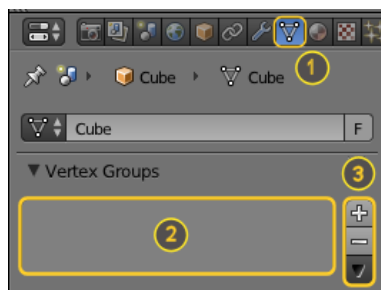
頂点グループを使ってメッシュ領域を定義すると、3回のクリックで簡単に部品全体を選択でき、別のオブジェクトを作らずに、隔離して作業を続けられます。非表示にする機能を使えば、頂点グループをビューから取り除くこともできます(あとで再表示する場合)。

- メッシュの一部の選択と複製

レゴブロックのモデリングをしているものとします。一番単純なブロックは土台と鋸(ブロックをつなげるための突起)でできています。4鋸ブロックを作るために、鋸の頂点を簡単に選択して、編集モードのまま複製し、目的の場所に配置したくなるでしょう。

Creating Vertex Groups

(頂点グループの作成)



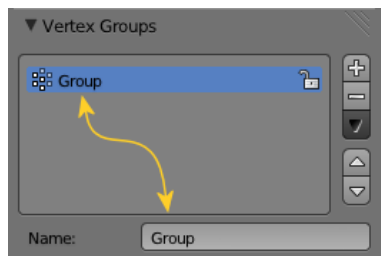
空の頂点グループパネル

頂点グループの管理は Object Data プロパティ ウィンドウ(①)内の Vertex Groups(頂点グループ) パネルで行います。頂点グループが定義されていなければ(新規メッシュオブジェクトのデフォルト)、パネルは空です(②)。

パネル右端にある + ボタン(③)を LMB して頂点グループを作れます。グループの名前は初期状態では Group (またはこの名前がすでにあれば Group.nnn)で、パネル(②)内に表示されます(次の図をご覧ください)。

新しいグループは常に空です

下記のとおり、グループに頂点を明示的に割り当てる必要があります。



オブジェクトモードの頂点グループパネル。グループがひとつあります


Group Name

(グループ名)

最初の頂点グループが作られるとすぐに、パネルのすぐ下に新しい入力欄が現れます。ここでグループ名を都合のよいものに変えることができます。

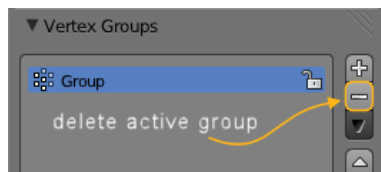
Active Group

(アクティブグループ)

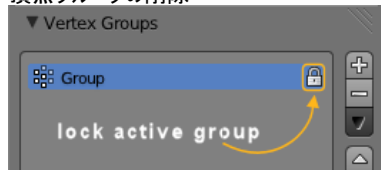
頂点グループが作られると、自動的にアクティブグループとしてもマークされます。これはパネル項目の背景が水色になることで示されています。もし一覧に二つ以上のグループがあれば、頂点グループパネルにある該当項目を LMB  するとアクティブグループを変えられます。

Deleting vertex Groups


(頂点グループの削除)



頂点グループの削除



頂点グループのロック

頂点グループを削除するには、まずアクティブグループにし(パネルで選択します)、次にパネル右端にある - ボタンを LMB  します。

頂点グループの削除は、頂点のグループへの割り当てだけを削除します。頂点そのものは削除されません。

Locking Vertex Groups

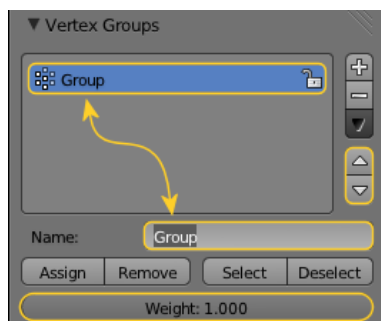
(頂点グループのロック)

頂点グループを作った直後は、開いた鍵アイコンが頂点グループのリスト項目の右側に表示されます。このアイコンは、その頂点グループが編集できることを表しています。グループに頂点の割り当てを追加したり、グループから割り当てを取り除いたりできます。そして、ウェイトペイントブラシなどを使って変更できる、という具合です。

アイコンをクリックすると閉じた鍵アイコンに変わり、頂点グループのすべての変更は無効化されます。グループにできる操作は改名と削除、そして再度鍵を外すことだけです。ロックされた頂点グループに他の操作は一切許されず、同等の機能を持つボタンはすべて、ロックされた頂点グループに対して無効化されます。

Working with Content of Vertex Groups

(頂点グループの中身への働きかけ)



編集モードにおける頂点グループパネル

Edit-Mode か Weight-Paint Vertex Selection Mode のいずれかに切り替えると、頂点グループパネルが広がってさらに2行表示します:

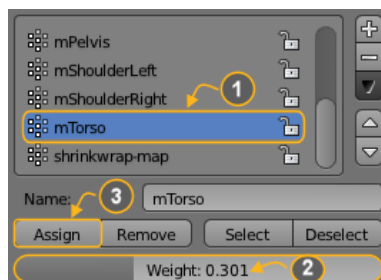
最初の行には、アクティブ頂点グループの頂点の、割り当てと選択を管理する4つのボタンがあります:

- **Assign (割り当て)**: 「Weight(ウェイト):」入力欄で指定されたおりのウェイト(後述)を使って、選択頂点をグループに割り当てます
- **Remove (除去)**: 選択頂点をグループから取り除きます(したがってウェイト値も削除します)
- **Select (選択)**: グループに含まれる全頂点を選択します
- **Deselect (選択解除)**: グループに含まれる全頂点の選択を解除します

ボタン行の下に「Weight(ウェイト):」の数値入力欄があります。割り当てボタンをおした時に、選択頂点に割り当てるウェイト値をここで指定します。

Assigning verts to a Group

(頂点のグループへの割り当て)



アクティブなグループへのウェイトの割り当て

1. 一覧から対象グループを選びます。
2. マウスでグループに追加したい頂点を \diamond Shift RMB \square 選択します。
3. 割り当て (Assign) ボタンを LMB \square クリックするか、CtrlG \gg 選択をアクティブグループに追加 (Add Selected to Active Group) を使います。

グループに頂点を追加するには次のようにします:

1. グループ一覧からグループを選び、アクティブグループにします (①)
2. 3D ビューポートで、グループに追加したい頂点をすべて \diamond Shift RMB \square 選択します。
3. ウェイト値を設定します。この値は選択されているすべての頂点に割り当てられます (②)
4. Assign(割り当て) ボタンを LMB \square して、選択中の頂点をアクティブグループに、指定したウェイトを使って割り当てます (③)

なおウェイトの割り当ては、ロックされた頂点グループには行えません。この場合割り当てボタンは灰色になります。

割り当ては追加式

Assign(割り当て) ボタンは、現在選択中の頂点だけをアクティブなグループに追加します。グループにすでに割り当てられている頂点はグループから取り除かれませんが、頂点は複数のグループに割り当てられることも忘れないでください。

Checking assignments

(割り当ての確認)

選択した頂点が頂点グループに追加されたことを確認するために、deselect(選択解除)ボタンを試せます。頂点の選択が解除されなければ、おそらく Assign(割り当て) ボタンを押すのを忘れていました。ただ、割り当てはいま支障なく行えます。忘れないでください: 選択した頂点にはすべて、「Weight:」入力欄に表示されている通りのウェイトが割り当てられます!

Removing assignments from a Group

(グループからの割り当ての削除)

グループから頂点を取り除くには以下のようにします:

1. 一覧からグループを選びます (アクティブグループにします)
2. そのグループから取り除きたい頂点をすべて \diamond Shift RMB \square します
3. Remove ボタンを LMB \square クリックします



なお、ウェイト割り当ての削除はロックされた頂点グループには行えません。この場合削除ボタンは灰色になります。

Using groups for Selecting/Deselecting

(グループを選択/選択解除に使う)

グループに割り当て済みの頂点すべてを、簡単に選ぶことができます:

1. (任意)A を一度か二度押して、全頂点の選択を解除します。
2. グループ一覧からグループを選びます (アクティブグループにします)。

3. Select (選択) ボタンを LMB  クリックすると、アクティブグループに割り当てられている頂点が選択され、3D ビューポートでハイライトされます。
4. 代わりに Deselect (選択解除) ボタンを LMB  クリックすると、アクティブグループに割り当てられている頂点の選択が 3D ビューポートで解除されます。

選択/選択解除は追加式

3D ビューですでに頂点を選択中なら、グループの頂点を選択すると、選択済みの頂点を選択したまま、グループの頂点が追加されます。逆もまた同様で、頂点グループの頂点の選択解除は、グループに割り当てられている頂点だけの選択を解除し、その他の頂点はすべて選択されたままにします。

Finding ungrouped verts

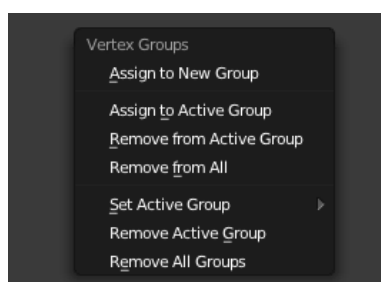
(グループに入っていない頂点を探す)

グループに入っていない頂点は次のようにして探せます:

1. A を一度か二度押して、全頂点の選択を解除します。
2. 3D ビューポートのフッターで: Select (選択) » Ungrouped Verts (非グループ頂点)

Keyboard Shortcuts

(キーボードショートカット)



頂点グループ ポップアップメニュー

編集モードで CtrlG を押すと、頂点のグループへの追加/グループからの削除メニューのショートカットを使えます。ポップアップメニューは次の明らかな機能を備えています:

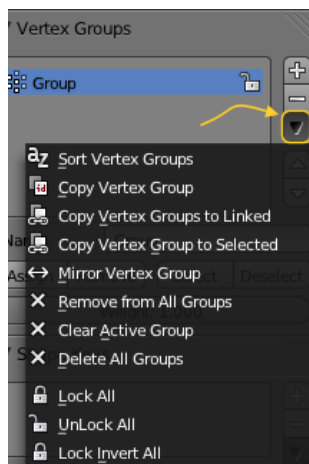
- Assign to New Group (新規グループに割り当て)
- Assign to Active Group (アクティブグループに割り当て)
- Remove from Active Group (アクティブグループから除外)
- Remove from All (すべてから除外)

次の機能はここに配置されるべきではなく、Blender の将来のバージョンでは取り除かれるかもしれません:

- Set Active Group (アクティブグループの設定)
- Set Remove Active Group (アクティブグループの削除)
- Set Remove All Groups (すべてのグループの削除)

Vertex Group Management

(頂点グループの管理)



頂点グループパネルのドロップダウンメニュー

プルダウンメニュー内に、頂点グループはもう少し複雑な機能を備えています。このメニューには、頂点グループパネルの右端にある、濃い灰色の下矢

印 アイコンをクリックしてアクセスできます。

プルダウンメニューの次の機能は割り当て済みの頂点に対して実行されます:

Sort Vertex Groups (頂点グループをソート)

頂点グループをアルファベット順に並べます

Copy Vertex Group (頂点グループをコピー)

アクティブな頂点グループのコピーを新規グループとして作ります。新しいグループの名前は、元のグループの名前の後ろに「_copy」をつけたものになります。そして、元の頂点グループにあるものと正確に同じウェイトを持つ同じ頂点への関連付けを持ちます。

Copy Vertex Groups to Linked (リンクしているものに頂点グループをコピー)

同じメッシュデータを使う、リンクされたオブジェクトのすべて(このデータの全ユーザー)に、このメッシュの頂点グループをコピーします。

Copy Vertex Group to Selected (選択したものに頂点グループをコピー)

インデックスの一致する(変形以外の編集がされていないメッシュのコピーは通常は一致します)他の選択オブジェクトに、すべての頂点グループをコピーします。

Mirror Vertex Group (頂点グループをミラー反転)

すべての頂点グループをミラー反転します。ウェイトや名前を反転し、選択頂点のみを編集、両側が選択されていれば反転し、そうでなければ選択していない頂点からコピーします。備考:この機能は今後のリリースで完全に作り直す予定です(ドキュメント化します)。

Remove from All Groups (全グループから取り除く)(ロックしたグループには無効です)

全グループから選択中の頂点の割り当てを解除します。この操作の実行後、頂点はもうどの頂点グループにも含まれなくなります。

Clear Active group (アクティブグループを消去)(ロックしたグループには無効です)

アクティブなグループから、割り当てられている頂点をすべて取り除きます。グループは空になります。なお、頂点は依然としてオブジェクトの他の頂点グループに割り当てられている可能性があります。

Delete All Groups (全グループを削除)

オブジェクトから全頂点グループを取り除きます。

次の機能はロック状態だけを操作します:

Lock All

全グループをロック

Unlock All

全グループをアンロック

Lock_Invert All

グループのロック状態を反転

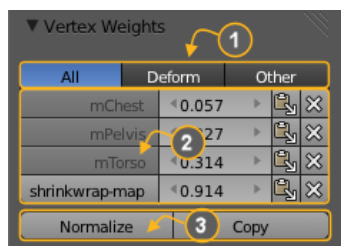
Hints

(ヒント)

- 同じメッシュデータを共有する複数のオブジェクトは、グループ名が保存された特別なプロパティを持っています。このためオブジェクトごとにグループの名前を変えることができますが、気をつけてほしいのは、頂点グループを削除すると、このメッシュを共有する全オブジェクトからグループが取り除かれることです。

Weight Editing

(ウェイトの編集)



頂点ウェイトパネル

以前 [頂点グループ](#) で説明されているように、頂点グループ内の各項目は [0.0, 1.0] の範囲のウェイト値を持ちます。Blender は Vertex Weights パネルを備えていて、メッシュの各頂点のウェイト値情報をここで取得(および編集)できます。どの頂点グループに、どのウェイト値とともに頂点が割り当てられているのか、という情報です。

頂点ウェイトパネルは 3Dビューポートの右側のプロパティサイドバー (N) にあります。編集モード、またはウェイトペイントモードで選択頂点のマスキングが有効なときに (V) 利用できます。パネルはセクション分けされています

- 頂点グループのカテゴリ(①)
- ウェイトテーブル(②)
- 機能バー(③)

Vertex Group Categories

(頂点グループのカテゴリ)

実際のところ Blender の頂点グループに厳格なカテゴリはありません。技術的にはすべて同じように振る舞います。ただ、頂点グループの潜在的なカテゴリを二つ特定できます:

The Deform Groups

(変形グループ)

この頂点グループは ウェイトグループとも呼ばれます。アーマチュアボーンウェイトテーブルを定義するのに使われます。オブジェクトのすべての変形グループは、ウェイト値を通じてお互いに正確に関連付けられています。

厳密に言うと、メッシュの任意の頂点のすべての変形ウェイトの合計は、ちょうど 1.0 になります。Blender ではこの制約は少しゆるやかです(後述)。そうは言っても、変形グループはいつも、お互いに関連付けられていると考えるべきです。このため我々は頂点ウェイトパネルに制限をかけて、オブジェクトの変形ボーンのみを表示するフィルターを用意しました。

The Other Groups

(その他のグループ)

頂点グループの他の使い方はすべて Other カテゴリにまとめました。こうした頂点グループはシェイプキー、モディファイアなどの中で見つかります。このカテゴリにはほんとうによい名前がないので、簡潔さを保ち Other と名づけました。

The Weight Table

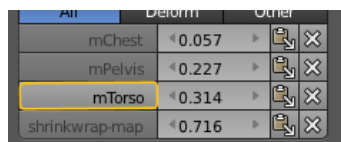
(ウェイトテーブル)

ウェイトテーブルは アクティブな頂点 に関連付けられたすべてのウェイトを表示します。なお、頂点を頂点グループに関連付けることは必須ではありません。この場合頂点ウェイトパネルは表示されません。

💡 アクティブな頂点

これは一番最近選んだ頂点です。メッシュ内で見やすいように、この頂点は常にハイライトされます。アクティブな頂点がウェイトを持たない場合、またはその時点で選択されているアクティブな頂点がない場合は、頂点ウェイトパネルは非表示になります。

ウェイトテーブルの各行には4つの有効な要素があります:

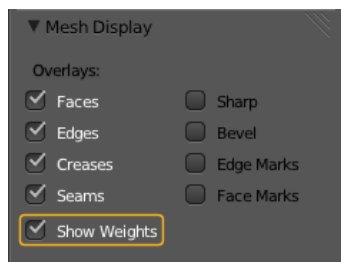


アクティブグループの変更

Set the Active Group

(アクティブグループの設定)

ウェイトテーブル内の頂点グループをどれでも選択するとすぐに、参照された頂点グループは新たなアクティブグループになります。



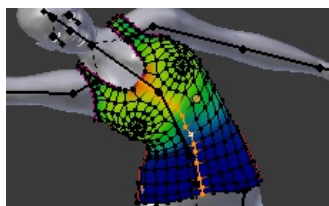
編集モードでウェイト表示を有効にする

Display Weights in Edit Mode

(編集モードでのウェイト表示)

編集モードに入っていれば、アクティブグループのウェイト値をメッシュ上に表示させることができます:

プロパティシェルフ(N)で Mesh Display(メッシュ表示) パネルを見つけて、Show Weights オプションを有効にしてください。こうすればアクティブな頂点グループのウェイトが、メッシュ表面に表示されるはずですが。

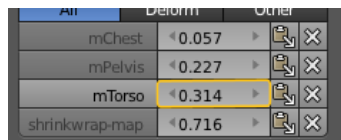


編集モードでのウェイト

Edit Weights in Edit Mode

(編集モードでのウェイト編集)

編集モードでウェイトマップを扱うのは非常に簡単になりました。メッシュのすべての編集オプションが利用でき、また、ウェイト値を編集した際のウェイトの変化の仕方を直接目で確認できるコントロールがあります。



ウェイト値の変更

Change a weight

(ウェイトの変更)

新たなウェイト値を自分で入力することも(数字をクリックして値を編集します)、LMB を押し続けながら左右にドラッグしてウェイト値を増減させることもできます。ウェイト値の周りに表示されている左右矢印を使って段階的にウェイトを変えることもできます。



ウェイトの貼りつけ

Paste a weight to other verts

(他頂点へのウェイトの貼りつけ)


貼りつけアイコンを LMB すると、アクティブな頂点のウェイトを、選択中のすべての頂点に送ることができます。ただしこのウェイトは、影響を受ける頂点グループにウェイト値をすでに持った頂点にだけ貼りつけられることに注意してください。



ウェイトの削除

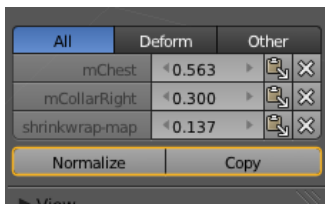
Delete a weight from a Group

(グループからのウェイトの削除)

削除アイコンを LMB  クリックするとすぐに、アクティブな頂点からそのウェイトを取り除きます。削除アイコンをクリックすると行全体が消えます。

The Function bar

(機能バー)



頂点ウェイトパネルの機能バー

機能バーには2つの機能があります:

Normalize (正規化)

アクティブな頂点のウェイトを正規化します。アクティブな頂点のウェイトがすべて、相対的なウェイトを維持しつつウェイトの合計が 1.0 になるように、再計算されます。

Copy (コピー)

アクティブな頂点用に定義されているすべてのウェイトを、選択中の全頂点にコピーします。したがって以前に定義されたウェイトはすべて上書きされます。

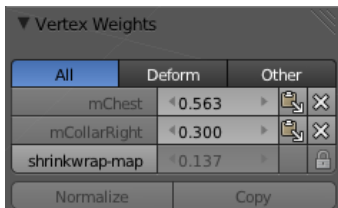


フィルターの設定は守られます

この機能はどちらも、ウェイトテーブルに現在表示されている頂点グループだけに作用することに注意してください。たとえば変形ウェイトだけが表示されていれば、Normalize (正規化) と Copy (コピー) は変形ボーンだけに作用します。

About locked Vertex Groups

(ロックされた頂点グループについて)



ロックされた頂点ウェイトパネル

ウェイトグループがロックされているときは常に、データを変更する機能はすべて無効化されます:

- 頂点ウェイトの正規化
- 頂点ウェイトのコピー
- アクティブな頂点のウェイト変更
- 選択頂点への貼りつけ

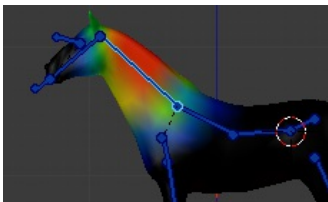


フィルター設定は守られます

たとえばすべての変形ウェイトグループのロックを外して、その他の頂点グループをすべてロックすれば、フィルター行から安全に Deform (変形) を選ぶことができ、ウェイトテーブルの利用可能な機能をすべて使うことができます。

Weight Paint in a nutshell

(ウェイトペイントの要約)




- Weight Paint(ウェイトペイント) モードにはモードメニュー (ホットキー=Ctrl⇧ Tab)から入れます。選択したメッシュオブジェクトに虹色のスペクトラムで少し陰影が付きま。
- 色はアクティブな頂点グループの各頂点に関連付けられたウェイトを視覚化します。青色はウェイトがつけられていません。赤色は最大までウェイトがつけられています。
- ウェイトのグラデーションに使われる色は、User Preferences(ユーザー設定) の System(システム) タブにある、Custom Weight Paint Range(ウェイトペイントの色の設定) を有効にするとカスタマイズできます。
- ウェイトはオブジェクトの頂点に、ウェイトブラシを使ってペイントすることで割り当てます。メッシュ上をペイントし始めれば、自動的にアクティブな頂点グループにウェイトが追加されます(必要なら新たな頂点グループが作られます)。



便利なキーボードショートカット

ショートカットを使えばウェイトペイント作業はもっとはかどるでしょう:


ウェイトカラーピッカー

Ctrl LMB  は、現在のウェイト値をクリックした頂点のウェイト値に変えます。

ブラシサイズの変更

F を押して新たなブラシのサイズをドラッグして決めます

線形グラデーションを作る

Alt LMB  を押してドラッグ

放射状グラデーションを作る

AltCtrl LMB  を押してドラッグ

クリッピングボーダーを描く

AltB を押してクリッピングボーダーをドラッグし、3D ウィンドウで見えたままにしたい部分を選択します。その後、この範囲内だけを描画することができます。クリッピングボーダーを取り除くには AltB をもう一度押してください。

Weight Paint Mode

(ウェイトペイントモード)

頂点グループは多数の関連した頂点を持つことができ、すなわち多数のウェイトを持つことができます(ウェイトは割り当てた頂点ごとにひとつ)。ウェイトペイントは多数のウェイト情報を直感的な方法で管理する方法のひとつです。ウェイトペイントは主にメッシュのリギングに使われ、頂点グループは関連したボーンがメッシュに与える影響を定義するために使われます。また、パーティクルの生成や髪の毛の密度、多数のモディファイア、シェイプキーなどの調整にも使います。

基本ルールは:ウェイト情報はウェイトブラシのセットを使って、文字通りメッシュ本体の上にペイントされます。ペイントは常に色と結びつけられるため、ウェイト色コードの定義も必要です。

The weighting Color Code

(ウェイト色コード)

ウェイトは寒色/暖色系を使って視覚化されます。影響度の低い(ウェイトが0.0に近い)領域は青色(寒色)で描かれ、影響度の高い(ウェイトが1.0に近い)領域は赤色(暖色)で描かれます。この間にある影響度はすべて、値に応じた虹の色(青、緑、黄、オレンジ、赤)で描かれます。

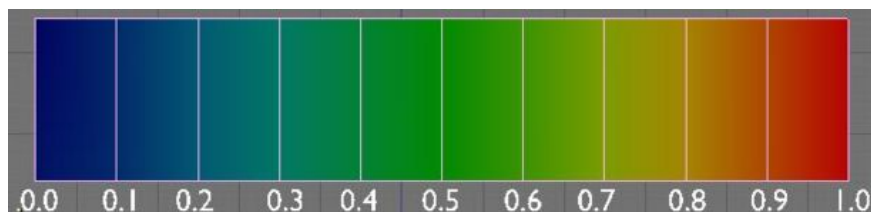
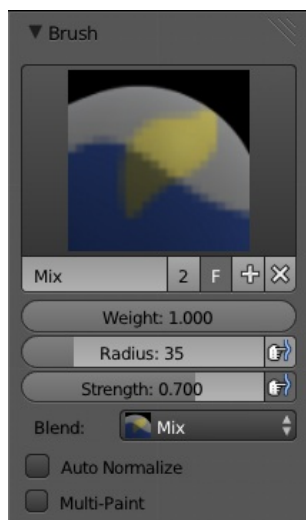


図3: カラースペクトラムとそのウェイト

上述のカラーコードに加えて、Blender は(オプションとして)参照されていない頂点用に特別な視角表記を行います。これらの頂点は黒で描かれます。したがって参照されている領域(寒色/暖色で描画)と、参照されていない領域(黒で描画)を同時に見ることができます。ウェイトの誤りを探すのに実際に非常に役立ちます。

Brushes

(ブラシ)



ツールシェルフにある ブラシ パネル

ペイントには塗装用ブラシが必要ですが、ウェイトペイントモードで操作中は、Blender はツールシェルフにブラシパネルを備えています。ブラシパネルの一番上にある大きなブラシアイコンをクリックすると、事前定義されたブラシプリセットが見つからうでしょう。必要なら独自のプリセットを作ることができます。利用可能なプリセットとカスタムプリセットの作成について、この下で説明します。

The main brush properties

(主要なブラシプロパティ)

以下はもっとも重要で、頻繁に編集するプロパティです:

Weight(ウェイト)

ブラシが使うウェイト(色)です。ただ、選択したブラシのブレンドモード(後述)によって、ウェイト値を頂点グループに適用する方法は変わります。

Strength(強度)

ブラシストロークごとに適用される ペイントの量 です。これが正確に意味するところは、ウェイト同様、ブラシのブレンドモードによって変わります。

Radius(半径)

この半径は、ブラシの作用する領域を決めます。なお、ペイント中にブラシの半径をキーボードショートカットで変更することもできます。好きなときにただ F を押せば、マウスドラッグでブラシ半径を増減できます。新しい設定を使うためには、最終的に LMB をクリックしてください。もしくはいつでも Esc キーを押して、現在の設定に戻すことができます。

Blend mode(ブレンドモード)

ブラシのブレンドモードは、ペイント中にどの方法でウェイト値を頂点グループに適用するかを決めます。Blender には7種類のブレンドモードがあります:

- **Mix(ミックス)**: このブレンドモードのウェイト値は、メッシュの同じ位置を十分長くペイントしたときに、最終的にたどりつく 目標ウェイトを定義します。Strength(強さ)は目標ウェイトにたどりつくのに必要なストロークの回数を決めます。なお Strength = 1.0 のときは目標ウェイトがただちにペイントされ、ウェイト = 0.0 のときはブラシは何もしません。
- **Add(加算)**: このブレンドモードでは、指定ウェイト値が頂点ウェイトに 足されます。Strength はストロークごとに足されるウェイト量を決めます。ただし、ブラシは 1.0 を越えるウェイト値をペイントしません。
- **Subtract(減算)**: このブレンドモードでは、指定ウェイト値が頂点ウェイトから 差し引かれます。Strength はストロークごとに引かれるウェイト量を決めます。ただし、ブラシは 0.0 を下回るウェイトをペイントしません。
- **Lighten(明るくする)**: このブレンドモードの指定ウェイト値は、Mix ブレンドモードによく似た 目標ウェイトとして解釈されます。ただ、目標ウェイトを下回るウェイトだけが作用します。目標ウェイトを上回るウェイトは変化しません。
- **Darken(暗くする)**: このブレンドモードの指定ウェイト値は、Lighten ブレンドモードによく似ています。ただ、目標ウェイトを上回るウェイトだけが作用します。目標ウェイトを下回るウェイトは変化しません。
- **Multiply(乗算)**: 頂点ウェイトと指定ウェイト値を乗算します。Subtract に少し似ていますが、取り除かれるウェイトの量に、ウェイト値そのものが影響します。
- **Blur(ぼかし)**: 近隣頂点のウェイトを滑らかにしようとします。このモードではウェイト値は無視されます。Strength はぼかしの適用効率を決めます。

Normalize Options

(正規化のオプション)

Blender は、影響下のすべての頂点グループの自動正規化に関するオプションも備えます:

Auto Normalize(自動正規化)

ペイント中、変形する頂点グループのウェイトが、合計で1までしか加算されないことを保証します。このオプションを無効にすると、どの頂点も 0.0 から 1.0 の間の任意のウェイトを持つことができます。ただしそうすれば、頂点グループがキャラクターアニメーション用の変形グループとして使われるときに、Blender は常に互いに関連したウェイト値の解釈を行います。これは Blender が、すべての変形ボーンに対して常に正規化を行うという事です。実際のところ厳格に正規化を保つ必要はなく、それ以上に正規化されたウェイトはまったくアニメーションに影響しないはずで

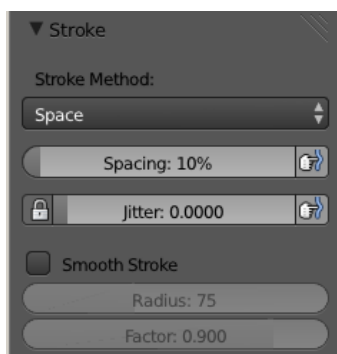
Multi-Paint

選択した頂点グループすべてに、同時にペイントします。アーマチュアを対象にした際には有用で、複数のポーズボーンを選ぶことで複数の頂点

グループを選ぶことができます。

The Brush stroke definition

(ブラシストロークの定義)



Stroke パネル

Stroke Method (ストローク方法)

- **Airbrush**(エアブラシ): マウスを押し続ける間、ペイント効果を適用し続けます (スプレー)
- **Space**(間隔): ブラシの適用を指定間隔ごとに制限します (下記参照)
- **Dots**(ドット): マウスが一定距離を動くたびにペイントします

Rate (レート) (Airbrush 専用)

エアブラシのペイント間隔

Spacing (間隔) (Space 専用)

ブラシの適用を指定間隔ごとに制限します

Jitter (ジッター)

ペイント中にブラシ位置を小刻みに動かします

Smooth Stroke (ストロークの補間)

ブラシがマウスより遅れて動き、より滑らかな経路をたどります

Radius (半径)

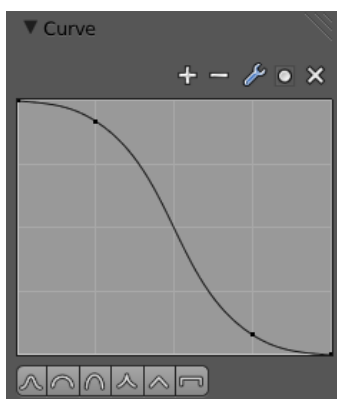
ストロークが継続しているとみなす最小距離です

Factor (強さ)

値が高いほど滑らかなストロークになります

The brush Falloff curve

(ブラシの減衰曲線)

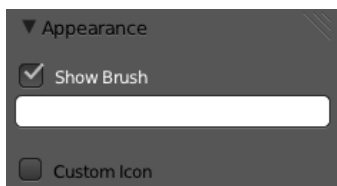


曲線パネル

ブラシの減衰エディタを使って、ブラシの特性を大幅に変えることができます。わかりやすく直感的に使えます。

The brush appearance

(ブラシの見た目)



ブラシの見た目

Show Brush(ブラシ表示)

ブラシを円で表示します(デフォルトで有効です)

Color setter(色)

ブラシの円の色を決めます

Custom icon(カスタムアイコン)

ブラシのカスタムアイコンを使えるようにします

Brush presets

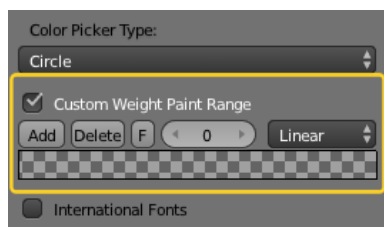
(ブラシのプリセット)

Blender にはブラシのプリセットがいくつか備わっています(各モードの説明は [主要なブラシプロパティ](#) をご覧ください):

- **Mix, Draw, Brush:** 強さの変化やブラシの減衰を使ってブラシウェイトを描くなら、ミックス混色モードを使います
- **Add:** 加算混色モードを使います
- **Subtract:** 減算ブレンドモードを使います
- **Lighten:** 明るくするブレンドモードを使います
- **Darken:** 暗くするブレンドモードを使います
- **Multiply:** 乗算ブレンドモードを使います
- **Blur:** ぼかしブレンドモードを使います

Customizing brush color space

(ブラシの色空間のカスタマイズ)



Blender ではウェイトペイントの色に使う色の範囲をカスタマイズできます。好みに応じてカラーバンドを定義できます。たとえばまったくの白黒(maya のウェイトペイントと同じもの)にすることもでき、アルファ値を使うことさえできます。

カスタマイズは User Preferences(ユーザー設定) の System(システム) タブで行えます。

Selection Masking

(選択部分のマスキング)

複雑なメッシュでは、ウェイトペイントモードですべての頂点をペイントすることが簡単ではないことがあります。メッシュの小さな範囲だけをペイントして他はさわらずにおいておけたらと思いませんか。そこで 選択部分のマスキング の登場です。このモードを有効にすると、ブラシは選択された頂点か面だけをペイントします。オプションは 3D描画域のフッターメニューバーから利用できます(黄色の枠で囲まれたアイコンです):



選択面のマスキング(左のアイコン)と 選択頂点のマスキング(右のアイコン)を選べます。

Select モードには ウェイトペイント モードに勝る利点があります:

1. モディファイアがアクティブなときにも、元のメッシュの辺が描かれます
2. 面を選択して、その面の頂点にペイントを制限できます。
3. 選択ツールは次のものを含みます:

Details about selecting

(選択操作の詳細)

次の標準的な選択操作がサポートされています:

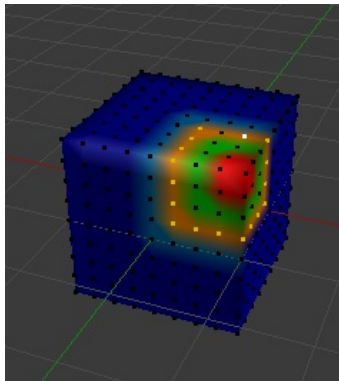
- RMB - 一枚の面。複数選択には Shift RMB を使います
- A - すべて(All)の面を選択、または選択の解除
- B - ボックス(Box)選択
- C - 円(Circle)選択
- L - (マウスカーソルの下にある)つながった(Linked)ものを選択
- CtrlL - つながったものを選択
- CtrlI - 選択の反転(Inverse)

変形(Deform)グループの選択

アーマチュアをとまなう変形ボーンにウェイトペイントをしているときは、対応するボーンを選ぶと変形グループを選べます。ただし選択部分のマスキングの実行中は、この選択頂点グループのモードは無効です!

Vertex Selection Masking

(選択頂点のマスキング)

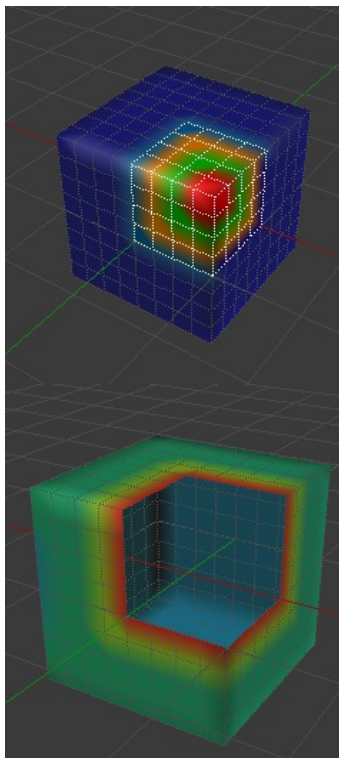


このモードでは一つ以上の頂点を選んで、選択部分にだけペイントすることができます。選択されていない頂点はすべて、意図しない変更から保護されます。

備考: このオプションは V キーで切り替えることもできます。

Face Selection Masking

(選択面のマスキング)



選択面のマスキングを使うと、選択した面にウェイトペイントツールを制限することができます。選択頂点のマスキングに非常によく似ています。

Hide/Unhide Faces

(面の表示/非表示)

編集モードでキーボードショートカット H を使えば、選択した面を隠すこともできます。残された見えている面にペイントし終わったら、AltH を使って隠した面を再表示できます。

Hide/Unhide Vertices

(頂点の表示/非表示)


選択頂点のマスキングモードでは、選択面を直接非表示にすることはできません。ただし、うまいやり方があります:

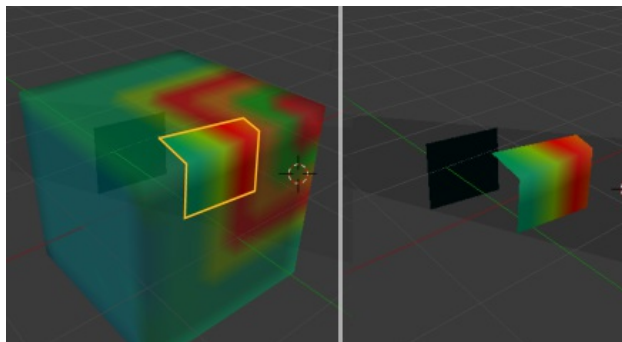
1. まず、選択面のマスクモードに入ります
2. 隠したい領域を選択し、面を隠します(上述のとおり)
3. 選択頂点のマスキングモードに戻ります

このとき、隠れた面に属する頂点は隠れたままです。

The Clipping Border

(クリッピングボーダー)

さらにペイント領域を制限するために、クリッピングボーダーを使えます。AltB を押して矩形の範囲を LMB -ドラッグします。選択範囲は関心領域として「切り離されます」。3D ウィンドウの残りは隠されます。

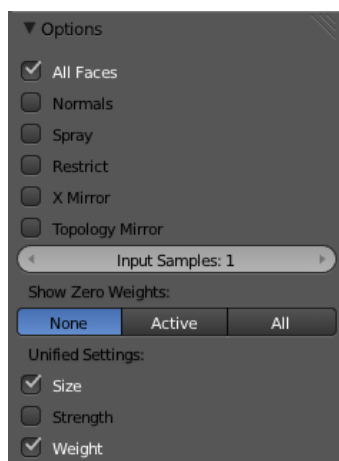


AltB をもう一度押せば、メッシュ全体を再び表示できます。

ボーダー選択、ウェイトグラデーション、それに当然ながらブラシのストロークといったビューを使うウェイトペイントツールはすべて、このクリッピングに従います。

Weight Paint Options

(ウェイトペイントのオプション)



ウェイトペイントのオプションは、ブラシの全体的な挙動を変更します:

All faces (すべての面)

オフにすると、カーソルの置かれた面に属する頂点だけをペイントします。複雑なメッシュの、メッシュ内では実際には距離があるのに、近くに見える面をペイントするとき便利です。

Normals (法線)

頂点の法線がペイント範囲を決めます (補助します)。光でペイントしたような効果を生みます。

Spray (スプレー)

マウスを動かすたびにウェイトを蓄積します。

Restrict

選択した頂点グループに属する頂点 (ウェイトがゼロでも) への、ペイントの影響を制限します。

X-mirror (X-ミラー)

`.R/.L` あるいは `_R/_L` のような拡張子を持つ、対称の名前を持つ一組のグループに、鏡映ペイントを行うためのオプションです。グループが鏡映された相手を持たなければ、アクティブなグループ自身に、対称にペイントします。命名規則について詳しくは [アーマチュアの編集: 命名規則](#) をご覧ください。アーマチュア/ボーン用の規則がここでも同じように使われます。

Topology Mirror (トポロジーミラー)

メッシュの両側が鏡映されたトポロジーを持つときに、トポロジーに基づくミラーリングを使います。

Input Samples (入力のサンプル)

...

Show Zero Weights (ウェイト0を表示)

参照されていないウェイトがゼロの領域を黒で表示します (デフォルト)

- None - 表示しません
- Active - アクティブグループに属していれば表示します
- All - 所属グループとは無関係にすべて表示します

Unified Settings (共通設定): ブラシの Size (サイズ)、Strength (強さ) および Weight (ウェイト) をブラシ別の設定ではなくブラシ間で共通の設定にします。

- Spray(スプレー): 常に塗り続けます(クリックのたびに1回ストロークを描くのとは対照的な挙動です)
- Restrict(制限): アクティブなウェイトグループですでにウェイトがつけられた頂点にだけ、ペイントします(新しいウェイトは一切作られません。既存のウェイトだけが編集されます)
- x-mirror(Xミラー): 左右対称に描きます。なおこのオプションは、人物の対称平面が z-y のときだけ動作します(人物が y 方向を見ているとき)。

Weight Paint Tools

(ウェイトペイントツール)



Blender はウェイトペイントの支援ツールを一式備えています。このツールはウェイトツールパネルにあります。

ウェイトペイントツールについて詳しくは [ウェイトペイントツール](#) のページをご覧ください。


Weight Painting for Bones


(ボーン用のウェイトペイント)

ウェイトペイントの用途としておそらくもっとも多いのがこの使い方です。ボーンを動かすと関節周りの頂点も同様に動きますが、間接周りの皮膚の伸び縮みを真似るため、移動は少しだけにすべきです。ボーンが回転したとき少ししか動かないように、関節周りは「軽い」ウェイト(10-40%)で塗ってください。アーマチュアに自動的にウェイトを割り当てる方法がありますが([スキニング \(eng\)](#) をご覧ください)、自分で行うこともできます。はじめから行うには、以下の手順を参照してください。自動的に割り当てたウェイトを編集するには、手順の中盤(註釈のあるところ)まで飛ばしてください:

- アーマチュアを作ります。
- アーマチュアのボーンが動いた時に変形させたいメッシュを作ります。
- このメッシュを選択した状態で Armature モディファイアを使います (Modifiers (モディファイア) パネル)。アーマチュアの名前を入力してください。

自動的に割り当てたウェイトを編集するにはここから始めてください。

- 3D View でアーマチュアを選び、ポーズ モードに入ります (Ctrl+ Tab または 3D ビューウィンドウのモード選択から)。
- アーマチュア内の目的のボーンを選びます。
- RMB  でメッシュを選び、そのまま ウェイトペイント モードに入ります。選択ボーンの動きがメッシュに影響するウェイト(度合い)に応じて、メッシュに色がつかます。最初はすべてがウェイトゼロを示す青色になります。
- ウェイトをペイントします。ボーンの周りのメッシュは(通常は)赤色にすべきです。ボーンから遠ざかるほど虹のように段階的に、青色に変化させます。

ウェイトペイントをしながら、上述のとおりアーマチュアは ポーズ モードに残したまま、別のボーンを RMB  で選ぶことができます。これは選択ボーンと同じ名前を持つ頂点グループをアクティブにし、関連ウェイトを表示します。

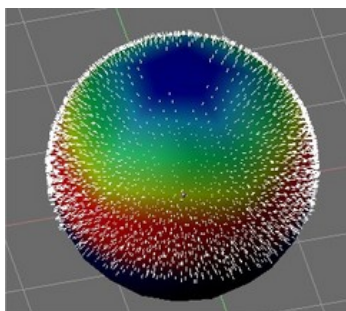
メッシュがボーンを覆っていると、メッシュがペイントされてボーンが見えなくなります。このときは Object Data (オブジェクトデータ) コンテキストの (Display (表示) パネルにある、X-Ray (レントゲン) をオンにします。同じパネルでは、ボーンを表示方法を Octahedron (八面体)、Stick (スティック)、B-Bone (Bボーン)、Envelope (エンベロープ)、Wire (ワイヤフレーム) から選べます。Draw Names (名前) をオンにすると、頂点グループに対応する選択ボーンの名前を確認できます。

メッシュをペイントすると、ボーン用の頂点グループが作られます。グループ外の頂点をペイントすると、その頂点は自動的に頂点グループに追加されます。

左右対称のメッシュと左右対称のアーマチュアがあれば、X-Mirror (Xミラー) オプションを使えます。有効にすると、ウェイトを鏡映コピーした鏡映グループが自動的に作られます。

Weight Painting for Particles

(パーティクル用のウェイトペイント)



ウェイトペイントされたパーティクル放射

ウェイトがゼロの面や頂点はパーティクルを生成しません。0.1のウェイトは、パーティクル量の10%をもたらします。このオプションは、実際に要求された数だけパーティクルを使う一方で、妥当なウェイトになるように配分調整して、表示されたパーティクルの合計数を「一定に保ちます」。メッシュに他より毛深い場所を作るには、頂点グループをウェイトペイントして、Object(オブジェクト) コンテキストの Particles(パーティクル) パネルにある Vertex Group(頂点グループ) 欄で、頂点グループの名前を呼んでください。

Weight Tools

(ウェイトツール)



Blender はウェイトペイントの支援ツールを一式備えています。このツールはウェイトツールパネルにあります。

The Subset Option

(使用部分オプション)

Subset(使用部分)パラメーターを備えたツールもあります。このパラメーターはツールが呼び出された後に、Operator パネルに表示されます。次の選択肢があります:

- Active Group(アクティブなグループ)
- Selected Pose Bones(選択中のポーズボーン)
- Deform pose Bones(変形ポーズボーン)
- All Groups(全グループ)

ツールはすべて、選択頂点マスキングや選択面マスキングとも連動します。これらのモードでは選択中の頂点や面にだけ処理が行われます。



Blend ツールについて

Blend ツールは「ペイント用の選択頂点マスキング」が有効なときだけ動作します。それ以外はボタンが灰色になり使えません。

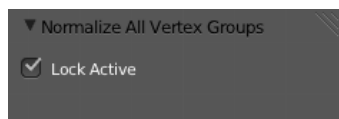
Normalize All

(すべてを正規化)

それぞれの頂点について、全頂点グループのウェイトの合計が必ず1になるようにします。すべての頂点グループを正規化しますが、ロックされたグループは例外で、ウェイト値がそのまま残されます。

Operator parameters

(パラメータ)



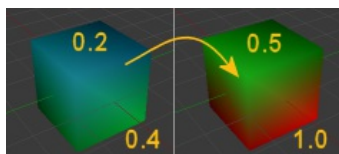
Lock Active(アクティブをロック)

アクティブグループの値は、他のすべてのグループの正規化中保持されます。

注意点: 現在このツールは、ロックされたグループを除くすべての頂点グループを正規化します。

Normalize

(正規化)



このツールはアクティブな頂点グループにだけ作用します。頂点はすべて相対的なウェイトを維持しますが、ウェイトの最高値が 1.0 になるように、ウェイト全体が増やされます。

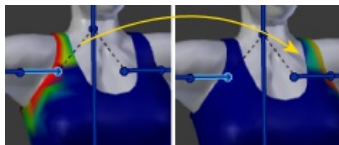
Operator parameters

(パラメータ)

なし

Mirror

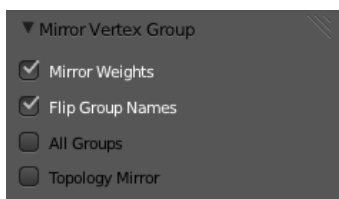
(ミラー)



メッシュの一方からもう一方へ、ウェイトを鏡映します (x軸に従う鏡映のみをサポートします)。ただ、ウェイトは逆側の対応するボーンウェイトグループには転送されないことに注意してください。鏡映は選択した頂点グループの中だけで発生します。

Operator parameters

(パラメータ)



Mirror Weights (ウェイトをミラー)

アクティブグループのウェイトを逆側に鏡映します。アクティブなウェイトグループだけに作用することに注意してください。

Flip Group Names (グループ名を反転)

左右の名前を入れ替えます。このオプションはグループだけを改名します。

All Groups (全グループ)

選択ボーンすべてに実行します。

Topology Mirror (トポロジーによるミラー)

100% の左右対称ではないメッシュ (近似ミラー) に鏡映します。



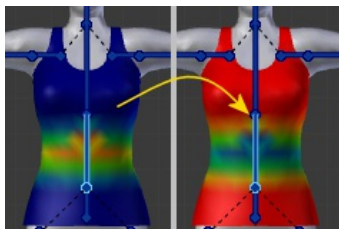
逆側のボーンへのミラー

(左右対称の人物の) 逆のボーンに鏡映されたウェイトグループを作りたい場合は、次のことが行えます:

- 目的の頂点グループを削除します (鏡映したウェイトを置く箇所)
- 元になるボーン頂点グループ (コピーしたいウェイトを持つグループ) のコピーを作ります
- このコピーの名前を、目的の頂点グループ (上の手順で削除したもの) の名前に変えます
- 目的の頂点グループを選び、ミラーツールを呼び出します (Mirror Weights (ウェイトをミラー) オプションだけを使います。メッシュが左右対称でなければ、トポロジーのオプションも使ってください)

Invert

(反転)



選択したウェイトグループの各ウェイトを、1.0 からウェイトを差し引いたものと置き換えます。

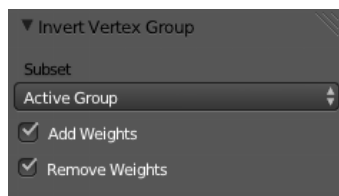
例:

- ウェイト 1.0 は 0.0 に変換されます
- ウェイト 0.5 は 0.5 のままです
- ウェイト 0.0 は 1.0 に変換されます

備考: パラメータの設定による挙動の変化を見てください。

Operator parameters

(パラメータ)

**Subset(使用部分)**

ツールの効果を指定した部分集合に限定します。定義方法については上述の [使用部分のオプション](#) をご覧ください。

Add Weights(ウェイトを追加)

反転前に、ウェイトを持たない頂点を追加します(この頂点のウェイトはすべて 1.0 に設定されます)

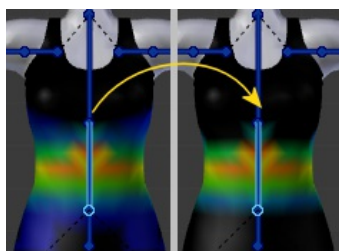
Remove Weights(ウェイトの削除)

反転後に、0.0 の頂点を頂点グループから取り除きます。

備考:ロックされた頂点グループは影響を受けません。

Clean

(クリーン)



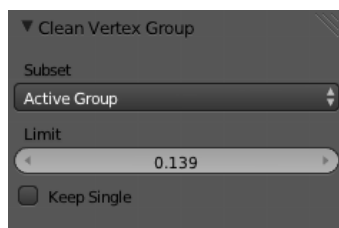
指定したしきい値を下回るウェイトを取り除きます。非常に低い(またはゼロの)ウェイトグループを取り除く際に便利です。

右の図では、切り捨て値(下のオプションをご覧ください)に 0.139 を使い、青い部分(左側)がすべて取り除かれています(右側)。

なお、この図では参照されていないウェイトが黒く表示されるように、Show Zero weights(ゼロウェイトを表示) オプションを Active(アクティブ)にしています。

Operator parameters

(パラメータ)

**Subset(使用部分)**

ツールの効果を指定した部分集合に限定します。定義方法については上述の [使用部分のオプション](#) をご覧ください。

Limit(制限)

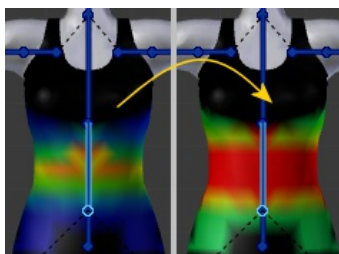
グループ内で保持される最小のウェイト値です。この値を下回るウェイトはそのグループから取り除かれます。

Keep Single(1つだけ残す)

制限値を下回る場合にも、各頂点に少なくともひとつウェイトを保持させて、まったく参照されない頂点(どの頂点グループにも属さない頂点)を作らないことを保証します。

Levels

(レベル)



選択したウェイトグループのすべてのウェイトにオフセットを足し、スケールします。このツールを使うとウェイトグループ全体の「温度」を上下させられます。

備考: 設定によらず、ウェイトには 1.0 を越えるか 0.0 を下回る値はセットされません。

Operator parameters

(パラメータ)



Subset(使用部分)

ツールの効果を指定した部分集合に限定します。定義方法については上述の [使用部分のオプション](#) をご覧ください。

Offset(オフセット)

頂点グループのすべてのウェイトに、[-1.0,1.0] の範囲内の値が足されます。

Gain(ゲイン)

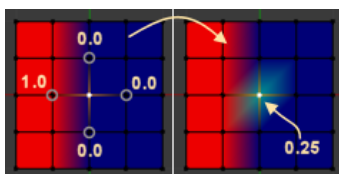
Subset(使用部分)のウェイトをすべて、ゲインで乗算します。値はスライダーのドラッグでは [-10.0, 10.0] の範囲に制限されますが、キーボードからは好きな値を入力できます。

備考: ゲインやオフセットに選んだ値がどのようなものであれ、各ウェイトの最終値は [0.0, 1.0] の範囲内に固定されます。したがってこのツールを使ってマイナスのウェイトやウェイトが 1.0 を越える過熱領域を得ることはできません。

Blend

(ブレンド)

選択頂点のウェイトを、隣接する非選択の頂点のものと混ぜ合わせます。このツールは選択頂点モードでのみ動作します。

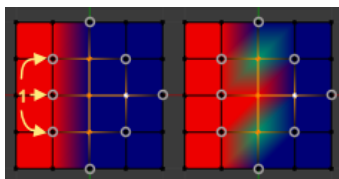


ツールが実際に行うことを理解するために、簡単な例を見てみましょう。選択頂点は隣接する4つの頂点につながっています(図の灰色の円)。隣接頂点はすべて、選択されていません。ここで、ツールはつながった頂点 および 非選択の頂点の平均ウェイトを計算します。この例では

$$(1 + 0 + 0 + 0) / 4 = 0.25$$

この値はパラメータ欄(後述)で指定した係数で乗算されます。

- 係数が 0.0 なら何も起きず、頂点の値は変わりません。
- 係数が 1.0 なら計算された平均ウェイトが使われます。ここでは 0.25 です。
- 0 から 1 まで係数をドラッグすると、元の値から計算された平均値まで、徐々に変化します。

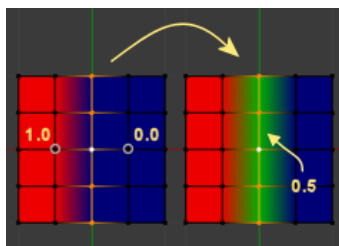


では、選択頂点に隣接する頂点を、ひとつを残してすべて選択したとき、何が起きるのか見てみましょう。ここでも、つながった頂点と非選択の頂点すべてに、灰色の円で印をつけました。ブレンドツールを呼び出して係数を 1.0 にセットすると、選択した頂点ごとに結果が違ってくるのがわかります:

- 一番上と一番下の選択頂点:
3つの非選択の頂点で囲まれており、この頂点の平均ウェイトは $(1 + 0 + 0) / 3 = 0.333$ です。このため色が薄い緑に変わりました。
- 中央の頂点:

非選択の頂点ひとつにつながっており、この頂点のウェイトは 1 です。したがってこの場合平均ウェイトは 1.0 になり、選択頂点の色は赤に変わりました。

- 右の頂点:
3つの非選択の頂点に囲まれており、この頂点の平均ウェイトは $(0+0+0)/3 = 0$ です。このため選択頂点の色はまったく変化しませんでした (ブレンドの適用前にすでに青でした)。

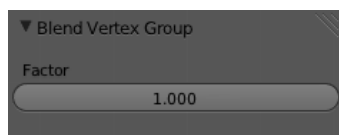


最後に実用的な例を見てみましょう。それからなぜこのツールをブレンドと呼ぶのかについても。この例では、中央のループ辺を選択しました。このループ辺を使って、左側領域を右側領域に混ぜ合わせたいのです。

- 選択頂点はすべて、非選択の隣接頂点を2つ持っています。
- 非選択頂点の平均ウェイトは $(1 + 0) / 2 = 0.5$ です。
- したがってブレンドの係数を 1.0 にしたとき、ループ辺は緑に変わり、最終的に冷たい側(右)を熱い側(左)にブレンドします。

Operator parameters

(パラメータ)



Factor(係数)

ブレンドの影響量で [0.0, 1.0] の範囲内の値をとります。係数を 0.0 にするとブレンドツールは何もしません。係数が 0 より大きければ、影響された頂点のウェイトが元の値から、つながった全頂点のウェイトおよび非選択頂点のウェイトの平均値に向けて、徐々に変わります(上の例をご覧ください)。

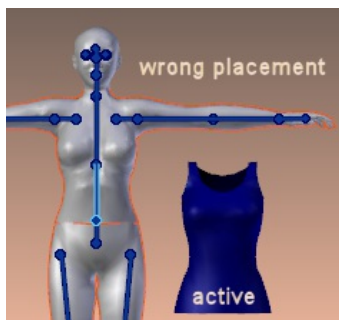
Transfer Weights

(ウェイト転送)

他のオブジェクトからアクティブオブジェクトの頂点グループに、ウェイトをコピーします。デフォルトではこのツールは選択されたオブジェクトにあるすべての頂点グループを目的のオブジェクトにコピーします。操作のやり直し (operator redo) パネルでツールの挙動を変えることができます (後述)。

Prepare the copy

(コピーの準備)





元になるオブジェクトをすべて先に選択してから、最後に目的のオブジェクトを選んでください(目的のオブジェクトはアクティブオブジェクトである必要があります)。

元のオブジェクトと目的のオブジェクトは同じ位置にあることが重要です。オブジェクトが隣り合っているとウェイト転送は動作しません。オブジェクトは別のレイヤーに置けますが、ツールを呼び出したとき、オブジェクトが必ず、すべて見えているようにする必要があります。

ここで、目的のオブジェクトがウェイトペイントモードになっていることを確認してください。

Call the tool

(ツールの呼び出し)

ツールシェルフを開いてウェイトツールパネルを探します。そこにあるウェイト転送ツールを呼び出してください。ツールははじめに元のオブジェクトから頂点グループをすべてコピーします。ただし、ツールには操作のやり直し(operator redo)パネルがあります。このパネルはツールの実行後にツールシェルフの一番下に現れます。やり直しパネルから、パラメータをあなたの要求に合うものに変えることができます(利用可能なパラメータについては後述します)。

Redo Panel Confusion

(やり直し操作によるパネルの混乱)

ウェイト転送後、操作やり直し(operator redo)パネルが使えるまま残ることに気づくかもしれません。パネルは、独自の操作やり直しパネルを持つ別の操作を行ったときに消えます。これは、頂点グループの変更後にウェイト転送を繰り返し行ったとき、混乱の元になります。このとき、見えたままになっている操作やり直しパネルを使うと、Blenderは最初にウェイト転送が呼ばれた直前の状態に、あなたの作業を巻き戻してしまうでしょう。

Workaround

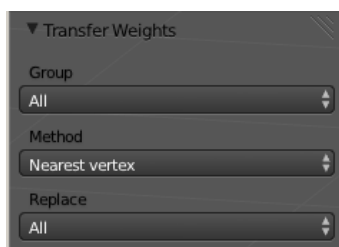
(回避策)

頂点グループに変更を行った後で再度ウェイト転送を呼び出すときはいつも、やり直しパネルがまだ使えたとしても、「ウェイト転送」ボタンを使ってください。ツールを最初に呼び出したところまで、ほんとうに変更をリセットしたい場合は別です。

Operator parameters

(パラメータ)

太字でデフォルトを示します:



Group(グループ)

- **Active(アクティブ)**: アクティブオブジェクトのアクティブグループにだけコピーします。このオプションは、アクティブオブジェクトがアクティブな頂点グループを持つときにだけ動作します。それ以外はウェイト転送は何もしません。
- **All(すべて)**: 選択オブジェクトからアクティブなオブジェクトへ、すべての頂点グループをコピーします。

Method(方式)

- **Nearest vertex In face(面内の最近接頂点)**: 記述予定
- Nearest Face(最近接面): 記述予定
- Nearest vertex(最近接頂点): 記述予定
- Vertex Index(頂点のインデックス番号)(逐語的コピー、インデックスの数が同じメッシュにだけ動作します)

Replace(置き換え)

- **Empty(空)**: 頂点がグループ内にまだウェイトセットを持っていなければ、アクティブオブジェクトにだけウェイトをコピーします。

- **All(すべて)**: 元オブジェクトからグループをコピーする前に、目的の頂点グループにあった内容をすべて削除します

💡 ご注意!

ひとつの頂点グループが2つ以上の選択オブジェクトに含まれる場合、結果は選択オブジェクトが処理された順番に左右されます。ただ、処理の順番は影響を受けません。

Limit total

(合計を制限)

頂点ごとのウェイトグループの数を指定した上限値に制限します。制限を満たすまで、一番低いウェイトから先に取り除かれます。

ヒント:ひとつ以上ウェイトグループを選択したときにだけ妥当な動作をします。

Operator parameters

(パラメータ)

Subset(使用部分)

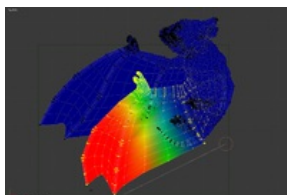
ツールの効果を指定した部分集合に限定します。定義方法については上述の [使用部分のオプション](#) をご覧ください。

Limit(制限)

各頂点で許可されるウェイトの最大数(デフォルトは4つ)。

Weight Gradient (wip)

(ウェイトグラデーション)



選択頂点にグラデーションツールを使う例

ウェイトの線形/放射状グラデーションを使うためのインタラクティブなツールです。ウェイトの段階的な塗りが難しい場合に役立ちます。

グラデーションツールの実行にはツールシェルフの Weight Tools(ウェイトツール)にあるボタンを使うか、ショートカットを使います:

- 線形: Alt LMB  を押してドラッグ
- 放射状: AltCtrl LMB  を押してドラッグ

ウェイトペイントの次のオプションは、グラデーションの制御にも使われます:

- Weight(ウェイト) - グラデーションは現在選択されているウェイトからゼロに向かって広がります。
- Strength(強さ) - ブラシと同じく、低い値は既存のウェイトとグラデーションの塗りを混ぜ合わせるのに使えます
- Curve(曲線) - ブラシの減衰曲線はグラデーションにも使われるので、グラデーションの混ぜ具合の調整に使えます

選択頂点のウェイトを非選択頂点のものと混ぜ合わせます。ヒント:このツールは選択頂点モードでのみ動作します。

Operator parameters

(パラメータ)

Type

- Linear
- Radial

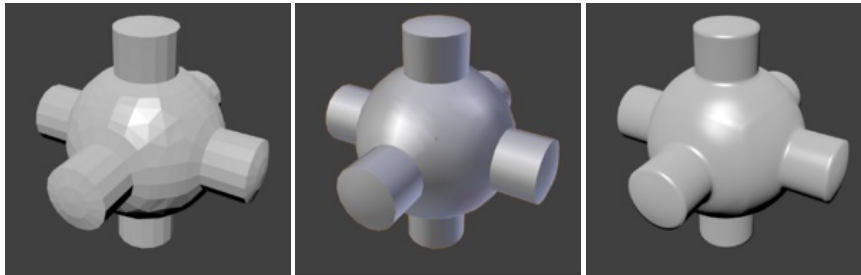
X Start

X End

Y Start

Y End

Mesh smoothing



例示のメッシュのフラットでスムーズにレンダリングされているものは、エッジ分割を使用し、Subdivision Surfaceを使用しています。エッジがレンダリングされる方法の違いに注目ください。

[Sample blend](#)

前のセクションで見たように、ポリゴンはBlenderの中核です。ほとんどのオブジェクトはポリゴンで表現され、本物の曲線状のオブジェクトは、多くの場合、ポリゴンメッシュで近似されます。画像をレンダリングするとき、これらのポリゴンが小さく平らな面の列として表示されることがあります。

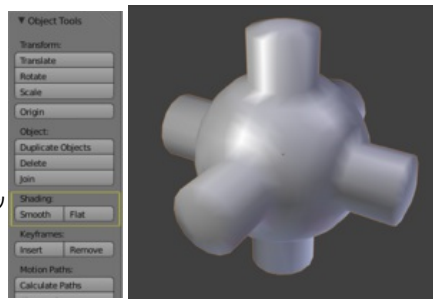
時には、これが望ましい効果をもたらすこともありますが、オブジェクトはきれいで滑らかな見た目であってほしいのが普通でしょう。このセクションでご覧いただくのは、オブジェクトを視覚的に滑らかにする方法、そして Auto Smoothフィルタを適用して迅速かつ容易に、同一のオブジェクト内でスムーズなポリゴンと切り子のポリゴンを組み合わせる方法です。

このページの最後のセクションでは、外観だけでなく、メッシュの形状を滑らかにして実現する方法を示しています。

Smooth shading

最も簡単な方法はオブジェクト全体にスムーズかファセットと同様のものを設定することです。メッシュオブジェクトを選択して、ObjectモードでTool ShelfにあるSmoothをクリックします。このボタンは押しっぱなしにはならず、あなたがジオメトリを追加または削除する際に再度、メッシュの各面に“smoothing”属性を割り当てせざるをえません。

オブジェクトの輪郭はまだ強く角張っていることに注意してください。スムーズ機能を有効にしても、実際にはオブジェクトのジオメトリは変更されません。表面の随所の陰影計算の方法を変更し、そこが滑らかな表面のような錯覚を与えます。Tool ShelfのShading panelにあるFlatボタンをクリックすると、シェーディングが元に戻り、上記の最初の画像の見え方に戻ります。



Same mesh smooth shaded

Smoothing parts of a mesh

また、どのエッジを滑らかにするかを選択することもできます。Edit modelに入って、いずれかの面を選択し、Smoothボタンをクリックしましょう。選択したエッジが黄色でマークされます。

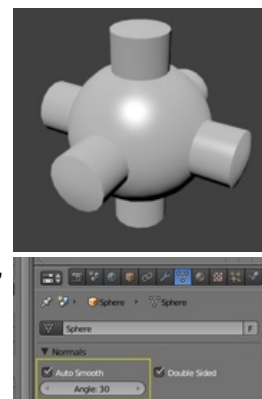
メッシュがEdit modelにあるときは、選択状態のエッジのみが“smoothing”属性を受けます。同様の方法でエッジを選択し、Flatボタンをクリックすれば、エッジをフラット (“smoothing”属性を削除する)として設定することができます。

Auto Smooth

上記手法だけの使用では、スムーズとソリッドの面をきっちりと組み合わせたものを作成することは難しい場合があります。回避策はありますが (例えば、選択して Y押すことでの面のオフセット分割など)、スムーズとソリッドの面を結合させるにはもっと簡単な方法があります。Auto Smoothの使用です。

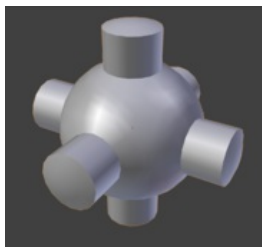
Auto smoothing は、Propertiesウィンドウ内のメッシュパネルで有効化することができます。モデル上の角度のうち、Angleボタンで指定した角度よりも小さいものが平滑化されます。ただしレンダリング時 (つまり3Dビューでなく) で、メッシュのその部分がsmoothに設定されている場合です。値を高くすると、生成される面が滑らかになります。一方、最も低い設定にすると、完全なソリッドに設定されたメッシュと同じになります。

メッシュ、またはいずれかの面がFlatとして設定されていると、Auto Smoothがアクティブ化されるときにそれらのシェーディングが変更されないことに注意してください: このようになっているのは、どの面を平滑化するのか、そしてAuto Smoothアルゴリズムの判断のよってオーバーライドされたくない面はどれか、といったことの制御を拡張できるからです。



Example mesh with Auto Smooth enabled

Edge Split Modifier



Edge Split modifier
enabled

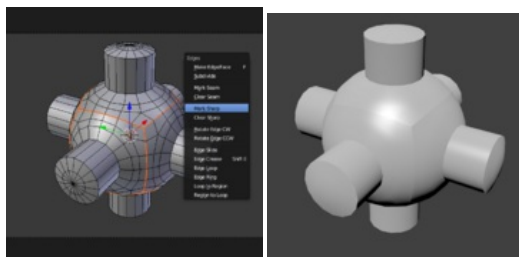
[Edge Split](#)モディファイアを使用すると、Auto Smoothに似た結果を得るとともに、分割すべきエッジを選択できるようになります。

Edge Angle

エッジ間の角度を使用して、どのエッジを分割するかを決定します。Auto Smoothに似ています。角度は0°から180°の間で設定できます。

Sharp Angles

シャープという印をつけたエッジを使用して、どのくらいの角度で分割するかを決定します。Edit modeで、エッジループのセットを選択することができます。例えば Alt+ Shift RMB を使用して、その後Edge menuでシャープとしてマーキングします (CtrlE>Mark Sharp。)



選択しているエッジループをMark sharpとして設定しておくと、メッシュがレンダリングされる方法を制御することができます。

Smoothing the mesh geometry

上記のテクニックは、メッシュ自体を変更せず、表示やレンダリングをされる方法だけを変更するものでした。見た目だけ表面が滑らかなメッシュを作るだけでなく、メッシュの形状を物理的に滑らかにする次のようなツールがあります:

メッシュ編集ツール

Edit modeで次のいずれかを適用できます:

- [Smooth](#)

選択しているコンポーネントをゆるめて、その結果スムーズなメッシュにします

- [Subdivide Smooth](#)

subdivideツールを使用した後、smoothパラメーターを調整し、結果をより有機的な形状にします。これは、細分化モディファイアを使用する時と似ています。

- [Bevel](#)

選択しているエッジを面取りして、鋭いエッジを平らに削ります

モディファイア

また、次のモディファイアのいずれか、または複数を使って、非破壊的にメッシュを滑らかにすることもできます:

[Smooth Modifier](#)

SmoothでのEdit modeツールのような働きをします。頂点グループを使用してメッシュの特定の部分に適用することもできます。

[Bevel Modifier](#)

Edit modeでの Bevel tool のような働きをします。ベベルは、角度のしきい値、またはエッジの重みの値で動作するように設定することができます。



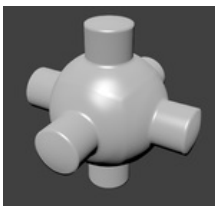
Subsurf

[Subdivision Surface Modifier](#)

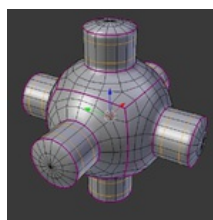
Catmull-Clark subdivisionは、滑らかな結果を生成します。鋭いエッジは、[subdivision creases](#)で定義するか、または特定のエッジを“sharp”に設定して、[EdgeSplit modifier](#) (set to From Marked As Sharp) をSubsurf modifierよりも前に追加することで定義できます。



Using creased edges, and resulting subsurf artifacts



Extra edge loops added



3D view showing creased edges (pink) and added edges loops (yellow)

Mesh Clean-up

(メッシュの掃除)

これらのツールは劣化したジオメトリをきれいにするのに役立ち、メッシュの欠けた部分を埋めます。

[edit] Fill Holes

(穴を埋める)

モード: 編集モード

メニュー: Mesh(メッシュ) » Clean up(クリーンアップ) » Fill Holes(穴を埋める)

大きな選択範囲を与えると、メッシュの穴を見つけて埋めます。

このツールと面の作成操作には、大きな違いが三つあります。

- 穴は自動検知されるので、穴の周囲の辺を自分で探して選ぶ必要がありません
- 一定数以上の辺を持つ穴は検知できません(例として四角形か三角形のものだけが埋まります)
- メッシュデータは周囲のジオメトリからコピーされます(UV、頂点色、多重解像度、すべてのレイヤー)。手動でこれらのデータを作るのは時間がかかるためです

[edit] Split Non-Planar Faces

(非平面を分割)

モード: 編集モード

メニュー: Mesh(メッシュ) » Clean up(クリーンアップ) » Split Non-Planar Faces(非平面を分割)

指定角度を越えて曲げられた平らでない面があれば、あいまいさをなくすため面分割します。

[edit] Delete Loose Geometry

(孤立を削除)

モード: 編集モード

メニュー: Mesh(メッシュ) » Clean up(クリーンアップ) » Delete Loose(孤立を削除)

接続されていない頂点と辺を取り除きます(設定によっては面も - デフォルトはオフです)。

[edit] Degenerate Dissolve

(大きさ0を溶解)

モード: 編集モード

メニュー: Mesh(メッシュ) » Clean up(クリーンアップ) » Degenerate Dissolve(大きさ0を溶解)

普通は望まないジオメトリを折り畳むか取り除きます。

- 長さのない辺
- 面積を持たない面(点上の面や狭い面)
- 面積を持たない、面のかど

Curves

(カーブ/曲線)



ベジエ曲線でできているロゴ

Curves/曲線と [Surfaces/曲面](#) は Blender オブジェクトの一種です。点の集まりではなく数学関数で表現されます。

Blender は [Bézier\(ベジエ\)](#) 曲線と [NURBS](#) 曲線および曲面のどちらも備えています。NURBS とは Non-Uniform Rational B-Splines (非均一有理 Bスプライン) の略です。どちらも制御点の集合で一つの「制御多辺形」を定義します。

ポリゴンメッシュに比較して曲線の一番の長所は、より少ないデータで定義され、モデリング中にメモリ消費やディスク使用量を抑えながらきれいな結果を生み出せることです。一方で、この処理方式ではレンダリング時間が増える可能性があります。

パスに沿う断面の [押し出し\(extruding\)](#) のようなモデリング技法は、曲線を使う場合にのみ利用できます。一方で、曲線を頂点レベルで操作するのはもっと難しくなります。正確な操作が必要なら、メッシュ編集式のモデリングをしたほうがよいでしょう。

ベジエ曲線は文字やロゴのデザインにもっともよく使われます。またアニメーションではオブジェクトの移動経路として、もしくはオブジェクトのプロパティを時間経過とともに変化させる Fカーブとして、広く使われます。

Tutorials

(チュートリアル)

[ベジエ曲線を使った鳥のロゴの作成](#)[複数曲線を使ったサーフェスの作成](#)[曲線を使ったメッシュ変形\(英文\)](#)

Curve Primitives

(曲線のプリミティブ)

Add Curve メニューにBlender は5つの曲線プリミティブを用意しています:

Bezier Curve (ベジエ曲線)

2つの制御点を持つ、開いた 2D ベジエ曲線を追加します。

Bezier Circle (ベジエ円)

閉じた円形の 2D ベジエ曲線を追加します(4つの制御点でできています)。

NURBS Curve (NURBS曲線)

4つの制御点と均一な knot を持つ、開いた 2D NURBS 曲線を追加します。

NURBS Circle (NURBS円)

閉じた円形の 2D NURBS 曲線を追加します(高さの制御点でできています)。

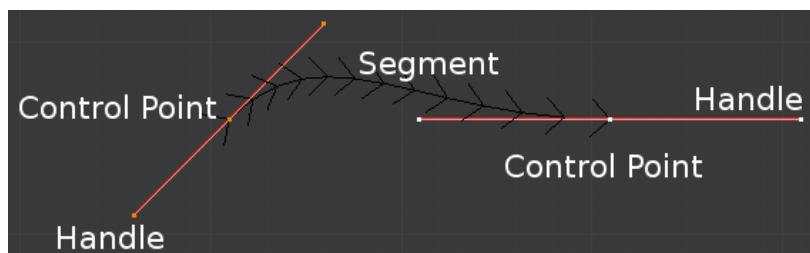
Path(パス/経路)

開いた 3D NURBS 曲線を追加します。5つの整列した制御点と Endpoint (終点) knot を持ち、CurvePath(曲線経路)設定が利用できません。

Bezier Curves

(ベジエ曲線)

ベジエ曲線の編集で中心になる要素が、制御点(Control Point)とハンドル(Handle)です。二つの制御点の間にはセグメント(実際の曲線)があります。下図では制御点はピンクの線の中央にあり、一方ハンドルはこの制御点から延長したものでできています。デフォルトではセグメント上の矢印はその向きと、オブジェクトを曲線に沿って動かすときの **相対的な** 速度と向きを表します。これは Speed Ipo をカスタム定義することで変更可能です。



編集モードのベジエ曲線

Editing Bezier Curves

(ベジエ曲線の編集)

ベジエ曲線は制御点とハンドルの位置を動かすと編集できます。

1. 曲線は、**ShiftA** で Add (追加)メニューを呼び出して、Curve (曲線) » Bezier (ベジエ) を選んで追加します。
2. **Tab** を押して 編集モードに入ります。
3. 制御点の一つを選んで動かします。制御点の位置の確定には **LMB** を、キャンセルには **RMB** を使ってください。
4. 次にハンドルの一つを選んで動かしてみてください。曲線の曲率が変わることがわかるはずです。

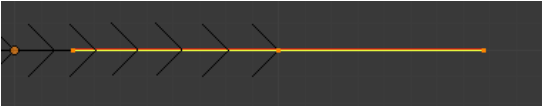



制御点をさらに追加するには

1. 隣り合う制御点を少なくとも二つ選びます。
2. **W** を押して Subdivide (細分化) を選びます。
3. 分割直後に **F6** を押して分割数を変更することもできます。

なお編集モードにいるときは、直接セグメントを選択できません。選択するには、動かしたいセグメントを構成するすべての制御点を選びます。

ベジエ曲線のハンドルは4種類あります。V を押すと一覧が表示されるので、項目を選ぶか対応するホットキーを押して選びます。ハンドルはメッシュの頂点と同じように回転、移動、拡大縮小、伸縮/膨張させることができます。

ベジエ曲線のハンドル種類

種類	ショートカット	用途	見た目
Automatic (自動)	VA	もっとも滑らかな結果になるように、Blender が自動的にハンドルの長さや方向を設定します。動かすと Aligned (整列) ハンドルに変換されます。	
Vector (ベクトル)	VV	ハンドルの両端が常に前のハンドルか後のハンドルを指し、そこから直線や鋭いコーナーでできた曲線や面を作ることができます。動かすと Free (フリー) ハンドルに変換されます。	
Aligned (整列)	VL	常に直線上に置かれ、鋭い角度を持たない連続した曲線を作ります。	
Free (フリー)	VF	ハンドルがそれぞれ独立しています。	

さらに VT ショートカットは Free (フリー) と Aligned (整列) ハンドルを切り替えるのに使えます。

Curve Properties

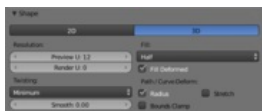
(曲線のプロパティ)

曲線のプロパティは Object Data コンテキストで設定できます (下図の青い部分)。



Shape

(形状)



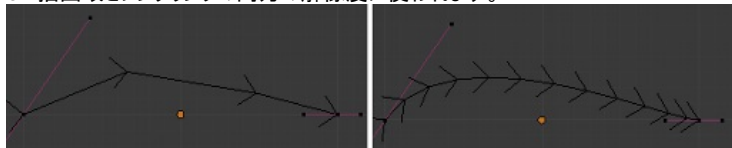
曲線の Shape パネル

2D/3D

デフォルトでは、新しい曲線は 3D に設定されています。つまり三次元空間のどこにでも制御点を置けるということです。曲線を 2D に設定して、制御点を曲線のローカルの XY 軸に制限することもできます。

Resolution (解像度)

対になる制御点間に生成される点の数を決めます。解像度の増減に応じて曲線の滑らかさが増減します。Preview U 設定は 3D 描画域での解像度を決め、Render U 設定は曲線のレンダリング時の解像度を決めます。Render U がゼロ (0) に設定されると、Preview U 設定が 3D 描画域とレンダリングの両方の解像度に使われます。



解像度 3 の曲線 (左) と 12 の曲線 (右)

Twisting (ねじれ)

3D 曲線には、曲線のローカルの XY 平面上にない制御点があります。この設定は曲線の法線に作用するねじりを曲線に与えます。曲線のね

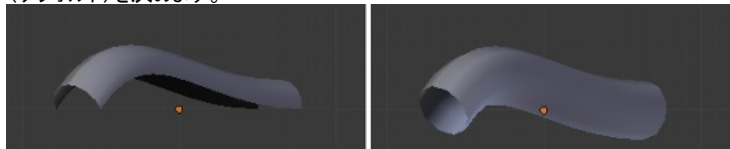
じりの計算方法は、ドロップダウンメニューで Minimum、Tangent、Z-Up オプションから選べます。



Minimumのねじれを持つ曲線(左)と Tangentのねじれを持つ曲線(右)

Fill(面張り)

Fill は、Bevel 処理をしたときの曲線の表示方法を決めます (Bevel の詳細は後述します)。Half(ハーフ、デフォルト)に設定すると、曲線は半分の円柱で表示されます。Fill Deformed(面を張り直す)オプションは曲線の Fill を シェイプキーやモディファイアの適用前に行うか後に行うか(デフォルト)を決めます。



Fill が Half (左)と Full (右)に設定された曲線

Path/Curve-Deform(パス/曲線変形)

主に曲線をパスとして使う場合や Curve Deform(曲線変形)プロパティを使う場合に利用されるオプションです。Radius(半径)、Stretch(ストレッチ)、Bounds Clamp(境界固定)は、オブジェクトが曲線をどのように利用するのか調整するオプションです。詳細は次のリンクをご覧ください。

[基本的な曲線編集についてもっと詳しく»](#)

[パス/経路についてもっと詳しく»](#)

[曲線変形についてもっと詳しく»](#)

Geometry

(ジオメトリ)



曲線のジオメトリパネル

Modification(変形)

Offset(オフセット)

デフォルトでは、Text オブジェクトは曲線として扱われます。Offset オプションは文字間の余白を変化させます

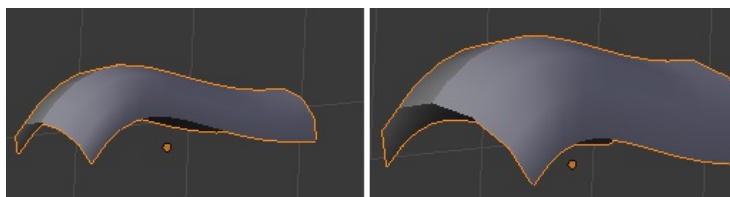
Extrude(押し出し)

ローカルの Z軸正負どちらの向きにも曲線を押し出します

Bevel(ベベル)

Depth(深度)

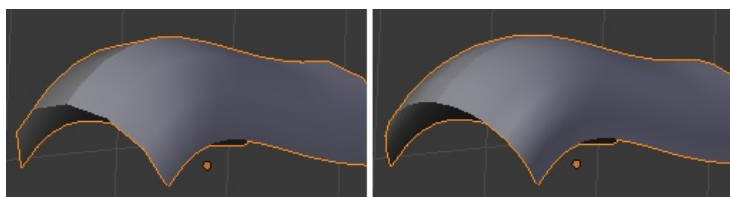
ベベルの大きさを変えます



Bevel の深さを変えた曲線

Resolution(解像度)

ベベルの滑らかさを変化させます

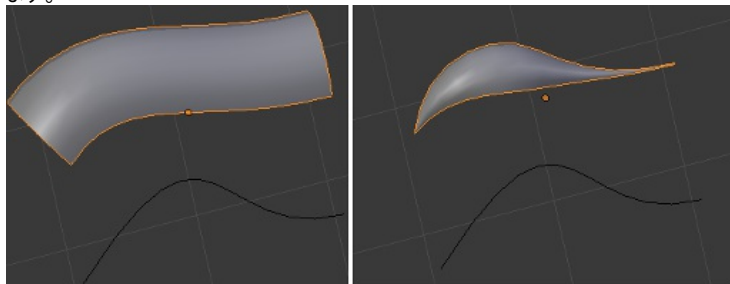


Bevel の解像度を変えた曲線

Taper Object(テーパオブジェクト)

曲線をテーパ加工して先細りにします。Taper Object の制御点を移動/拡大縮小/回転させることで、テーパ加工の比率を変えることもできます。Taper Object には他の曲線のみを設定できます。Taper Object のハンドルや制御点を編集すると、元のオブジェクトの形状が変わり

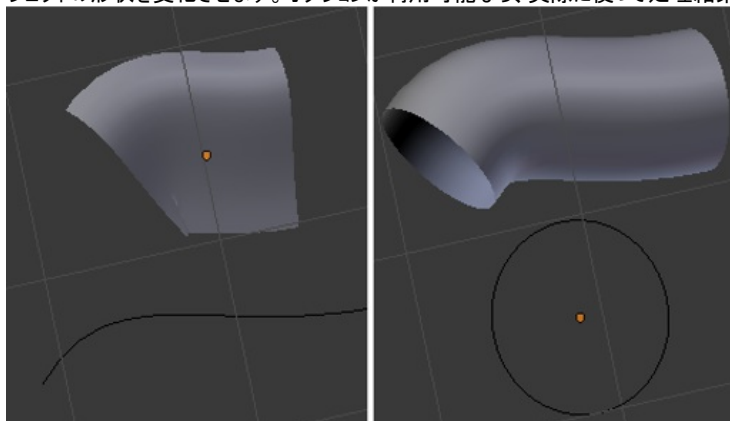
ます。



ベジエ曲線を Taper Object として使用前の曲線 (左) と使用後の曲線 (右)

Bevel Object (ベベルオブジェクト)

ベジエ曲線をベジエ曲線を Bevel Object として使ってベベル加工すると、通常は平面の形状になります。一方ベジエ円を Bevel Object として使うと円柱の形状になります。Bevel Object には他の曲線のみを設定できます。Bevel Object のハンドルと制御点を編集すると、元のオブジェクトの形状を変化させます。オプションが利用可能なら、実際に使って処理結果を確かめるのがよいでしょう。



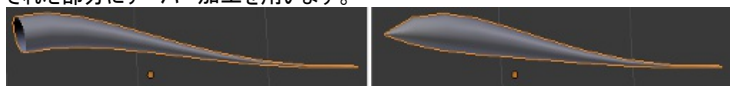
Bevel Object としてベジエ曲線を指定した曲線 (左) とベジエ円を指定した曲線 (右)

Fill Caps (端をふさぐ)

ベベル加工した曲線の端にふたをします

Map Taper (テーパーマッピング)

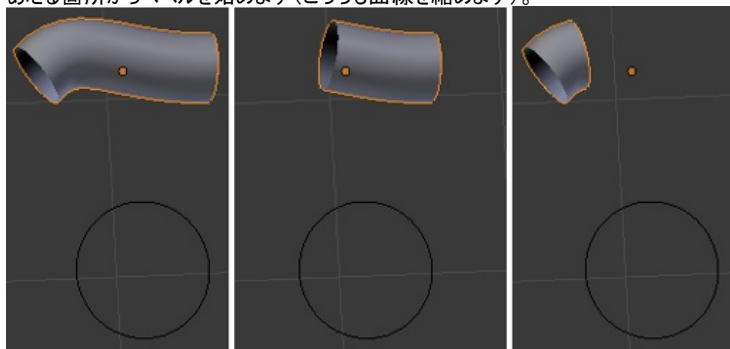
Taper Object を使って Start/End Bevel Factor を変更した曲線用のオプションです。Map Taper オプションは曲線全体ではなくベベル加工された部分にテーパ加工を適用します。



Map Taper を適用していない曲線 (左) と適用した曲線 (右)

Start Bevel Factor (開始ベベル係数) および End Bevel Factor (終端ベベル係数)

曲線のどの部分からベベルを始めるのかを決めるオプションです。Start Bevel Factor を 0.5 に上げると、曲線の開始位置から 50% の距離にあたる箇所からベベル処理を始めます (事実上曲線を縮めます)。End Bevel Factor を 0.25 まで下げると、曲線の終点から 25% の距離にあたる箇所からベベルを始めます (こちらも曲線を縮めます)。



A Curve with no Bevel factor を使わない曲線 (左)、50% Start Bevel Factor を指定した曲線 (中央)、25% End Bevel Factor を指定した曲線 (右)

[高度な曲線編集についてもっと詳しく](#) »

Path Animation

(パスアニメーション)

Path Animation 設定は、どうやって特定の経路に沿ってオブジェクトを動かすのかを決めます。詳しくは次のリンクをご覧ください。

[アニメーションでパスに曲線を利用する方法についてもっと詳しく](#) »

Active Spline

(アクティブなスプライン)



曲線の Active Spline パネル

Active Spline パネルは 編集モード にいるときに使えます。

Cyclic(循環ループ)

曲線を閉じます

Resolution(解像度)

分割数を変えて各セグメントの滑らかさを変えます

Interpolation(補間)

Tilt(傾き)

セグメントの傾きの計算方法を変えます

Radius(半径)

ベベル加工された曲線の半径の計算方法を変えます。制御点を Shrinking/Fattening (収縮/膨張) AltS させると効果がわかりやすくなります

Smooth(スムーズ)

曲線の法線を滑らかにします

Non-Uniform Rational B-Splines (NURBS)

(NURBS)

ベジエオブジェクトと NURBS オブジェクトの大きな違いの一つは、ベジエ曲線が近似値であることです。たとえばベジエ円は円の *近似値* ですが、NURBS 円は *正確な* 円です。NURBS 論は非常に複雑な話になり得ます。導入として [Wikipedia ページ\(現時点で日本語ページはありません\)](#) を調べてみてください。実際には、上述のベジエ曲線の操作の多くが NURBS 曲線に同じ方法で用いられます。以下 NURBS 曲線特有の操作に絞って説明します。

Editing NURBS Curves

(NURBS 曲線の編集)

NURBS 曲線は制御点の位置を動かして編集します。

1. ⇧ ShiftA で Add メニューを呼び出し、Curve (カーブ) » NURBS curve (NURBSカーブ) で曲線を設置します。
2. ⇐ Tab を押して 編集モード に入ります。
3. 制御点の一つを選んで動かします。制御点の新たな位置を確定するには LMB を、キャンセルするには RMB を使ってください。
4. 制御点を増やしたい場合には、両方の制御点を選び、W を押して Subdivide (細分化) を選びます。すぐに F6 を押して、分割数を指定します。

Active Spline

(アクティブなスプライン)

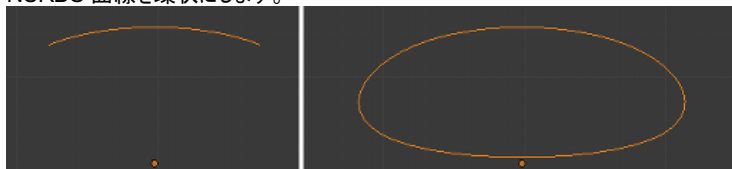


NURBS Active Spline パネル

NURBS オブジェクト特有なものの一つが *knot vector* です。これは一連の数値からなり、曲線にある制御点の影響度を決めるのに使われます。knot vector を直接編集することはできませんが、Active Spline パネルにある Endpoint(終点) および Bezier(ベジエ) オプションを介して変化させることができます。なお Endpoint と Bezier 設定は開いた NURBS 曲線にのみ使われます。

Cyclic(循環)

NURBS 曲線を環状にします。



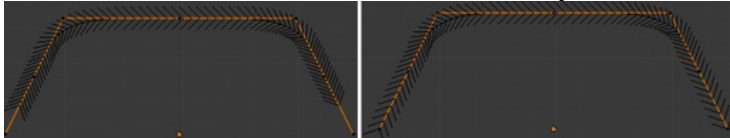
Cyclic が適用された NURBS 曲線

Bezier(ベジエ)

NURBS 曲線をベジエ曲線のように振舞わせませす

Endpoint(終点)

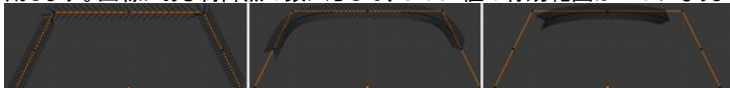
曲線を終端の制御点に接触させます。動作させるためには、Cyclic オプションを無効にしておく必要があります。



Endpoint が有効な NURBS 曲線

Order(次数)

NURBS 曲線の次数は、曲線にある制御点の影響範囲を決めます。次数が大きいほどひとつの制御点が曲線の相対的な大きさに大きく作用します。曲線にある制御点の数に応じて、Order 値の有効範囲は 2-6 になります。



次数が 2(左)、4(中央)、6(右)である NURBS 曲線

Path




(パス/経路)

上記のとおり曲線はよく [パス](#) として使われます。Path Animation オプションが選ばれていると、こういった曲線でもパスとして使うことができます。

Object Data (オブジェクトデータ) コンテキストで利用できる Path のオプションは、Active Spline パネルを開けないことを除けば、3D NURBS 曲線とまったく同じものです。

曲線の選択

編集モードで曲線を選択するのはメッシュよりもはるかに簡単です！この主たる理由は、選択可能な要素が制御点の1種類しかないからです（選択モードがありません）。ただ、制御点は単純な頂点よりも少し複雑で、特に Bézier には中心の頂点と2つのハンドルがあるため複雑です。

基本ツールは [メッシュ \(en\)](#) と同じなので、LMB -クリックで1つの制御点を選ぶことができ、選択中の制御点への追加は  Shift LMB -クリックや B 選択で行える、といった具合です。

Bézier 制御点を一言で言えば、中心の頂点を選ぶと、2つのハンドルも自動的に選ばれ、曲線に角度を作らずに、全体をつかむことができます。しかしながら、ハンドルを選んだときはこの頂点だけが選ばれ、制御ベクトルを編集できます。

L (または CtrlL) はマウスマウスカーソルが一番近い制御点と、それに繋がるものすべて（言い換えれば、同じ曲線に所属するすべての点）を選択に加えます。Bézier に関する注意点は、選択したハンドルに L を使うと、制御点とそれに繋がったものをすべて選ぶことです。

選択 (Select) メニュー

曲線に関するすべての「上級の」選択オプションは 3D ビューのヘッダーの 選択 (Select) メニューにまとめられています。詳細を見ていきましょう。

ランダム選択 (Random...)

反転 (Inverse)

全てを選択/解除 (Select/Deselect All)

矩形選択 (Border Select)

これらのオプションはすべて [オブジェクトモード \(en\)](#) と同じ意味を持ち、同じ動作をします（編集モードでの矩形選択 (Border Select) の特徴については、[こちら \(en\)](#) で議論されています）。

N番目ごとに選択/チェッカー選択 (Every Nth)

モード: 編集モード

ホットキー: なし

メニュー: 選択 (Select) » チェッカー選択 (Every Nth)

少なくとも1つ制御点を選択しているときにだけ動作します。現在選択している制御点の、N番目の制御点だけを残し、他を選択から外します。「選択間隔」は、ツールを選んだ時の N ポップアップの数値入力欄に表示されます。

最初/最後を選択/解除 (Select/Deselect First/Last)

モード: 編集モード

ホットキー: なし

メニュー: Select (選択) » 最初を選択/解除 (Select/Deselect First), Select » 最後を選択/解除 (Select/Deselect Last)

オブジェクト内の曲線の最初または最後の制御点の選択/非選択を切り替えます。曲線の始点を素早く見つけるのに便利です（パスとして使う場合など...）。

次/前を選択 (Select Next/Previous)

モード: 編集モード

ホットキー: なし

メニュー: 選択 (Select) » 次を選択 (Select Next), 選択 (Select) » 前を選択 (Select Previous)

現在の選択を基準に、次または前の制御点を選びます。

より多く/より少なく選択 (More and Less)

モード: 編集モード

ホットキー: Ctrl+ NumPad/Ctrl- NumPad

メニュー: 選択 (Select) » より多く/より少なく (More/Less)

この2つは逆の動作をします。[メッシュに対する同名操作 \(en\)](#) に似ています。現在選択中の制御点に対して、制御点を減らしたり増やしたりするためのものです。

アルゴリズムはメッシュ用のものと同じですが、結果はもっとわかりやすいものになります:

- より多く (More): 選択した制御点それぞれに、繋がっているすべての点 (2つ以上の場合もあります) を選びます。
- より少なく (Less): 選択した制御点それぞれについて、その点に繋がっている「すべての」点を選択されていれば、選択したままにします。その他の制御点は選択を解除します。

これは次の2つを意味しています:

- 1つ目に、曲線の「すべての」制御点を選択されていれば何も起きません (より少なく (Less) では、繋がっている点はすべて、常に選択されたままです。また、当然ながらより多く (More) ではもう追加できません)。逆に、どの制御点も選択されていない場合も同じことが言えます。

- 2つ目に、このツールが曲線の「外側」を選択することはありません(同じオブジェクト内の別の曲線に「ジャンプ」しません)。

Curve Editing

(曲線の編集)

このページでは曲線編集の基礎を扱います。曲線の基礎、選択、そして高度な編集については次のページで扱います:

- [曲線の基礎](#)
- [曲線の選択](#)
- [曲線の変形と押し出し](#)

Curve Display

(曲線の表示)

Display Options

(表示オプション)

編集モードのとき、プロパティシェルフ(N)のCurve Display パネルに、3D ビューポートでの曲線表示のオプションがあります。

ハンドル(Handles)

編集モードでのベジエのハンドル表示を切り替えます。これは曲線そのものの外見には影響しません。

法線(Normals)

曲線の法線の表示非表示を切り替えます。

法線サイズ(Normal Size)

曲線法線の表示スケールを設定します。

Hiding Elements

(要素を隠す)

編集モードでは、選択要素を隠したり再表示したりできます。画面上に多くの要素のある複雑なモデルで作業するときに役立ちます。

選択要素を隠す

Hを使うか、3D ウィンドウヘッダーの Curve(曲線) » Show/Hide(表示/隠す) » Hide Selected(選択しているものを隠す) メニューオプションを使います。

隠れた要素を表示

AltHを使うか、3D ウィンドウヘッダーの Curve(曲線) » Show/Hide(表示/隠す) » Show Hidden(隠したものを表示) メニューオプションを使います。

選択していない要素を隠す

⇧ ShiftHを使うか、3D ウィンドウヘッダーの Curve(曲線) » Show/Hide(表示/隠す) » Hide Unselected(選択していないものを隠す) メニューオプションを使います。

Basic Curve Editing (translation, rotation, scale)

(基本的な曲線編集(移動、回転、拡大縮小))

モード: 編集モード

ホットキー: G/R/S

メニュー: Curve(曲線) » Transform(トランスフォーム) » Move, Rotate, Scale(移動, 回転, 拡大縮小), ...

Blender の他の要素と同様に、曲線の制御点は移動(G)、回転(R)、拡大縮小(S)させることができます([3D空間の操作](#))。編集モードでは [プロポーショナル編集](#) もトランスフォーム操作に使えます。

Snapping

(スナップ)

モード: 編集モード

パネル: 曲線ツール

曲線の構成要素には [メッシュのスナップ](#) も動作します。制御点とハンドルはどちらも、それ自身の内部(アクティブな曲線の他要素)を除いて、スナップの影響を受けます。スナップは 2D 曲線に動作しますが、点の位置がローカルの XY 座標に制約されます。

Deforming Tools

(変形ツール)

モード: 編集モード

メニュー: Curve(曲線) » Transform(トランスフォーム)

変形ツール 球へ変形(To Sphere)、せん断(Shear)、湾曲(Warp)と押す/引く(Push/Pull)については [3D の操作](#)をご覧ください。

他の2つ、Tilt(傾き)と収縮/膨張(Shrink/Fatten Radius)は [曲線の押し出し](#) に関係しています。

Smoothing

(スムーズ)

モード: 編集モード

ホットキー: W » Smooth(スムーズ)

曲線のスムーズはスペシャルメニューから利用できます。現在のところ、ベジエ曲線では接線ではなく、制御点の位置を滑らかにするだけです。スムーズ時、両端の点は移動しません。

Mirror

(ミラー)

モード: 編集モード

ホットキー: CtrlM

メニュー: Curve(曲線) » Mirror(ミラー)

[メッシュの頂点](#) にミラーを使うのと同じ動作をします。

Set Bézier Handle Type

(ベジエ曲線のハンドル種類)

モード: 編集モード

パネル: Curve Tools(曲線ツール) » Handles(ハンドル)

ホットキー: V

メニュー: Curve(曲線) » Control Points(制御点) » Set Handle Type(ハンドルタイプの設定)

ハンドル種類は [ベジエ曲線](#) のプロパティで、曲線の性質を変えるのに使われます。たとえば Vector ハンドル に切り替えれば鋭いコーナーを持つ曲線を作れます。詳細は上のリンクを参照してください。


Extending Curves



(曲線の拡張)

モード: 編集モード

ホットキー: Ctrl LMB  または E

メニュー: Curve(曲線) » Extrude(押し出し)

曲線を一旦作れば、そこから押し出すか、Ctrl LMB  クリックで新たなハンドルを置いて新しいセグメントを追加できます(実際には、新たな制御点を追加して新たなセグメントを定義します)。新しいセグメントは曲線のいずれかの終端に追加されます。曲線のいずれかの終端にある、単一の頂点かハンドルが選択されているときにだけ、新しい線分が追加されます。もし2つ以上制御点を選択していると、新しい閉じたベジエ曲線ができます。

メッシュと違って、何も選択せずにCtrl LMB  クリックするだけでは、編集集中のオブジェクトの中に新たな曲線を作ることはできません。そのためには、[\(セグメントの削除\)](#)で既存の曲線を2つの部分にわけ、既存のものを[複製するか](#) ( ShiftD) 新しく追加します (Add(追加)メニュー)...

Subdivision

(細分化)

モード: 編集モード

パネル: Curve Tools(曲線ツール)

ホットキー: W » 1

メニュー: Curve(曲線) » Segments(セグメント) » Subdivide(細分化)

細分化は、単に選択されたセグメントの間に1つ以上の制御点を追加することで、行われます。分割数の変更は W を押して一度細分化したあとで、F6 で分割数メニューを呼び出します。

Duplication

(複製)

モード: 編集モード

ホットキー: ⇧ ShiftD

メニュー: Curve(曲線) » Duplicate(複製)

選択された制御点を同位置に複製します。複製が選択されて移動モードに入るので、移動先を決めることができます。

Joining Curve Segments

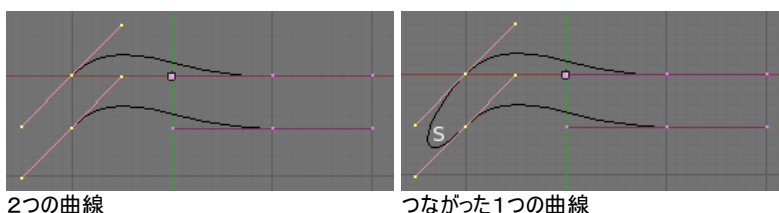
(セグメントの結合)

モード: 編集モード

ホットキー: F

メニュー: Curve(曲線) » Make Segment(セグメントを作る)

2つの開いた曲線の間にセグメントを作って、1つに結合できます。別々の曲線をつなげるには、両方の曲線の端にある制御点をひとつ選んで F を押してください。2つの曲線はセグメントで繋がられて1つの曲線になります。



さらに、端点をつなげると曲線を閉じることができます。同じタイプの曲線だけを繋がられる点に注意してください(ベジエ曲線とベジエ曲線、NURBS曲線とNURBS曲線)

Separating Curves

(曲線の分離)

モード: 編集モード

ホットキー: P

メニュー: Curve(曲線) » Separate(別オブジェクトに分離)

複数の独立した曲線でできた曲線オブジェクトは、目的のセグメントを選んで P を押すと別のオブジェクトに分離できます。なお、曲線オブジェクトに曲線が一つしかない場合は、P を押すと制御点のない曲線オブジェクトが作られます。

Deleting Elements

(要素の削除)

モード: 編集モード

ホットキー: X or Del

メニュー: Curve(曲線) » Delete(削除)...

削除 ポップアップメニューが現れます。次の3つの選択肢があります:

選択 (Selected)

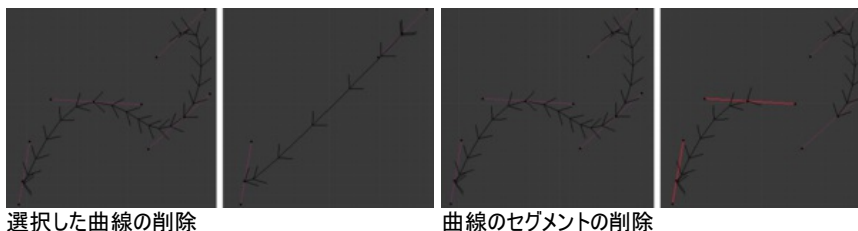
選択している制御点を「曲線を壊さずに」削除します(消えた点を挟んでいた二点が直接つながります)。NURBS の次数が制御点の数より多くできないことを忘れないでください。制御点を消すと、次数を減らす可能性があります。もちろん、点がひとつだけ残されると曲線はもう目に見えなくなり、すべての点が削除されれば、曲線そのものが削除されます。

セグメント (Segment)

このオプションは前のオプションと対照的です。選択されたセグメントを削除して、制御点を一切削除せずに曲線を切断します。このオプションは、複数のセグメントが選択されていても、常に *セグメントを1つだけ*(最後に選択されたもの) 削除します。選択しているセグメントをすべて消したい場合は、同じオプションを繰り返す必要があるでしょう。

すべて (All)

メッシュを対象とするのと同じで、オブジェクト内のすべてを削除します



選択した曲線の削除

曲線のセグメントの削除

Opening and Closing a Curve

(開いた曲線と閉じた曲線)

モード: 編集モード

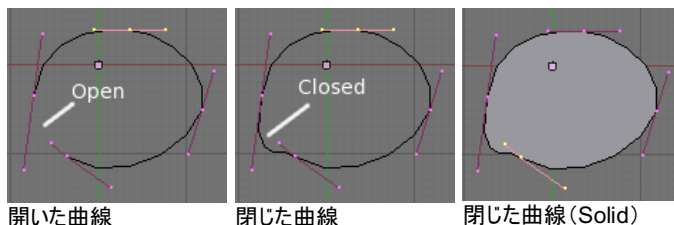
ホットキー: AltC

メニュー: Curve(曲線) » Toggle Cyclic(ループ切り替え)

開いた曲線と閉じた(環状の)曲線を切り替えます。少なくとも1つの制御点が選択されていないと、曲線を開いたり閉じたりすることはできません。

閉じたセグメントの形状は、ベジエ曲線では開始ハンドルと終了ハンドルに基づくものになり、NURBS ではいつもどおり隣接する制御点に基づくものになります。ハンドルがAuto(自動)タイプの場合だけは、閉じたあとでハンドルが調整されます。下の画像(開いた曲線)と(閉じた曲線)は、同じベジエ曲線を開いたものと閉じたものです。

この操作は、もともとあった開始制御点か、追加した最後の制御点にだけ動作します。セグメントを削除しても動作の仕方は変わらず、開始点と最後の制御点だけを操作します。これは、AltC は1つの曲線を閉じるのではなく、実際には2つの曲線をつなげるかもしれないということです！ 2D 曲線が開いていると、レンダリング可能な平面を作ることを忘れないでください。



開いた曲線

閉じた曲線

閉じた曲線 (Solid)

Switch Direction

(向きの反転)

モード: 編集モード

ホットキー: W » 2 NumPad

メニュー: Curve(曲線) » Segments(セグメント) » Switch Direction(向きの反転), Specials(スペシャル) » Switch Direction(向きの反転)

少なくともひとつ要素が選択されているすべての曲線の向きを「反転」します(開始点は終点になり、逆も同様です)。主に、曲線をパスとして使うときやベベルとテーパオプションを使う場合に役立ちます

Converting Tools

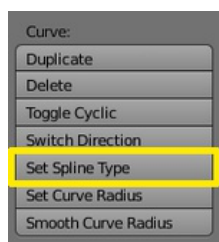
(変換ツール)

Converting Curve Type

(曲線種類の変換)

モード: 編集モード

パネル: Curve Tools(曲線ツール)



スプラインのタイプ設定ボタン

曲線オブジェクト内のスプラインを、ベジエ、NURBS、Poly曲線間で変換できます。Tを押してツールシェルフを開きます。Set Spline Type(スプラインのタイプを設定) ボタンをクリックすると、選択したタイプを選べます。

なお、これは「賢い」変換ではありません。Blenderは形状を維持しようとせず、制御点の数を維持しようとしません…。例えば NURBS をベジエに変換すると、3つの NURBS 制御点のグループはそれぞれ、ベジエ曲線固有の制御点(中心点と2つのハンドル)になります。

Convert Curve to Mesh

(曲線のメッシュへの変換)

モード: オブジェクトモード

メニュー: Object(オブジェクト) » Convert To(変換)

オブジェクトモードでのみ動作する、曲線からメッシュへの「外部」変換もあります。曲線の解像度を使って辺や頂点を作り、曲線オブジェクトをメッシュオブジェクトに変換します。なお、閉じている曲線や押し出された曲線の作る面やボリュームは維持されます…

Convert Mesh to Curve

(メッシュの曲線への変換)

モード: オブジェクトモード

メニュー: Object(オブジェクト) » Convert To(変換)

一連の繋がった頂点でできたメッシュオブジェクトは曲線オブジェクトに変換できます。多角形タイプの曲線が作られますが、上述のように滑らかなセグメントをもつように変換することができます。

Curve Parenting

(曲線の頂点ペアレンティング)

モード: 編集モード

ホットキー: CtrlP

メッシュオブジェクトと同じく(CtrlP で、選択オブジェクトを1つか3つの制御点の [子](#) にできます。

制御点を1つまたは3つ選び、次に他のオブジェクトをCtrl RMB を選んで、CtrlPを使って頂点ペアレントを作ります。

Hooks

(フック)

モード: 編集モード

ホットキー: H

メニュー: Curve(曲線) » Control Points(制御点) » Hooks(フック)

1つ以上の点を他のオブジェクトで操作するために [フック](#)を使えます。

Set Goal Weight

(ゴールウェイトの設定)

モード: 編集モード

メニュー: W » ゴールウェイトの設定

Set Goal Weight(ゴールウェイトの設定)

選択した制御点の「ゴールウェイト」を設定します。曲線が軟体物理を持つときに使われ、このウェイトに応じて曲線をもとの位置に「粘着」させます。

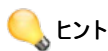
Curve Deform

(曲線の変形)

曲線の変形を使うと、簡単に効率的なメッシュ変形が行えます。メッシュオブジェクトと曲線に親子関係を作ると、メッシュを曲線に沿わせたり、変形軸に直行させて動かしたりしてメッシュの変形度合いを変えることができます。プリンター内を流れる紙、カメラの中にあるフィルム、水路の水など、複雑なパスにオブジェクトに沿わせるときに最も役立つツールです。

曲線の変形は(グローバルの)変形軸(deformation axis)である X、Y、Z軸に対して動作します。これは、変形軸方向にメッシュを動かすと、メッシュが曲線上を動いていくということです。変形軸に直角な方向にメッシュを動かすと、メッシュオブジェクトは曲線から近づくか、遠ざかるかします。Blender のデフォルトでは Y軸が変形軸に設定されています。オブジェクトを曲線の終端を越えて動かすと、オブジェクトは曲線の終端のベクトルに沿って変形を続けます。

「曲線の経路」が 3D なら、その制御点の傾き(Tilt) の値がその周りの「曲げられた」オブジェクトをねじる(twist)のに使われます(上の[押し出し](#)の章をご覧ください)。半径(Radius)プロパティは、残念なことに使われていません(以前は例えば「曲げられた」オブジェクトのサイズ変更をするために使えたのかもしれませんが...)



ヒント

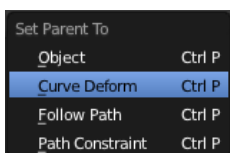
オブジェクトを追加したらすぐ、曲線変形を追加する前に、オブジェクトを曲線の上に置いてください。変形のしかたを制御しやすくするには、こうするのが一番良いでしょう。

モディファイアを使いましょう！

曲線の変形の動作は [カーブ](#) モディファイアでも実現できます。基本的にカーブモディファイアは同じ動作をしますが、違うのは、プロパティパネルで「変形軸(deformation axis)」を設定すること、Track X/Y/Z ボタンがもう効力を持たないことです。そして追加機能では、メッシュの頂点グループにあるものだけを曲げるようなことができます...

Interface

(インタフェース)

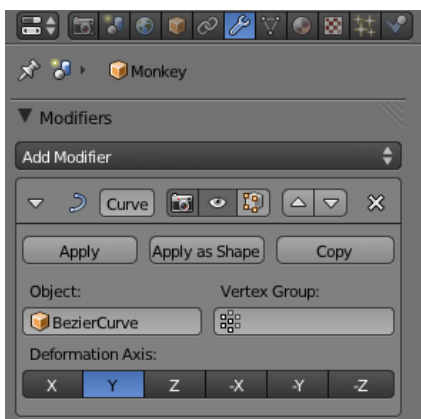


ペアレント対象(Make

Parent)メニュー

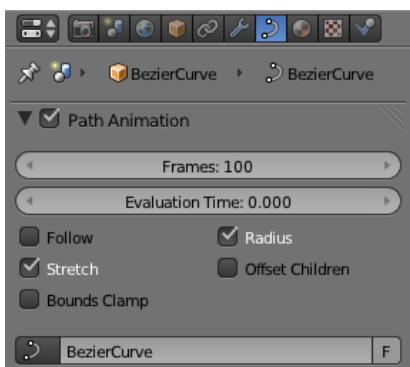
オブジェクト(メッシュ、曲線、メタ...)と曲線に親子関係を作る(CtrlP)と、ペアレント対象(Make Parent)メニューが表示されます(右図)。

曲線の変形を選ぶと、メッシュオブジェクトの曲線変形機能を有効化できます。



アニメーション設定(Anim settings)パネル

変形軸(deformation axis)の設定はメッシュオブジェクトに対して行います。Blender のデフォルトの変形軸は Y です。Anim パネル(アニメーション設定パネル)にある、Track X、Y、Z ボタンを押して変えられます。



曲線/サーフェスパネル

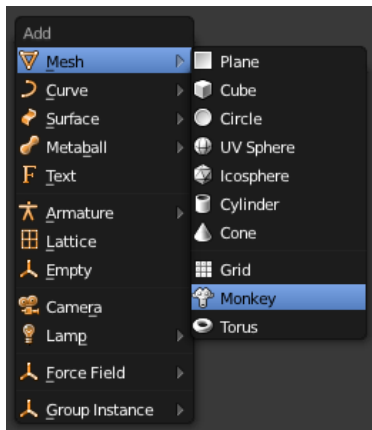
環状の(閉じた)曲線では、環状の経路に沿ったオブジェクト変形が期待通りに動作します。ただし「親」オブジェクトに複数の曲線があるときは、その「子」は最初の1つにのみ沿います。

ストレッチ(Stretch)オプションを使うと、メッシュオブジェクトの伸縮、押しつぶしを曲線全体にあわせて行うようにできます。この設定は「親」の曲線に行います。画像 [曲線/サーフェスパネル](#) をご覧ください。

Example

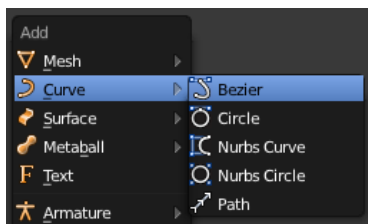
(例)

簡単な例を作ってみましょう:



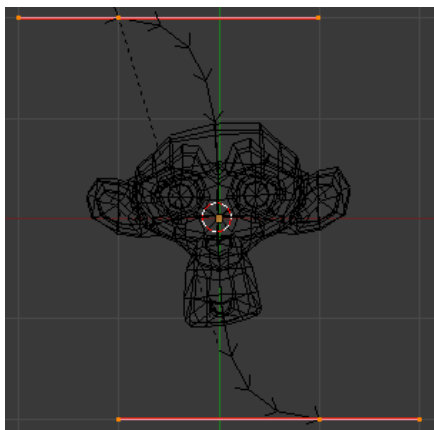
Monkeyを追加!

- デフォルトのキューブオブジェクトをシーンから削除して、Monkey(⇧ ShiftA » Add(追加) » Mesh(メッシュ) » Monkey(モンキー)、右図 *Monkey を追加!*)を追加します。
- ⇧ Tabを押して編集モードを抜けます。



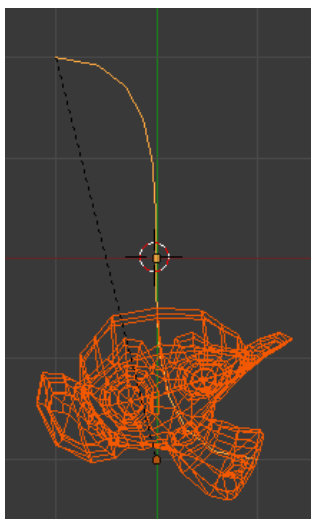
曲線を追加

- 曲線を追加します(⇧ ShiftA » Add(追加) » Curve(曲線) » Bezier(ベジエ)、右図 *曲線を追加*)。



曲線を編集

- 編集モードにいる間、曲線の制御点を右図 *曲線を編集* のように動かし、編集モードを抜けます(⇧ Tab)。



曲線上の Monkey

- さて、新しいモディファイアを使って Monkey を「曲げて」見ましょう:
 - Monkey を選択します (RMB)。
 - モディファイア (Modifiers) パネルからカーブ (Curve) モディファイアを追加します。
 - 曲線の名前をモディファイアのオブジェクト (Object) 欄に入力し (「Curve」にしましょう) 変形の軸 (Deformation Axis) を Y にします。
- 右図 *曲線上の Monkey* のように、Monkey は曲線上に配置されなければなりません。
- Monkey を選び (RMB) Y 方向に (デフォルトの変形軸) に動かすと (G)、monkey は曲線に沿ってうまく変形します。

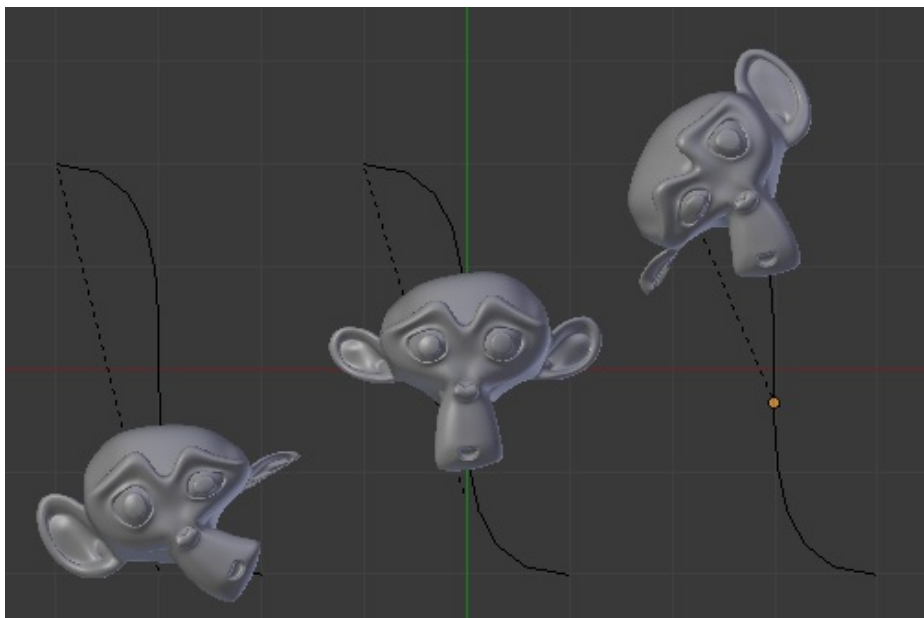
ヒント

Monkey を動かすときに MMB (または X/Y/Z のいずれか) を押すと、1 つの軸方向にだけ移動を制限できます。

- 画像 *Monkey の変形* は、Monkey が曲線上数カ所に置かれています。変形を見やすくするために、細分割面モディファイア (SubSurf) を細分化 (Subdiv) を 2 にして使い、スムーズ (Smooth) を Monkey のメッシュに設定しました。

ヒント

変形の軸以外の方向に Monkey を動かすと変わった変形になります。作りたいものができることもあるので、試す価値はあります。試してみてください！



Monkey の変形

Curve Extrusion

(曲線の押し出し)

このセクションでは曲線の押し出し方、厚みをつける方法、パスに沿って厚みを調整する方法を扱います。

Extrusion

(押し出し)

モード: オブジェクトまたは編集モード

パネル: Curve (曲線) または Surface (曲面)

ああ！ 押し出しは！ おそらく曲線のモデリングで一番面白いツールです。特にベベル (bevel) / テーパー (taper) / 傾き (Tilt) / 半径 (Radius) オプションと組み合わせると…。なお、[以前のページ](#)で述べられた 押し出し (E) コマンドとは無関係です！

動作の影響範囲に応じてさまざまな設定があります：

Width (幅)

押し出した曲線の、曲線自身に相対的な「境界線」の位置を調整します。閉じた 2D 曲線 (後述) では、とてもわかりやすいものです。幅を **1.0** より広くするほど押し出したボリュームの幅が広くなり、幅を **1.0** にすると境界線は曲線にぴったり合う大きさになり、Width を **1.0** より狭くするほどボリュームは狭くなります…。この原理は開いた 2D と 3D の曲線にもあてはまりますが、曲線の「外側」と「内側」の判断の仕方が少しおかしいようです…。押し出した「ベベル」オブジェクトと同じ効果です…。

Tilt (傾き)

残念ながら、この値は Blender のどこを探しても見つかりません。この設定は曲線周りの各点の「ねじれ角度 (twisting angle)」を調整します。したがって 3D の曲線にだけ関係します！

値は 傾き (Tilt) トランスフォームツール (T か Curve (曲線) » Control Points (制御点) » Tilt (傾き)) を使って設定でき、AltT (または Curve (曲線) » Control Points (制御点) » Clear Tilt (傾きをクリア)) でデフォルト値 (元の曲線の平面に垂直) に戻すことができます。NURBS では常に、傾きが滑らかに補間されます。一方ベジエ曲線では、補間アルゴリズムを Curve Tools (曲線ツール) パネルにある、傾きの補間 (Tilt Interpolation) ドロップダウンリストから選ぶことができます (古典的な 線形 (リニア、Linear)、カーディナル (Cardinal)、Bスプライン (B Spline) および ゆるやか (イーズ、Ease) オプションがあります…)

Simple Extrusion

(単純な押し出し)

最初にベベル/テーパーを使わない、「単純な」曲線の押し出しを見てみましょう。

Extrude (押し出し)

押し出しの幅 (または高さ) を調整します。オブジェクトのスケールが 1 のとき、押し出しを **1.0** にすると、1 BU の大きさを持つ曲線を、曲線の平面に垂直な軸に沿って、両側に押し出します (3D 曲線の仕様については後述します)。オブジェクトのスケールが変わると押し出しの長さも当然変わってきます。

0.0 にセットすると押し出しはありません！

Bevel Depth (ベベルの深度)

押し出しにベベルを追加します。効果については下をご覧ください…

ベベルは押し出しをより広く、高くすることに注意してください。

0.0 にセットされると、ベベルはなくなります (最大値は **2.0**)。

Bev Resol (ベベルの解像度)

ベベルの深度を 0 より大きくしてできた、ベベルの解像度を調整します。

0 (デフォルト) に設定されるとベベル表面は単に「平ら」になります。

曲線自身の解像度と同じく、値が高いほどベベルは滑らかで、角の丸みが増します…

曲線が開いているか閉じているか、または 3D であるかどうかによって 3 種類に分類できます。

開いた 2D の曲線

押し出しは曲線形状に沿う「壁」や「リボン」を作ります。ベベル深度が使われると、壁が滑り台か溝のようなものになります。ベベルの方向は奇妙で予期しないものになることがあり、同じ曲線の逆方向になることがよくあります…。方向は曲線の [Switch Direction \(方向の切り替え\)](#) で変えられます。

これを使うと例えば押し出してベベルをつけた曲線と、その曲線にパスに沿う制約をつけた球を組み合わせ、複雑な滑り台を大理石が転がり落ちるところを手早くシミュレートするようなことができます…

閉じた 2D の曲線

ボリュームを手早く作れるため、おそらくいちばん役立つ状況です。押し出された「壁」の両側を、デフォルトでは 2 つの平らで平行な面で埋めます。3D の隣にある 後ろ (Back) や 前 (Front) の切り替えボタンを無効にすると、片方か両方の面を取り除けます。

ここで追加されるベベルは常に「右方向」になり、ボリュームの「辺」を滑らかにできます。

3D 曲線

ここでは曲線が開いているかどうかは重要ではありません。3D 曲線の押し出しでボリュームを作ることはできず、開いた 2D 曲線を使ったときのようない壁かリボンしかできません。

ただ、3D 曲線にはもっと多くの機能があります：制御点の傾き (Tilt) です (上記参照)。曲線の周りにねじるリボンを作ります…例えばメビウスの帯のような！

Advanced Extrusion

(高度な押し出し)

ここでは、非常に入り組んだ有機形状を作るために、1 つか 2 つ追加の曲線オブジェクトを使って押し出します。

この種の押し出しを有効にするには、曲線のベベルオブジェクト (BevOb) 欄に、押し出しの「脊柱」として使う、有効な曲線オブジェクトの名前を入力する必要があります。「ベベル」曲線は押し出したオブジェクトの交差部分を制御します。ベベルオブジェクト (BevOb) 曲線が 2D であるか 3D であるかに関わらず、それが閉じていれば「チューブ状」の押し出しを作り、そうでなければ溝か滑り台のようなオブジェクトを作ります…

オブジェクトは内部の曲線の長さ全体に沿って押し出されます。デフォルトの押し出し幅は固定値ですが、制御点の半径 (Radius) プロパティが「テーパー (taper)」オブジェクトのどちらかのやり方で調整できます。

点の半径 (Radius) は収縮/膨張 (Shrink/Fatten Radius) トランスフォームツール (AltS か 曲線 » トランスフォーム » 収縮/膨張 (Shrink/Fatten Radius)) を使うか、スペシャル (Specials) メニュー (W) にあるカーブ半径を設定 (Set Radius) 入力欄から設定できます。ここでも、残念ながら制御点の半径 (Radius) はどこにも見つかりません...

半径 (Radius) を使って、「脊髄」に沿う押し出しの幅を直接制御できます。傾き (Tilt) (上記参照) と同じく、ベジエ曲線に使われる補間アルゴリズムを曲線ツール パネルにある 半径の補間 (Radius Interpolation) ドロップダウンリストで選べます。

ただ、別のもっと正確なオプションを使うこともできます。「テーパー (taper)」オブジェクトです。「ベベル」と同じで、メインの曲線のテーパーオブジェクト (TaperOb) 欄で名前を決められます。これは *開いた曲線* でなければなりません。テーパー曲線はローカルの X 軸に沿って作られ、ローカルの Y 軸で幅を制御します。次の点にも注意してください:

- テーパーは押し出されたすべての曲線に、それぞれ独立して用いられます。
- テーパーオブジェクトの曲線が複数のわかれたセグメントを持っていたとしても、最初の曲線だけが使われます。
- 拡大縮小は左の最初の制御点から始まり、曲線に沿って動き、右の最後の制御点まで動きます。
- 負の拡大縮小 (テーパー曲線のローカルの 負の Y 座標) も同様に使えます。ただし、レンダリングの乱れ (rendering artifact) が起きるかもしれません。
- テーパー曲線の計算に基いた通常の押し出し幅を拡大縮小します。これは、テーパー曲線の角の立ったコーナーが見にくくなるということです。土台になる曲線の (解像度 (DefResoLU)) を上げる必要があるでしょう。
- 閉じた曲線では、テーパーオブジェクト (TaperOb) のテーパー曲線変形はオブジェクトの長さだけでなく曲線全体 (オブジェクトの周囲) に行われ、押し出しの深さを変えます。このとき、循環点 (cyclic point、曲線の終端同士が繋がる場所) の変わり目を滑らかにするには、テーパーオブジェクト (TaperOb) のテーパー曲線の両端の高さを同じにします。

大事なことを言い忘れました、3D の「脊柱」曲線では、制御点の 傾き (Tilt) で、曲線に沿って押し出した「ベベル」のねじれを制御できます！

Examples

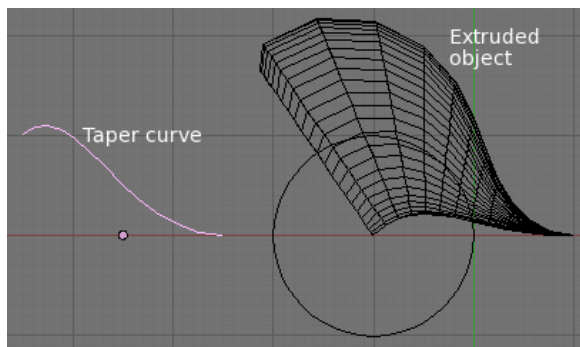
(例)

TODO: 簡単な押し出しの例を追加

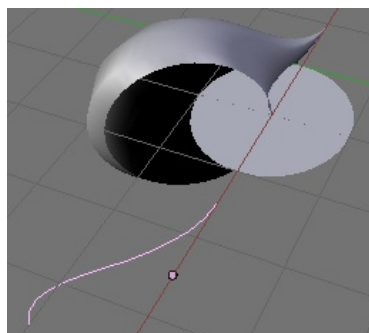
TODO: 半径を指定したベベル押し出しの例を追加

テーパー曲線を使って簡単な曲線の円をテーパー押し出ししてみましょう。曲線を追加して編集モードを抜けます。曲線をもう一つ追加して (円のような閉じた曲線) 名前を「BevelCurve」にし、最初の曲線のベベルオブジェクト (BevOb) 欄に入力します (プロパティウィンドウの Object Data コンテキストから)。パイプができました。3つめの曲線をオブジェクトモードで追加し、名前を「TaperCurve」にします。左の制御点を5単位ほど上に移動します。

では最初の曲線の Object Data パネルにある テーパーオブジェクト (TaperOb) 欄に、先ほど「TaperCurve」と名付けた新しいテーパー曲線を入力します。Enter を押すとテーパー曲線が用いられ、すぐに結果が表示されます。(画像 *テーパー押し出しを使った曲線*)。

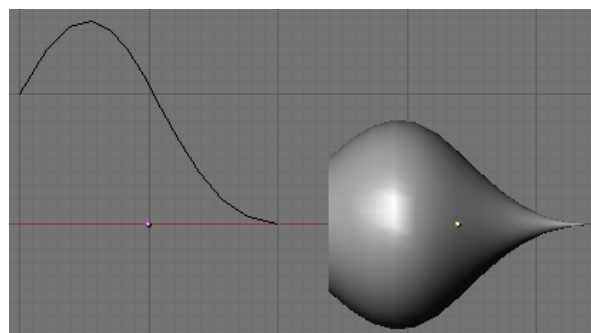


テーパー押し出しを使った曲線



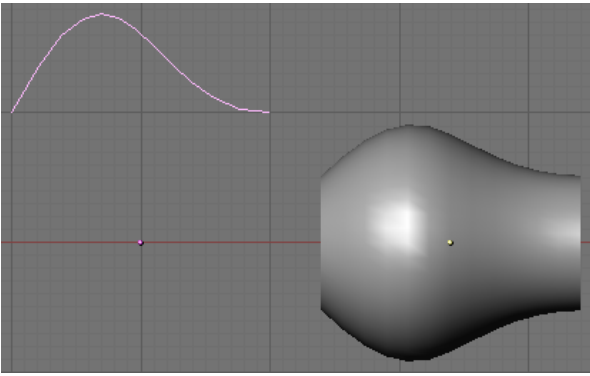
テーパーソリッドモード

ご覧のように「テーパー曲線」が押し出したオブジェクトに用いられています。左から右に向かうにつれ、パイプのボリュームがどのように縮み、最終的になくなるかを見てください。テーパー曲線がローカルの Y 軸より下になると、パイプの内側が外側になり、描画の乱れが生じます。もちろん、アーティストとして望んだものかもしれませんが！



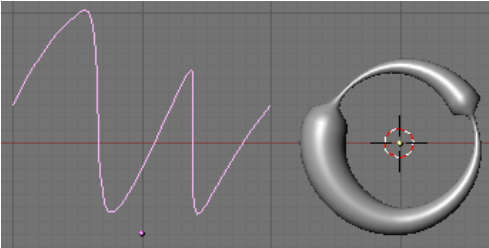
テーパーの例 1

右の画像 *テーパー例 1* では、ご覧のように左のテーパー曲線が右の曲線オブジェクトに与える影響がはっきりとわかります。左のテーパー曲線がオブジェクトの中心に近づくほど、右の曲線オブジェクトは小さくなります。



テーパーの例 2

画像 テーパーの例 2 では、左のテーパー曲線の制御点が中央から離されていて、右の曲線オブジェクトの幅が広がっています。



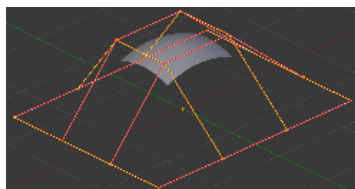
テーパーの例 3

画像 テーパーの例 3 は、テーパー曲線を曲線の円に用いるもっと変わった例を示します。

TODO: 傾き (Tilt) を使ったベベル押し出しの例を追加

Surfaces

(曲面/サーフェス)



曲面

Curve (曲線) は 2次元のオブジェクトで、Surface (曲面、訳語は表面/サーフェスのことも) はその 3次元拡張です。ただし Blender では NURBS の曲面はありますが、ベジエや(ベジエのノット(knot)タイプ)はあります。下記参照) ポリゴンはありません(ポリゴンにはメッシュを使えます!)。曲線と曲面(とテキストオブジェクトも...) は同じオブジェクト型を共有しているというものの、同じものではありません。例えば同じオブジェクトに曲線と曲面の両方を持つことはできません。

曲面は 2次元なので、ふたつの補間軸 U と V を持ちます(曲線では U だけを使いました)。重要なのは補間規則(ノット/次数/解像度)は U 方向と V 方向のそれぞれ別に調整できることです。

「でも曲面は 3次元で表示されるよ、何でこれが 2次元なの?」と思われるかもしれませんが、3次元であるためには、オブジェクトは体積(Volume)と表面を持つ必要があります。そうでなければ、たとえ図形が閉じていても体積はなく、無限に薄くなります。体積があれば曲面は 3つ目の次元である厚さを持っていたでしょう。このように、曲面は 2次元オブジェクトでしかなく、2つの補間次元、または軸、あるいは座標だけを持っています(数学を少しご存知であれば、非ユークリッド幾何学を思い出してみてください。曲面とは非ユークリッドの 2次元平面のことでした...)。もっと現実に近づく例で言えば、紙を丸めると円柱を作れますが、体積のある物体ができたように見えても、紙そのものは 2次元(に近い)オブジェクトのままなのです!

実際のところ、曲面は **曲線の押し出し** の実行結果に非常によく似ています(ところで私は、押し出した曲線を曲面に変換することが、少なくとも NURBS だけでできたものなら、できるように思います。しかし現時点では Blender ではできません...)。

Finding Surface Tools

(曲面ツールの場所)



曲面ツール

プロパティウィンドウに現れる曲面タブは **曲線** のものと同じですが、オプションが少し減っています。あとはいつものように 3Dビューのヘッダーにある Select(選択) と Surface(サーフェス)メニュー、そして Specials(スペシャル) (W) のポップアップメニューからツールを呼び出せます。

Visualization

(表示)

3Dビューでの表示は NURBS 曲線とほぼ同じです。ただし u 方向が黄色のグリッド線で示され、v 方向がピンク色のグリッド線で示されます(図「曲面」)。

曲線と同じように制御点を **隠す/再表示** ことができ、ツールシェルフの **Surface Tools(サーフェスツール)** パネルには曲線の Curve Tools(曲線ツール)と同じ描画オプションがあります。

Surface Structure

(曲面の構造)

曲線、特に **NURBS** の概念の多くが、NURBS 曲面に持ち越されています。例えば Control-Point(制御点)、Order(次数)、Weight(重み)、Resolution(解像度)などです。ここでは両者の違いだけを説明します。

NURBS曲線とNURBS曲面の違いを理解するのは重要です。前者は 1次元で、後者は 2次元です。Blender の内部的な扱いは、NURBS 曲面と NURBS 曲線で完全に異なります。ふたつを隔てる属性がいくつかありますが、もっとも重要なのは、補間軸が NURBS 曲線には一つ(U)、NURBS 曲面には二つある(UとV)ことです。

しかし「2次元の」曲面は曲線で作ることができます(押し出しツールを使うか、それほど多くはありませんが閉じた 2D 曲線に面を貼ります)。曲面で作られた「1次元の」曲線を作ることもできます。NURBS曲面の制御点の並びを UとVのいずれか一方方向にすると、曲線になります。

両者を見分けるには、編集モードに入って 3Dウィンドウのヘッダーを見ます。ヘッダーにメニュー項目“Surface”(サーフェス)と“Curve”(曲線)のどちらかがあります。NURBS 曲面の曲線全体を押し出して 曲面を作ることはできますが、NURBS 曲線ではできません(ここでは非常に専門的な曲線押し出しツールではなく、E ショートカットで実行できる標準的な押し出しツールの話をしています。わかりやすい話ではないと思いますが...)

Control Points, Rows and Grid

(制御点、行とグリッド) NURBS 曲面の制御点は NURBS 曲線のものと同じです。しかし配置は非常に限られています。“Segment”(セグメント/曲線分)の概念は消えて “Row”(行)と全体に渡る “Grid”(グリッド)に置き換えられています。

「行」とは、補間方向に「線」を作って並ぶ制御点の集まりです(メッシュにおける [Edge Loop](#)(ループ辺)に少し似たものです)。したがって NURBS 曲面には “U行” と “V行” があります。理解のカギとなるのは、種類(UかV)の同じ行はすべて、同じ数の制御点を持つということです。制御点はすべて、ひとつの U行とひとつの V行に属します。

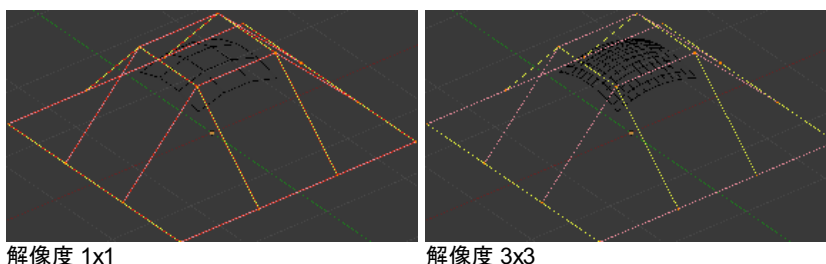
これら全体で形作られた “グリッド” か “檻” のようなものを操作すれば、NURBS 曲面の形状を操作することができます。 [Lattice](#) に少し似ています。

次の点を把握しておくのが大切です。NURBS 曲面に、制御点を単独で追加することはできず、制御点の数が他の行と正確に同じになるように、U行 または V行全体を一度に追加する必要があります(実際には、押し出しツールや複製ツールで追加することになるでしょう)。これは同じ数の制御点を持つ行が最低でも一つないと、曲面同士の「結合」はできないということです。

Surface Resolution

(曲面の解像度)

ちょうど [NURBS 曲線](#)のように、Resolution(解像度)は曲面の詳細度を調整するものです。解像度が高いほど曲面は滑らかに、低いほど粗くなります。曲線と違うのは解像度の設定が補間軸 UとVそれぞれに一つずつ(厳密にはプレビュー時レンダリング時でわかれるので全部で四つ)あることです。この設定はプロパティウィンドウの Object Data(オブジェクトデータ)タブにあります。



図「解像度 1x1」は曲面解像度の UとVをどちらも 3にした例です。図「解像度 3x3」は曲面解像度の UとVをどちらも 12にした例です。



解像度パネル

描画領域での処理を遅くせずに、レンダリング結果の質は保ちたい場合、プレビュー時とレンダリング時の解像度を別にすることができます。

Closed and Open Surfaces

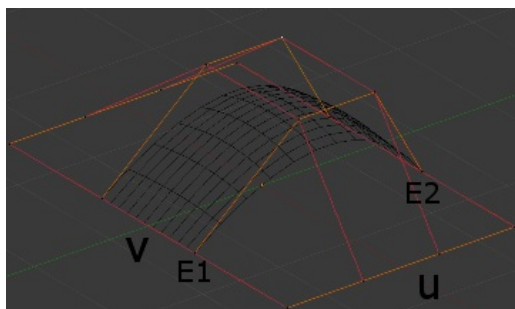
(閉じた曲面/開いた曲面)

曲面には曲線と同じように、閉じている(循環している)か開いているかの区別がありますが、曲面ではこの区別が補間方向ごとになります。チューブやドーナツ、球の形状を簡単に作る事ができ、編集モードでは「立体」として描画されます。これは曲面を非常に扱いやすくします。

Knots

(ノット/結び目)

[NURBS 曲線](#)にもある knot ベクトルは NURBS 曲面には二つあり、それぞれ U軸、V軸のためのものです。ここでも knot は Uniform(均一)、Endpoint(終点)、または Bezier(ベジエ)のいずれかになり、曲線のものと同じプロパティを持ちます。曲線と同じように knot の設定は(対応する補間軸方向に)開いた曲面だけに影響します。



終点 U

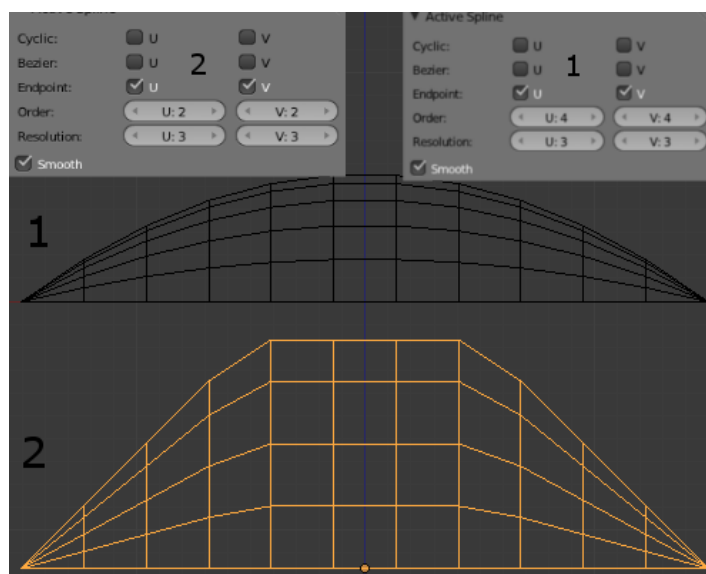
図「終点 U」では、U 補間軸が u 、V 補間軸が v で示されています。U 補間軸は Endpoint(終点)に設定され、曲面は U 補間軸上の E1 から E2 までの外辺にあわせて広がっています。

すべての辺に合わせて曲面を広げるには、 v 軸も Endpoint に設定します。

Order

(次数)

このプロパティも [NURBS 曲線](#) と同じもので、制御点が曲面形状内の曲線の計算にどの程度関わるかを決めます。次数を高くするほど(図「次数 2 と次数 4 の曲面」の 1) 曲面が制御点から離され、滑らかな曲面を作ります(解像度が十分高い場合を想定)。次数を最低値にすると(同じ図の 2) 曲面は制御点をなぞり、グリッドに追従しがちな曲面を作ります。



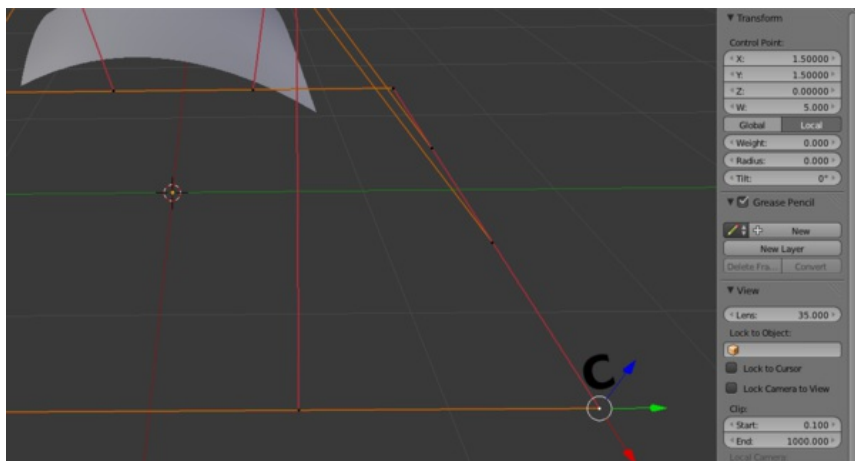
次数 2 と次数 4 の曲面

説明用に図の次数 2、次数 4 の曲線はどちらも knot ベクトルを Endpoint(終点)にして、曲面を両方向の端まで広がっています。

補間軸ごとに違う次数をセットでき、曲線と同じで次数を 2 より小さくしたり、6 または対応する軸にある制御点の数より大きくすることはできません。

Weight

(重み)



重みって何でしょう？ そうです、[NURBS 曲線](#) とまったく同じように動作します！ Weight(重み)は各制御点が曲線を引き寄せる度合いを決めます。

図「重み 5 の曲面」では制御点 c の Weight が 5.0 に設定されています。他はすべてデフォルトの 1.0 です。ご覧のように制御点が曲面を自分のほ

うに引き寄せています。

もし全ての制御点と同じ Weight を持っていれば、お互いに効果を打ち消します。曲面を制御点に近づけたり離したりするのは、Weight の差なのです。

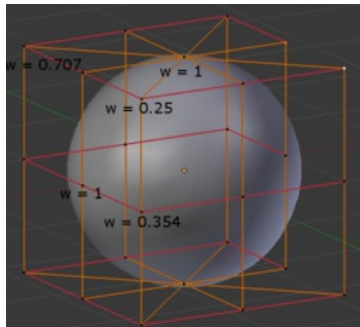
特定の制御点の Weight は [Transform Properties \(トランスフォームプロパティ\) パネル \(N\)](#) の Control Point: (制御点) の W 欄で調整できます (Weight 欄ではありません...)

Preset Weights

(重みのプリセット)

重みのプリセットの場所

Blender 2.4 にあった重みのプリセットは、2.6 では実装されていないか、少なくとも同じ場所にはありません。



球曲面

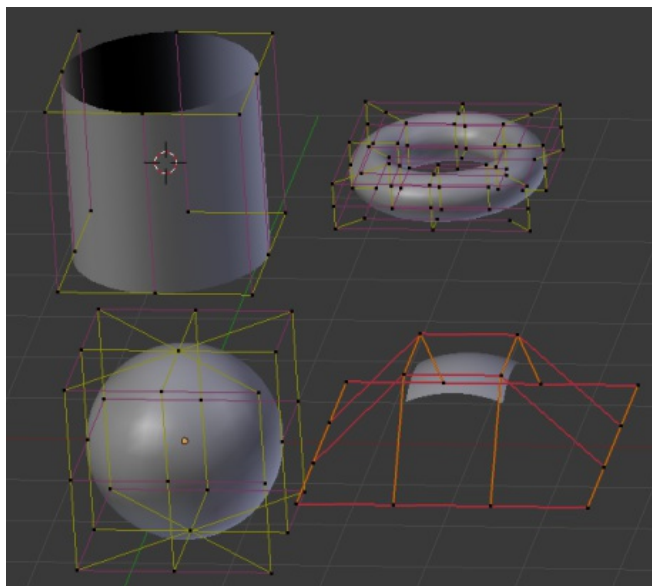
NURBS は円、円柱、球のような純粋な形状を作ることができます (なおベジエ円は純粋な円ではありません)。純粋な円、球、円柱を作るには、制御点の重みを決まった値にしなければいけません。いくつかはプリセットとして Curve Tools パネルで利用できます (右下の隅です)。直感的ではないので、始める前に NURBS についてよく知っておく必要があります。

1D の [NURBS 曲面 \(2.4, 英文\)](#) で円を作る方法は見たことがあります。ここでは、2D の曲面で球を作る方法を見てみましょう。考え方は同じです。球を作るには 4つの重みが必要で (1.0、 $0.707 = 0.5$ の平方根、 $0.354 = 2$ の平方根/4、0.25)、この 4つの値は Curve Tools パネルで利用できるプリセットでした。

Primitives

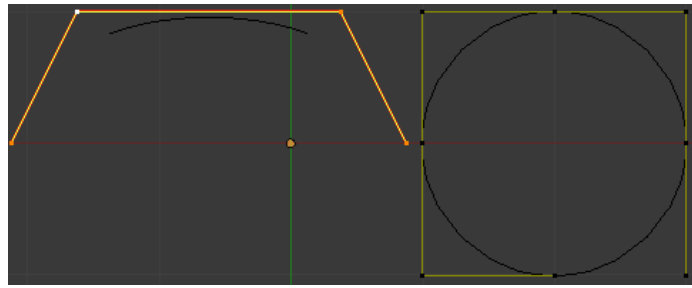
(基礎形状/プリミティブ)

曲面を作り始めるとき役立つのが 4つの NURBS曲面のプリセット NURBS Surface (NURBS曲面)、NURBS Tube (NURBSチューブ)、NURBS Sphere (NURBS球) そして NURBS Torus (NURBSトーラス) で、Add (追加) » Surface (サーフェス) メニューから選べます。



NURBS 曲面のプリミティブ

NURBS 曲面の曲線 (どの V 行にも一つだけ制御点があります) のプリセットも 2つあります。NURBS Curve (NURBS曲線) と NURBS Circle (NURBS円) です。






NURBS 曲線プリミティブ

「本当の」曲線でできた円と違って、円形の NURBS 曲面に面を張ることはできないので注意してください。

Surface Selection

(曲面/サーフェスの選択)

編集モードでの曲面の選択は [NURBS 曲線の選択](#) によく似ています。基本的なツールは [メッシュ](#) のものと同じです。RMB -クリックで単一の制御点を選ぶことができ、現在の選択部に加えるには  Shift RMB -クリックや B (ボーダー選択) を使う、といった具合です。

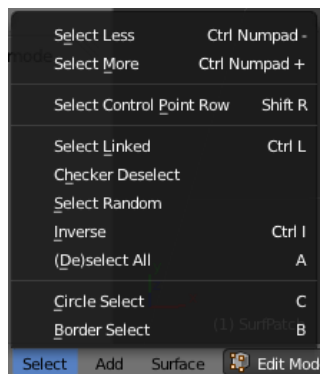
L (または CtrlL) は、マウスカーソルから一番近い制御点とそこに繋がるものすべて (たとえば同じ曲面に属するすべての点) を選択部に加えます。

Select Menu

(選択メニュー)

3D ビューヘッダーの Select (選択) メニューは、曲線のものよりシンプルになっています。

これらのオプションにはすべて、[オブジェクトモード](#) のものと同じ意味があり、同じふるまい方をします。(編集モードでのボーダー選択の特殊性については、[こちら](#) に書かれています)。



Every Nth

(N番目ごとに選択)

モード: Edit (編集) モード

ホットキー: なし


メニュー: Select (選択) » Every Nth (N番目ごとに選択/日本語UIでは「チェッカー選択」)

[曲線の選択](#) 用のものと同じオプションです。ただ、NURBS 曲面の「檻」は2次元であるため、N (「選択間隔」) パラメータの効果はとてわかりづらく思えます...


Control Point Row

(制御点行)

モード: Edit (編集) モード

ホットキー:  ShiftR

メニュー: Select (選択) » Control Point Row (制御点行)

メッシュにおける [edge loop 選択](#) に少し似た動作をします。アクティブな (最後に選択された) 制御点を基準に、制御点の [行](#) 全体を選択します。はじめに  ShiftR を押すと、アクティブな点を通る (含む) V行が *現在の選択部に加わります*。このショートカットをもう一度使うと、この点の U行とV行を交互に切り替えます。このとき *その他一切は選択部から取り除かれます*。

More and Less

(より多く/より少なく)

モード: Edit (編集) モード

ホットキー: Ctrl+ NumPad/Ctrl- NumPad

メニュー: Select (選択) » More/Less (より多く/より少なく)

これら2つのオプションは相補的で、[メッシュ用のもの](#) によく似ています。現在選択されている制御点を基準に、選択部を増減させるためのものです。

アルゴリズムはメッシュに使う場合と同じです:

- More (より多く): 選択されている制御点それぞれに対して、繋がった点を **すべて** (2~4個) 選択します。
- Less (より少なく): 選択されている制御点それぞれに対して、この点に繋がった **すべての** 点を選択されていれば、選択されたままにします。選択されたその他の制御点はすべて、選択を解除します。

このことから次のことが言えます:

- 第1に、曲面の **すべての** 制御点が選択されていれば、何も起きません(Less については、繋がった点がすべて選択されているので何も起きません。また当然ながら More では何も追加できません)。逆に、制御点がひとつも選択されていないときにも同じことが言えます。
- 第2に、このツールが曲面の「外側に及ぶ」ことは決してありません(同一オブジェクト内の他の曲面に「ジャンプする」ことはありません)。

Surface Editing

(曲面の編集)

曲面編集のツールとオプションは、曲線のものほど多くありません。また共通点を多く持ちます。このためこのページでは、曲面編集の基礎から retopology のようなもっと高度なものまで、すべての話題を扱います(扱えるように努めます)。

Basic Surface Editing (translation, rotation, scale)

基本的な曲面編集(移動/回転/拡大縮小)

モード: Edit(編集) モード

ホットキー: G/R/S

メニュー: Surface(曲面) » Transform(トランスフォーム) » Grab/Move(つかむ/移動), Rotate(回転), Scale(拡大縮小), ...

1つ以上の制御点を選択すれば、これに対して移動(G)、回転(R)や拡大縮小(S)の操作を行います。[3D空間での操作](#) で書かれているように、Blender の他の多くのものと同じ要領です。

編集 モードでこうした基本的な操作を行っているときは、オプションが増えます: [プロポーション編集](#) です。

Advanced Transform Tools

(高度なトランスフォームツール)

モード: Edit(編集) モード

メニュー: Surface(曲面) » Transform(トランスフォーム)

トランスフォームツール To Sphere(球に変形)、Shear(せん断)、Warp(湾曲)と Push/Pull(押す/引く) については [メッシュ編集](#) の章で書かれています。曲面専用のトランスフォームツールはありません。

NURBS Control Points Settings

(NURBS 制御点の設定)

モード: Edit(編集) モード

パネル: Transform Properties(トランスフォームのプロパティ)


[前のページ](#) で書かれていたように、NURBS 制御点はそれぞれウェイト(weight)を持ちます。ウェイトとは、制御点の曲面への影響度です。セットするには Transform Properties(トランスフォームのプロパティ) パネル(N)の weight(ウェイト) 数値欄に直接値を入力します。

Adding or Extruding


(追加と押し出し)

モード: Edit(編集) モード

パネル: Surface Tools(サーフェスツール)

ホットキー: E (または Ctrl LMB )

メニュー: Surface(曲面) » Extrude(押し出し)


メッシュや曲線と違い、普通は(Ctrl LMB ) クリックで 曲面に新たな制御点を直接追加することはできず、U行か V行全体をいちどに追加することのみ、広げることができます。唯一の例外は NURBS 曲面曲線(surface curve、各 U行か V行の制御点数がひとつだけの曲面)での作業中です。この特殊なケースでは、すべてが [曲線](#) とまったく同じように動作します。

ほとんどの場合は押し出しだけが利用できます。いつものように、一旦ツールが起動されるとすぐに押し出しの効果が生じて つかむ/移動モードになり、新たに押し出された曲面を目的の場所までドラッグできる状態になります。

非常に重要なことが二つあります:

- 曲面は **2次元** のオブジェクトです。したがって曲面の **中**にあるもの(行の「内部」にあるもの)は何も押し出せません。無意味です!
- 制御「グリッド」は「**四角張ったままである必要があります**」。これは、飛び飛びに行の一部だけを押し出すことはできず、行全体だけが対象になるということです。

まとめると、Extrude ツールは曲面の外縁にある行全体が選択されたときにだけ動作します。それ以外では何も起きません(訳注: 2.6 では自動的に Grab(つかむ/移動) の操作になるようです)。

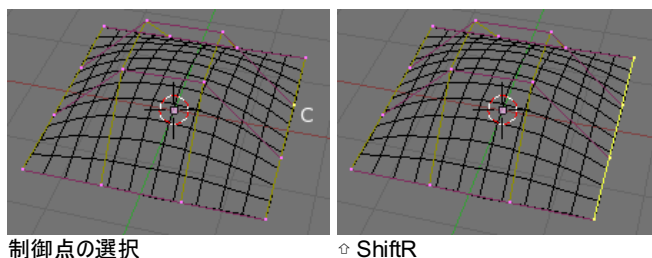
オブジェクト内で何も選択せずに CtrlLMB  クリックするだけでは、出し抜けに新たな曲面を作成することはできません。曲面を複数の部分にわたる「切り分け」の機能はありません。したがって既存部分を複製するか、Add(追加)メニューから新たなものを追加するかしないと、新たな曲面を作成することはできません。

Examples

(例)

図「制御点の選択」から「完成」は、曲面の側部にそった典型的な押し出しの様子です。

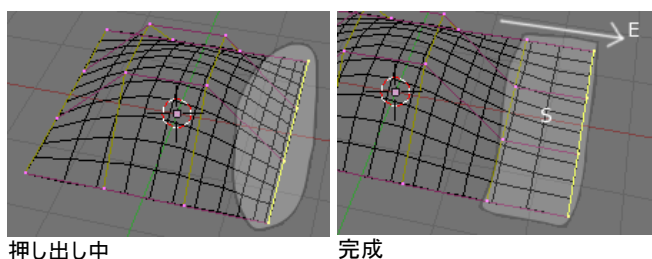
図「制御点の選択」と「⇧ ShiftR」では、“c”で示された制御点を選択することにより、外縁の制御点行がハイライトされています。次に、便利な行選択ツール(⇧ ShiftR)を使って残りの制御点を選択しています。



制御点の選択

⇧ ShiftR

次に、端を E を使って押し出しました(図「押し出し中」)。ハイライトされた辺の隣に、メッシュがどのように集まったか見てください。問題の領域は薄い灰色で丸くハイライトされています。押し出された新たな部分の曲面もそこに集まっているからです。



押し出し中

完成

新たな部分をこの領域から離すことで、曲面は「離散」しはじめます。動かした方向を“E”、新しい部分を“S”で示しています。

モデルが最終的な形状に達するまで、この新たな曲面の押し出し(あるいは追加)処理を続けることができます。

Opening or Closing a Surface

(曲面の開閉)

モード: Edit(編集) モード

パネル: Surface Tools(サーフェスツール)

ホットキー: AltC

メニュー: Surface » Toggle Cyclic(ループ切り替え)

曲線と同じく、曲面は閉じた状態(cyclic/環状)と開いた状態にできます。曲面は2次元であるため、このプロパティは U および V 軸それぞれに対して調整できます。

単独の軸に沿った曲面の環状プロパティを切り替えるには AltC を使い、切り替えのポップアップメニューで cyclic U(日本語UIではループ U) か cyclic V(ループ V) のいずれかを選んでください。曲面の外縁にある対応する辺が結合し、「閉じた」曲面を作ります。

内側と外側

曲面には「内側」と「外側」の面があります。前者は黒くなるのに対して、後者は正しく陰影がつきます(曲面用の「両面(double sided)」のシェーディングオプションは見当たりません...)。曲面を一方か両方に閉じると真っ黒なオブジェクトができるかもしれません! この場合は「方向」の切り替えを行なってください...

Duplication

(複製)

モード: Edit(編集) モード

パネル: Surface Tools(サーフェスツール)

ホットキー: ⇧ ShiftD

メニュー: Surface(曲面) » Duplicate(複製)

メッシュや曲線と同様、この命令は選択部を複製します。いつものように複製が選択されてつかむ/移動モードに入るため、他の場所に動かすことがで

きます。

しかしながら曲面に関しては、選択部の複製ができない場合もあります。この場合単につかむ/移動モードに入るだけです... (訳注: 2.6 では複製できない旨のエラーが出ます)。実際のところ、*単独の有効なサブグリッド*を作る選択部だけが複製可能です。実例を見てみましょう:

- 単独の制御点を複製することができます。ここから U軸に沿って「曲線曲面」を「押し出す」ことができます。さらにこの単独の U行を V軸に沿って押し出し、新しい本当の曲面を作ることができます。
- 単独の、行の中の連続した一部、または行全体は複製できます。この結果、もし V行 (の一部) を選択していたとしても、新しい **U行** ができます。
- 単独のサブグリッド全体を複製できます。

別々の、有効な「サブグリッド」(単独の点であっても)を一度に複製しようとしてもできません。ひとつずつ行う必要があります...

Deleting Elements

(要素の削除)

モード: Edit(編集) モード

パネル: Surface Tools(サーフェスツール)

ホットキー: X または Del

メニュー: Surface(曲面) » Delete...(削除)

曲面の Erase ポップアップには二つのオプションがあります:

(訳注: バージョン 2.67 か 2.68 あたりでオプションが Vertices(頂点)と Segments(セグメント)の二つに変わりました。頂点は選択点を含めて削除します。セグメントは、セグメントだけを削除して両端の点を残すようですが曲面曲線以外での動作は不明です)

Selected

曲面を壊さずに選択行を削除します(行を削除するとすぐに、削除した行の隣にあった行同士が直接繋がります)。選択部は必ず次のルールに従わねばなりません:

- 行全体が(行全体だけが)選択されなければなりません。
- 同じ軸上の行のみが選択されなければなりません(すなわち U行とV行の両方を同時に削除することはできません)。

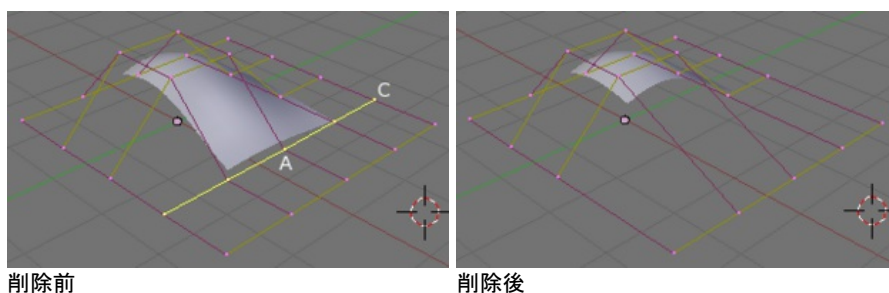
また NURBS の次数(order)は指定軸上にある制御点の数より高くできないので、制御点を削除すると減る可能性があることを覚えておいてください。もちろん、行がただひとつだけ残っているときは曲面は「曲面曲線」になります。一点だけが残されると、曲面は目に見えなくなります。そしてすべての点が削除されると、曲面そのものが削除されます。

All

メッシュや曲線に使うのと同様、オブジェクト内のすべてのものを削除します!

Example

(例)



図「*削除前*」でははじめに“A”のラベルのついた制御点を選択し、 \diamond ShiftR を使って残りの制御点を選択して、制御点の行が一行選ばれています。次に [Erase\(削除\)メニュー](#) (X) を使うと、*選択された* 制御点の行は削除されました。この結果が図「*削除後*」です。

Joining or Merging Surfaces

(曲面の結合と統合)

モード: Edit(編集) モード

ホットキー: F

メニュー: Surface » Make Segment(セグメントを作る)

二つの曲面を統合(merge)するためには、[曲線](#)と同様、それぞれの曲面の端部、すなわち外縁上の制御点の行がひとつ選択されている必要があります。これは、二つの曲面は同じオブジェクトの一部である必要があるということです。たとえばオブジェクトモードでは二つの曲面を結合できません。しかしもちろん、同種の他のオブジェクトと同様、[二つ以上の曲面オブジェクトを結合\(join\)](#)して一つのオブジェクトにすることはできます(CtrlJ)。ひとつに「繋がったり(link)」「統合されたり(merge)」することはまったくありません...はい、ちょっと紛らわしいのです!

この命令はメッシュに辺や面を作るのと同等で(ゆえにショートカットになります)、編集モードでのみ動作します。選択部には同じ解像度の(制御点の数が同じ)外縁上の行だけが含まれる必要があります。そうでなければ Blender は何と何を統合すべきか可能な限り推測しますが、それもできないと統合は失敗します(何もメッセージが出ないか、点の数が異なる行が選択されていれば「解像度が合いません(Resolution doesn't match)」,ひとつの曲面の点だけを選択していれば「統合する選択部分が少なすぎます(Too few selections to merge)」と表示されます)。

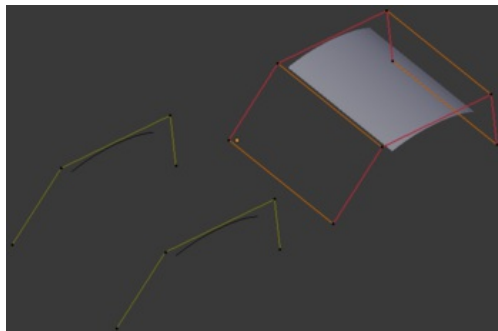
問題を避けるため、常に外縁上の同数の点を持つ行だけを選ぶようにしてください。なお、片方の曲面で外縁上の U 行を、もう片方で外縁上の V 行を選ぶと、Blender は片方の曲面の軸を自動的に「反転」させて、正しく合うようにします。

NURBS 曲面曲線 (surface curve) はよく、オブジェクト全体の横断面を定義して、船体のようなものを作るのに使われます。上述のように感じがよく滑らかで調和のとれた形状を作るためには、これに「スキニング」する必要があります。詳しい工程については [こちらのチュートリアル \(eng 2.4\)](#) をご覧ください。

Examples

(例)

図 (結合可能) は二つの NURBS 曲面曲線の例です (NURBS 曲線 **ではありません**)。編集モードで結合できる状態です。図 (結合完了) は二つの曲線を結合した結果です。



結合可能/結合完了

Subdivision

(細分化)

モード: Edit(編集) モード

パネル: Surface Tools(サーフェスツール)

ホットキー: W » 1 NumPad

メニュー: Surface(曲面) » Segments(セグメント) » Subdivide(細分化), Specials(スペシャル) » Subdivide(細分化)

曲面の細分化は非常に単純です。Specials(スペシャル) メニュー(W)の項目 Subdivide(細分化) または Curve Tools パネルの Subdivide(細分化) ボタンを使うと、**完全選択されたグリッド**をすべて、それぞれの「四角」を4つの小さな四角に分けることにより、細分化が1度行われます。

1D の曲面(「曲面曲線」)に対して使うと [曲線](#) に使うのとまったく同じように動作します。

Spin

(スピン)

(訳注: Blender 2.6 では曲線/曲面に直接スピンツールを使えないようです。Screw 値を 0 にした [スクリューモディファイア](#)を使うか、曲線/曲面をメッシュに変換してから [スピンツール](#) を使います)

Switch Direction

(向きの反転)

モード: Edit(編集) モード

パネル: Surface Tools(サーフェスツール)

ホットキー: W » 2 NumPad

メニュー: Surface(曲面) » Segments(セグメント) » Switch Direction(向きの反転), Specials(スペシャル) » Switch Direction(向きの反転)

少なくとも一要素が選択されている曲線の向きを、「逆転」させます(始点が終点になり、逆もまた同様です)。主に、曲線をパスとして使うとき、あるいは bevel と taper オプションを使うときに役立ちます。

Other Specials Options

(その他の特殊な機能)

モード: Edit(編集) モード

ホットキー: W

メニュー: Specials(スペシャル)

Specials メニューには [曲線](#) 用のものとまったく同じ追加オプションがあります。—ただ、Set Radius(半径を設定)と Smooth Radius(半径を滑らかに)は何もしないように思えます...

Conversion

(変換)

NURBS 曲面だけしかないので、「内部」変換は存在しません。

しかしながら、オブジェクトモードでのみ動作する曲面からメッシュへの「外部」変換を利用できます。これは曲面の解像度を両方向に使うと、面、辺、頂点を作り、Surface(曲面) オブジェクトを Mesh(メッシュ) オブジェクトに変換します。

Retopology

(リトポロジー/接続形態の再編)

曲面の構成要素への吸着機能は、メッシュや曲線のものと同じです。さらなる情報は [Retopology](#) をご覧ください。

Misc Editing

(その他の色々な編集)

メッシュ用のもの、あるいは オブジェクト モードで使えるものと同じオプションを使えます。指定した曲面を [分離\(separate\)](#) したり(P)、別の選択オブジェクトを1つまたは3つの制御点の [子](#) にしたり(CtrlP – なお、3つの制御点への親子化は曲線に対して変な動作をします...)、[フック\(hooks\)](#) ([eng](#)) を追加して他のオブジェクトを使って制御点を操作できるようにしたりできます。

Mirror ツールも利用できます。[メッシュの頂点](#) に使った場合とまったく同じ動作をします。

Text Objects

(テキストオブジェクト)

モード: 編集モード (テキスト)

パネル: Curve(曲線)とSurface(サーフェス), Font(フォント) および Char(文字)

メニュー: Add(追加) » Text(テキスト)



テキストの例

テキストオブジェクトはまったく名前のとおりのもので、テキストはテキストをいくつか含みます。現代的なフォント(OpenType、TrueType 他)がベクトル的で、曲線で作られているように(一般的にはベジエ曲線)、テキストオブジェクトは曲線や曲面と同じ型のオブジェクトです。

Blender は文字コードとそれを3Dビューで表現するオブジェクトの対応を管理するために「フォントシステム」を使います。フォントシステムは *組み込みの* フォントを持っていますが、それだけでなく *PostScript Type 1*、*OpenType* それに *TrueType* を含む外部フォントを使うこともできます。つまり Blender でもそれらのフォントを使えるということです。さらに、現在の .blend ファイルにあるオブジェクトはすべて、文字として使うことができます...

Bender では 2D と 3D のテキストを作成/レンダリング可能で、後述のように、自由に陰影をつけたり、多様な配置オプションを使ったり(両端揃えやフレームなど)できます。デフォルトでは、文字はちょうど閉じた 2D 曲線と同じで平らな面を持つサーフェスですが、もちろん押し出すことができます...。そして他の曲線に沿わせることができます。

テキストの形状に満足したら、(オブジェクトモードで AltC を使って)曲線に変換したり直接メッシュに変換したりして、そのオブジェクトの強力な機能をすべて使うことができます。

画像「テキストの例」はさまざまな活用例で、例えば「青い」フォントを曲線のパスに沿わせています。

Notes

(注意点)

各テキストオブジェクトには最大 **50000** 文字が入りますが、テキストオブジェクトの持つ文字が増えるほど、オブジェクトの操作反応は悪くなる点に注意してください。

オブジェクトモードと編集モードを切り替えてみると、フォントパネル(Object Data)は変化しないことがわかります。これはフォントパネルの設定はどちらのモードでも同じように使われるということです。つまり、メッシュの一部だけに用いるようなことはできません。したがって、フォントやフォントサイズなどは、一つのテキストオブジェクト内のすべての文字で同じです。例外が一つだけあります。Bold/Italic ボタンは文字ごとにプロパティを操作します(この方法で一つのテキストで4つまでの別のフォントを使えます)。

リソース使用の最適化のため、(文字セット全体ではなく)実際に使われている文字だけがメモリを消費します。

Editing Text

(テキストの編集)

モード: 編集モード

ホットキー: 下記参照



編集モードのテキスト

テキストの編集は、Blender の他のオブジェクトとかなり違って、主に2箇所で作業します。1つはもちろん文字を入力する 3Dビューで、[スタイル適用](#)などのショートカットが用意されています。ただし、編集モードで使えるホットキーの大半はテキストでは使えないことに注意してください！ もう1つはプロパティウィンドウの **フォントパネル** (Object Data (オブジェクトデータ) コンテキスト) です。

3D ビューのヘッダーにあるメニューはほぼ使いません。また スペシャル (Specials) メニューもありません…。トランスフォームやミラーツールも使えません。一方、曲線用のものと同じモディファイアを使えます。

テキストの編集は、通常のテキストエディタを使うのと似ていますが、機能は充実しておらず、違いもあります

訳注: Blender 2.67 のテキストオブジェクトと日本語テキスト

Windows 版とMac OS X 版の Blender 2.67のテキストオブジェクトは、日本語インプットメソッドを通じた文字入力に未対応です。ただし、ファイルから日本語をペーストする (編集モードで Edit (編集) » Paste File (ファイルの貼り付け)) ことや、Blender 内部のテキストエディタに入力したものを交換して使う (テキストエディタウィンドウで Edit (編集) » Text To 3D Object (テキストを3Dオブジェクト)) ことはできます。いずれにしても Object Data の Font パネルで、(Bfont以外の) 日本語を表示可能なフォントを指定しないと表示されません

編集モードを抜ける

⇔ Tab はテキストにタブ文字を挿入するのではなく、他のオブジェクト型と同様 編集モードから出たり入ったりするのに使います。

コピー

文字をバッファにコピーするには CtrlC またはツールシェルフの **コピー (Copy)** ボタンを使います。

カット

文字をバッファにカットするには CtrlX またはツールシェルフの **カット (Cut)** ボタンを使います。

ペースト

バッファから文字をペーストするには CtrlV またはツールシェルフの **ペースト (Paste)** ボタンを使います。

すべての文字の削除

Ctrl← Backspace を使います。

Home/End

↶ Home と ↷ End はカーソルを行頭/行末に移動します。

次/前の語

カーソルを単語の境界に移動するには Ctrl← または Ctrl→ を使います。

テキストバッファはデスクトップとやりとりせず Blender 内だけで動作します。Blender の外からテキストを挿入するには、この下の [テキストの挿入](#) をご覧ください。

Inserting Text

(テキストの挿入)

テキストの挿入方法は3つあります。内部のテキストバッファから行う方法 ([テキストの編集](#)) と、テキストファイルから行う方法です。

テキストファイルから読み込むには、Text (テキスト) » Paste File (ファイルの貼り付け) ツールを使います。これで有効な UTF-8 ファイルを選ぶためのファイルブラウザウィンドウが表示されます。例によって、ファイルの文字数が多すぎると操作の反応が悪くなるので注意してください。

Special Characters

(特殊文字)

モード: 編集モード

メニュー: Text (テキスト) » Special Characters (特殊文字)

Alt キーか 3D ウィンドウヘッダーの Text (テキスト) メニューを使うと利用できる特殊文字がいくつかあります。

以下は特殊文字の一覧です:

AltC:	コピーライト (©)	AltR:	登録商標 (®)
AltG:	度 (°)	AltX:	掛け算記号 (×)
AltS:	ドイツの“ss” (ß)	AltF:	通貨記号 (¤)
AltL:	イギリスのポンド (£)	AltY:	日本の円 (¥)
Alt1:	上付き文字の 1 (¹)	Alt2:	上付き文字の 2 (²)
Alt3:	上付き文字の 3 (³)	Alt:	円
Alt?:	スペイン語の疑問符 (¿)	Alt!:	スペイン語の感嘆符 (¡)
Alt<:	左二重引用符 («)	Alt>:	右二重引用符 (»)

キーボードにある文字は、強制母音などを含めてすべて受け付けます。特殊文字 (アクセント記号付きの文字など、US キーボードにないもの) の多くは2文字を組み合わせて打つことができます。メインの文字を入力し、Alt← Backspace を押し、それから特殊文字を作るのに必要な「修飾子」を入力します。以下はその例です:

A, Alt← Backspace, ~:	ã A, Alt← Backspace, ':	á A, Alt← Backspace, `:	à
A, Alt← Backspace, O:	â E, Alt← Backspace, ":	ë O, Alt← Backspace, /:	ø

Convert text to text object

(テキストのテキストオブジェクトへの変換)

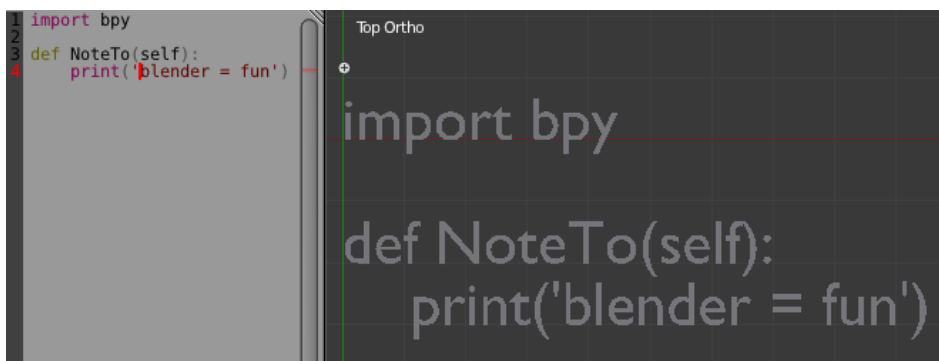


Blender でテキストを得る簡単な方法は、テキストエディタに文字を入力することです。ウィンドウを分割して、3D描画域とテキストエディタを同時に見られるようにすることをおすすめします。テキストエディタで Text(テキスト) » Create Text Block(テキストブロックを作成)を選んでから、入力を始めてください。入力が終われば、Edit » Text to 3D Object » 必要に応じて One Object または One Object per Line を選びます。Text » Open Text Block からテキストファイルを読み込むこともでき、大量のテキストを一度に取り込みたいときに便利です。

3D Mesh

(3Dメッシュ)

テキストオブジェクトを 3D メッシュオブジェクトに変換できます。頂点を編集モードで編集できることが役に立つ可能性があります、テキストそのものの編集能力は失われます。変換するには、オブジェクトモードに入ってテキストオブジェクトを選択してください。AltC を押し、Mesh From Curve/Meta/Surf/Text を選びます。これで [編集モード](#) に戻って自分で頂点を編集できます。変換後のメッシュはふつう、少しごちゃごちゃしているので、Limited Dissolve(限定的溶解)による削除か [Remesh\(リメッシュ\)](#) オブジェクトモディファイアを低いしきい値で使ってメッシュをきれいにすることが役立つかもしれません。



左が通常のテキスト、右がそこから作られたテキストオブジェクト

Text Selection

(テキストの選択)



編集モードのテキスト

編集モードでは、白いカーソルは他のテキストエディタと同じく文字の挿入位置を示しています。カーソルは矢印キー(→/↓/←/↑)または Page up/Page down および Home/→ End キー で動かせます。

文字の一部を選択するには⇧ Shift を押しながら矢印キーを動かします。normal/bold/italic の属性指定に使えます。

Formatting Text

(テキストの書式化)

Fonts

(フォント)












モード: 編集モード

パネル: Font(フォント)

フォントパネルには文字の見た目を変えるオプションがあります。

Loading and Changing Fonts

(フォントの読み込みと変更)

	18thCtrInIt.ttf	29 KB
	18thCtrKurStart.ttf	89 KB
	18thCtrKurTxt.ttf	84 KB
	1610_Cancellaresca_lim.TTF	116 KB
	BIRTH OF A HERO.ttf	116 KB
	CRBLATRIAL.otf	45 KB
	CRBLATRIAL.ttf	54 KB
	CRIALTRIAL.otf	59 KB
	CRIALTRIAL.ttf	76 KB
	kabog.ttf	23 KB
	kingthingspetrock.ttf	14 KB

Type 1 フォントファイルの読み込み

Blender はデフォルトで *組み込み* フォントを備えており、4つのフォントスタイルのフォント選択欄に表示されます。この *組み込み* フォントはいつも“BFont”としてこの一覧に表示されます。最初のアイコンを押すと、現在読み込まれているフォントのドロップダウンリストが現れます。フォントスタイルごとにフォントをひとつ選びます。

別のフォントを読み込むには、Font パネルにある **読み込み** ボタンをクリックして、*有効な* フォントを指定します。ファイルブラウザウィンドウでは有効なフォントを大文字の F のアイコンで表示します (画像「Type 1 フォントファイルの読み込み」)。

Unix に関する註記

フォントは通常は `/usr/lib/fonts` かそれに似た `/usr/lib/X11/fonts` などで見つかりますが、いつもそうだとは限りません。フォントは他の場所にもあるかもしれません。たとえば `/usr/share/local` や `/usr/local/share`、あるいは関連するサブツリーです。

Blender の理解できないフォントを選ぶと、エラー“Not a valid font”が発生します。

テキスト内の同じスタイルの文字には同じフォントが用いられ、スタイルごとに別のフォントが必要であることを注意してください。例えば、文字や単語をイタリックにするには、*Italics* フォントを読み込む必要があります。フォントを一旦読み込めば、そのフォントの「スタイル」を選択した文字やオブジェクト全体に用いることができます。各スタイル (**Normal**, **Italics**, **Bold**, **Bold-Italics**) を表現するためには、最低限4つの別のタイプのフォントを読み込む必要があるでしょう。

Blender は、あなたが「normal」「bold」などのスタイル用にどのフォントを読み込むのか、気にすることはありません。これを理解しておくことは大切です。4つまでの別のフォントを同じテキスト内で使えるのは、この仕組みによるものです。ただし、同じフォントの別のスタイルか、別のフォントを選ばなければなりません。フォントパネルには文字のスタイルや配置を変えるためのタイポグラフィのコントロールがたくさんあります。

Size and Shear

(サイズとせん断)

サイズ (Size)

テキスト全体のサイズです (文字ごとのサイズを制御する方法はありません)。ただし、使っているフォント (使っているスタイル、後述) によって見た目の大きさが変わる可能性があります。



shear: 'blender' のせん断の値は 1、'2.59' のせん断の値は 0 です。

せん断 (Shear)

テキスト全体の傾きを操作します。もしイタリックと同じように見えたとしても *これは別のものです!*

Objects as Fonts

(フォントとしてのオブジェクト)

Blender の中に独自の「フォント」を「作る」こともできます！ 非常に複雑な手順なので、詳しく見てみましょう:

- まずは、各文字を作る必要があります。文字はそれぞれがひとつのオブジェクトで、*どのタイプでも構いません* (メッシュ、曲線、メタ...)。次のルールに従って名前をつけます: 共通の接頭辞に続けて 文字名 (例“ft.a”、“ft.b”、など)。
- 次に、テキストオブジェクトを選択してプロパティウィンドウでオブジェクトコンテキストを開き、Duplication (複製) パネルの頂点 (Verts) ボタンを有効にする必要があります。

- フォントパネルにあるオブジェクトフォント(Ob Family)欄をフォントオブジェクトの **共通接頭辞** にします(訳注:上の例で言えばピリオドを含めた".ft."を入力します)。

さてこれで、「フォント」オブジェクト名の文字名部分と一致している文字が文章内にあれば、その文字に重なるようにオブジェクトが複製されます。元の文字は見えなままです。オブジェクトは複製され、対応する文字の右下隅に中心が来るように配置されます。

Text on Curve

(曲線上のテキスト)

[カーブモディファイア](#) を使うと曲線にテキストを沿わせることができます。



曲線上のテキスト

図「曲線上のテキスト」は曲線(2D ベジエ円)によって変形したテキストです。

カーブモディファイアを用いるには、AltC でメッシュを選び、まずテキストオブジェクトをメッシュに変換する必要があります。

註釈

曲線にテキストを沿わせる機能もありますが、カーブモディファイアのほうが多くのオプションを使えます。

Underline

(下線)

下線(Underline)

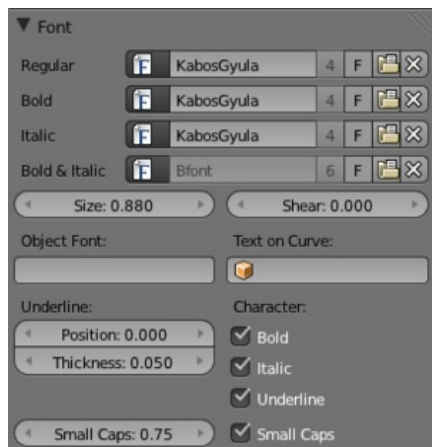
文字入力の前にUnderline(下線)ボタンで切り替えます。文字を選択してツールシェルフの **Underline(下線)** ボタンを押しても下線にできません。

Position

下線の位置を垂直方向にずらす量です。

Thickness

下線の太さです。



たとえば Boldなどの文字(Character)オプションをチェックします

Character

(文字)



Bold text.

太字体 (Bold)

文字入力の前に太字体ボタンで切り替えます。文字を選択してツールシェルフの **太字体** ボタンを押しても下線にできます。

斜字体 (Italics)

文字入力の前に斜字体ボタンで切り替えます。文字を選択してツールシェルフの **斜字体** ボタンを押しても下線にできます。

下線 (Underline)

上述の下線設定の指定どおりに下線を有効にします。

小型大文字 (Small Caps)

小型大文字を入力します。

Blender の 太字体 と 斜字体 ボタンは他のアプリケーションとは違って、特定のフォントを独自に読み込むためのプレースホルダーとして動作し、対応するスタイルが定義されたときに使われます。[上述](#) もご覧ください。

太字体/斜字体/下線の属性を文字に用いるには、文字入力の前に 太字体/斜字体/下線 をチェックしておくことも、先に文字をハイライト(選択)して太字体/斜字体/下線を切り替えることもできます。

Setting Case

(大文字小文字の切り替え)

大文字小文字の切り替えは、文字を選択してツールシェルフにある **大文字 (To Upper)** か **小文字 (To Lower)** を押すと行えます。

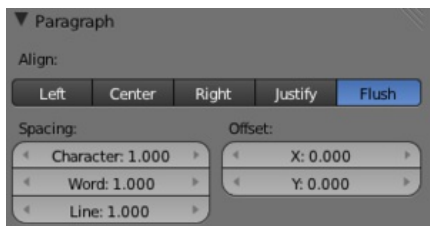
小型大文字 (Small Caps) オプションを有効にすると文字を小型大文字として入力します。

小型大文字のサイズは小型大文字のサイズ (Small Caps Scale) 設定で変えられます。小型大文字のサイズは Small Caps (小型小文字 の書式を持つすべての文字に適用されることに注意してください)。

Paragraph

(段落)

段落 (Paragraph) パネルには文字揃えと文字間隔の設定があります。



段落タブ

Align

(文字揃え)

左 (Left)

テキストボックスが使われていればフレーム内で左寄せし、使われていなければ テキスト オブジェクトの中心をテキストの始点として使います (右に延びていきます)。

中心 (Center)

テキストボックスが使われていればフレーム内で中央寄せし、使われていなければ テキスト オブジェクトの中心をテキストの中心として使います (左右両方に均等に延びていきます)。

右 (Right)

テキストボックスが使われていればフレーム内で右寄せし、使われていなければ テキスト オブジェクトの中心をテキストの終点として使います (左に延びていきます)。

両端揃え (Justify)

ワードラップ (↵ Enter)によるものとは**違います**)によって行が終わるときにだけ文字寄せを行います。行を埋めるのに **文字間隔の調整 (カーニング)**ではなく **空白文字**を使います。

均等割 (Flush)

常に、文字入力中であっても行の文字寄せを行います。行を埋めるのに文字間隔の調整 (カーニング)を使います。

両端揃えと均等割はどちらもテキストボックス内でのみ動作します。

Spacing

(文字間隔)

文字 (Character)

文字間隔です。文字幅単位で指定します

単語 (Word)

(英文の) 単語間隔です。文字幅単位で指定します。選択部分の間隔を Alt→ か Alt← を押して **0.1** ずつ増減できます。

行 (Line)

縦方向の行間隔です。

Offset

(オフセット)

X オフセットと Y オフセット

テキストの通常の配置に対する X オフセットと Y オフセットを設定します。[テキストボックス](#) を使うと、ボックスの内容すべてに用いられることに注意してください...

Shape

(シェイプ)

モード: オブジェクトまたは 編集モード

パネル: Shape (シェイプ)

Shape (シェイプ) パネルには曲線とほぼ同じオプションがあります。

Resolution

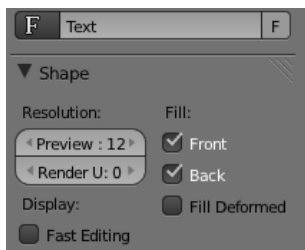
(解像度)

プレビュー (Preview)

ビューポートでの [解像度](#) です。

レンダラー (Render)

レンダラーでの [解像度](#) です。



シェイプの設定

高速編集モード (Fast Editing)

編集モードで曲線の面張りを無効にします。

Fill

(面張り)

面張りオプションは、ジオメトリ (Geometry) パネルで押し出し (Extruded) か ベベル (Beveled) が使われているテキストの、曲線の面張りの方法を決めます。

前 (Front)

サーフェスの前面に面を張ります。

後ろ (Back)

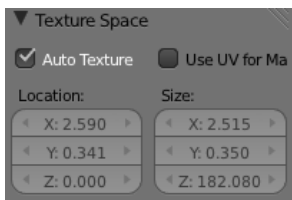
サーフェスの背面に面を張ります。

Fill Deformed

シェイプキーとモディファイアの適用後に面を張ります。

Textures

(テクスチャ)



テクスチャ設定

マッピングに UV を使う (Use UV for Mapping)

生成テクスチャの座標に UV 値を使います。

自動テクスチャ空間 (Auto Texture Space)

オブジェクトのトランスフォーム時、アクティブオブジェクトのテクスチャ空間を自動的に調整します。

Geometry

(ジオメトリ)

テキストオブジェクトは [曲線の押し出し機能](#) をすべて持っています。

Text Editing

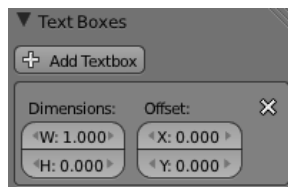
(テキストの編集)

Text Box

(テキストボックス)

モード: オブジェクトまたは 編集モード

パネル: Font(フォント)



テキストボックス

テキストの「ボックス」を使うと、単体のテキストオブジェクト内のテキストを、四角形の枠内で文字揃えできるようになります。テキストボックスは自由に位置を決められ、大きさを変えることができ、1つのテキストオブジェクトにいくつでも作れます。

テキストは低い番号のボックスから高い番号の枠に連続して流れ、それぞれの枠内で折り返し (word-wrap) が行われます。テキストは、低い番号の枠にもう入らなくなった時、次の枠に流れます。最後の枠に達すると、それ以上のテキストは範囲外にあふれます。

テキスト枠 (text frame) の概念は、Scribus のような DTP アプリケーションのものと非常に似ています。枠を使ってテキストの配置や流れを制御できません。

枠は Text Boxes (テキストボックス) パネルで操作します。

Frame size

(枠の大きさ)

デフォルトでは新しいテキストオブジェクトの最初の枠や、追加した枠のサイズは、幅 (Width)、高さ (Height) とともに **ゼロ** です。つまり、枠は初めは見えていません。

0.0 の幅を持つ枠ではテキストの折り返し起きません。**0.0** の高さを持つ枠は折り返しを無限回行い、次の枠にテキストが流れません。

枠を目に見えるようにするには、枠の幅を **0.0** より広くする必要があります。

註記

フォント自身に高さがあるため、事実上、高さが **0.0** になることはありません。



枠の幅

図「枠の幅」**5.0** の幅を持つテキストオブジェクトです。枠の幅が **0.0** より広いと可視化され、アクティブなテーマ色で点線の四角形が描画されます。テキストが枠からあふれていますが、これはテキストがデフォルトの枠 (最後の枠です) の端に達したためです。

Adding/Deleting a Frame

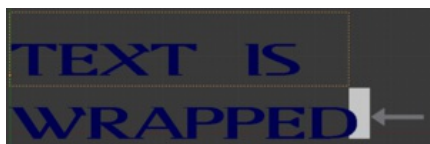
(枠の追加と削除)

枠を追加するには、Text Boxes (テキストボックス) パネルにある **Add Textbox** (テキストボックスを追加) ボタンを使います。新しい枠が現在の枠のすぐ後ろに (テキストの流れる順番に) 位置とサイズ属性を持って追加されます。新しい枠の X と Y 欄でオフセットを変更してください。X だけを変更すると新たな列を作れます。

現在の枠を削除するには **Delete** ボタン (×印) を押します。高い番号の枠にあるテキストは低い番号の枠への下向き流し込み処理をやり直します。

Example: Text Flow

(例: テキストの流れ)



折り返し処理

2つ以上の枠を使ってテキストを細かな精度でまとめることができます。例として、テキストオブジェクトを作って “Blender is super duper” と入力します。このテキストオブジェクトには枠がありますが、幅が **0.0** なので見えていません。

幅を **5.0** にします。枠が見えるようになり、テキストは新たな幅にあわせて折り返されます(図「折り返し処理」)。テキストが枠からあふれていることに注意してください。これは、デフォルトの枠がたまたま最後の枠でもあったためです。

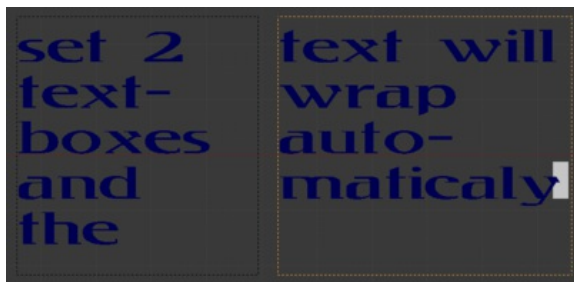


枠 1 から 枠 2 へのテキストフロー

別の枠を追加して幅と高さを設定すると、テキストはその新しい枠に流れるようになります。

Example: Multiple columns

(例: 複数列のテキスト)



Text 5.

2列のテキストを作るには、まずテキストオブジェクトを作って最初の枠の幅と高さを設定したら、枠を追加します。新しい枠は最初の枠と同じサイズになります。X 位置を最初の枠の幅よりいくらか広いか狭い程度にします(図「Text 5」)。

Assigning Materials

(マテリアルの割り当て)

モード: 編集モード

パネル: Materials (マテリアル)

文字ごとに別の Material index (マテリアル番号) を割り当てることができます。

文字を入力しながらでも入力後でも、文字を選択して (Shift + ←/→) マテリアルパネルにある **Assign** (適用) ボタンを押します。



red green blue

例えば図「red green blue」のテキストを作るには、3つの別々のマテリアル(と3つの別々のマテリアル番号)が必要です。単語を選択して適用ボタンを押す作業を繰り返し、マテリアル番号を割り当てていきます。図「red green blue」は1つのテキストオブジェクトです。

メタオブジェクト

Mode: Object or Edit modes

Hotkey: ◊ ShiftA

Menu: Add » Meta

メタオブジェクトは、*面を暗示的に定義する*ものです。つまり頂点(メッシュがそうであるように) や制御ポイント(面がそうであるように)によって *明示的に* 定義されるものではありません: *手続き上の存在*です。メタオブジェクトは文字通り、Blenderによってオンザフライ(その場で)計算される数式です。メタの非常に明確な視覚的特性は、流体 *水銀*や、*粘土*のような形といった、“丸い”形状です。さらに、2つのメタオブジェクトがお互いに近づく、それらは互いに影響し始めます。特にゼロ-Gでは、それらは水滴のように“混ざり”、もしくは“融け合わせ”ります(ついでながら、水の流れをモデリングするとき、流体シミュレーションを実行したくない場合、これが非常に便利です)。その後、移動させてお互いを遠ざけると、それらは元の形状に復元されま

す。

これらはそれぞれ、自身の基礎となる [数学的構造](#)で定義され、いつでも Active Elementパネルを使用してそれらを切り替えることができます。

一般的にメタオブジェクトは、特殊効果、またはモデリングのための下地として使用されます。たとえば、メタのコレクションを使用してモデルの初期形状を成形したり、そこから先のモデリングのために別のオブジェクトタイプに変換することもできます(まあ、利用できるのはメッシュだけではありませんが...)。メタオブジェクトはレイトレーシングにも非常に効果的です。

[後述](#)のように、メタオブジェクトの挙動は、Objectモードではわずかに異なることに注意してください。

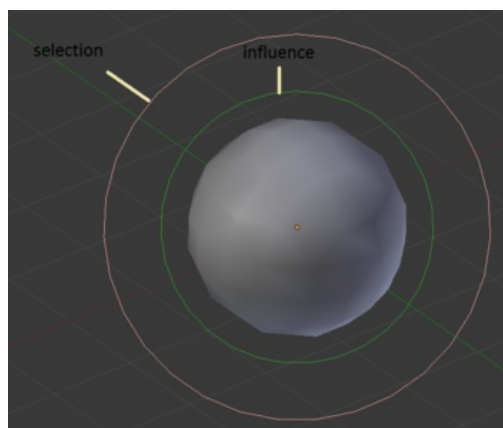
プリミティブ(基本形状)

事前定義されたメタ“プリミティブ”(または外形)は5つあり、Add » Meta サブメニューで利用できます:

- メタボールは、点にもとづく構造のメタを追加します。
- メタチューブは、線分にもとづく構造のメタを追加します。
- メタプレーンは、平面にもとづく構造を持つメタを追加します。
- メタ楕円体は、楕円体にもとづく構造のメタを追加します。
- メタキューブは、容量物の詰まった立方体にもとづく構造のメタを追加します。

可視化

オブジェクトモードでは、計算済みのメッシュが表示され、黒色の“選択リング”が一緒にあります(b選択時はピンクになる)。オブジェクトモードでのメタ情報の詳細については、[下記](#)を参照してください。

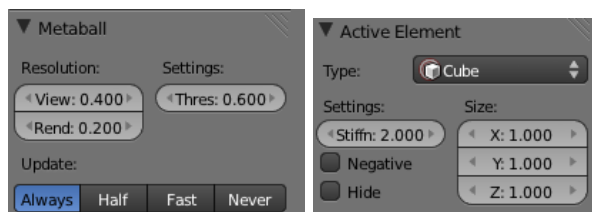


Meta Ball example.

編集モードでは (*Meta Ball example*)、メタはメッシュとして描かれ(シェーディングされたものでも黒いワイヤーフレームでもどちらでも、ただし頂点は全く無い)、色のついた2つの円で囲まれています。赤い円は選択用で(選択状態だとピンクになる)、緑の円はメタのstiffnessを直接制御するためのものです([下記](#)参照 - アクティブ時に薄緑になる)。スケール(S)変換の場合を除き、強調表示された緑の円を持つことは赤を持っていることと等価であることに注意しましょう。

メタボールのオプション

シーン内のすべてのメタオブジェクトは、それぞれ相互作用を及ぼします。メタボールセクションの設定は、すべてのメタオブジェクトに適用されます。Editモードでは、Active Elementパネルが表示され、個々のmeta要素を編集できます。



global meta properties.

individual meta properties.

分割数

Resolutionは、Metalによって生成されるメッシュの分割数を制御します。

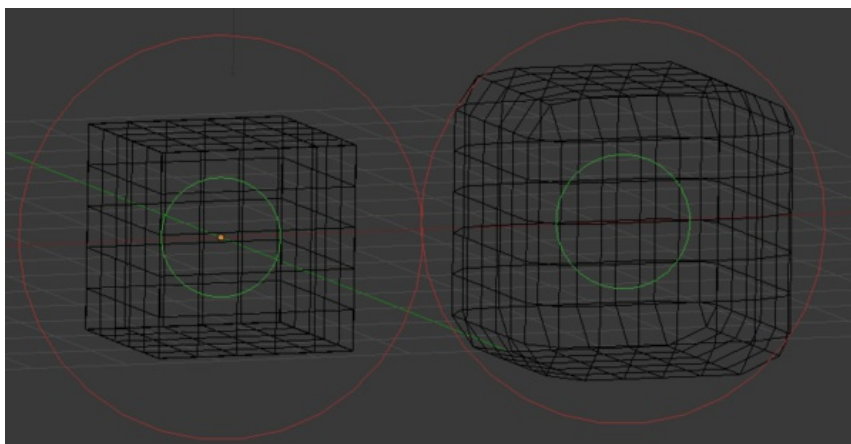
View

生成されるメッシュの3Dビューでの細密さ。範囲は **0.05** (きめ細かい) から **1.0** (粗い) です。

Render

生成されるメッシュのレンダリング時の細密さ。範囲は **0.05** (きめ細かい) から **1.0** (粗い) です。

内部的な数学的構造を確認する方法のひとつとしては、Resolutionを下げ、しきい値を増加させ、Stiffness (下記参照) をしきい値よりも少し上に設定しましょう。(基礎構造) に示したものは、上記のように設定した (メタキューブ) で、次のような構成を適用しています: Resolution を **0.410**, しきい値 を **5.0**、そして Stiffnessは少しだけ増えて **5.01**。



左: 基礎構造、右: 形状。

メタキューブの形状のもとになる基本的な立方晶構造をはっきりと見ることができます。

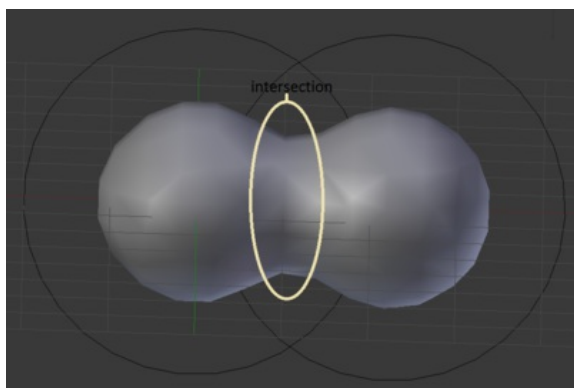
しきい値 (影響)

Mode: Object or Edit modes

Panel: MetaBall (Editing context, F9)

Thresholdは、メタの表面が他のメタにどのくらい“影響を与える”かを定義します。表面が計算される時のフィールドレベルを制御します。設定は、メタオブジェクトの [group](#) に対してグローバルです。しきい値が増加するにつれて、メタ同士がお互いに与える影響も増加します。

影響力には2つの種類があります: 正 と 負 です。タイプの切り替えは、編集モードのActive Elementパネルで、Negativeボタンを使用して行えます。正 はメッシュの引力、負 はメッシュの反発力と考えることができます。負のメタは、正のMetaオブジェクトのメッシュを押し分けたり撥ねつけます。



Positive.

positive の影響を引力と定義すると、影響の輪が交差するところではメッシュが互いに向かって引っ張り合うことになります。(Positive) では 2つのメタボールの 影響の輪 が交差して 正 の影響を及ぼすところを示しています。

メッシュがお互いに向かってどのように引っ張り合っているかに注目しましょう。影響の輪が交差する共有地を、白い丸で囲んだエリアで示しています。

Update

メタを変形するとき (グラブ/移動、スケールなど)、可視化の“モード”は四つあり、MetaBallパネルの Updateボタングループに配置されています:

- Always – 変形中にメタを完全に描画します。
- Half Res – During transformations, draw the meta at half its Wiresize resolution.

- Fast – 変形中にメタメッシュを表示しない。
- Never – メタメッシュを表示しません (レンダリング時にしかメタが表示されないで、全くお勧めできないオプションです!)

これは、複雑な実験のような場合には役立つではありませんが (メタ情報は極めて集中的に計算される...)、しかし現代のコンピュータでは、多くすぎるメタ情報を使ったり、またはあまりにも高い解像度を使用しない限り、これの出番がくることはないはずです...

メタの構造

技術詳細

メタオブジェクトをより正式に定義するなら、静的フィールドのソースとして参照される *方向指示構造* ととらえることができます。フィールドは正または負のいずれかになり、それゆえ近隣の方向指示構造によって引き寄せられたりはね除けられたりしながらフィールドが生成されます。

暗黙的な表面は、3Dフィールドに表面として定義されます。この3Dフィールドは、与えられた値を前提とするすべての指示構造によって生成されます。たとえばメタボールでは、その指示構造は点で、周りに等方性 (すなわちすべての方向に同一の) フィールドを生成し、フィールド値が一定の表面では指示点を中心とする球になります。

メタオブジェクトは、相互に論理演算 (AND, OR) を行う数式でしかなく、互いに加算や減算をすることができます。このメソッドは、**Constructive Solid Geometry** (CSG) と呼ばれています。その数学的性質のため、CSGはほとんどメモリを使用しませんが、計算には多大な処理能力を必要とします。

Underlying Structure

Mode: Edit mode

Panel: MetaBall tools (Editing context, F9), Transform Properties

Blenderには5種類のメタがあり、それぞれその基礎となる (または方向を定める) 構造によって決定されます。編集モードでは、この構造を変更でき、MetaBall tools パネル内の該当するボタンを使用するか、もしくは Transform Properties パネル (N) のドロップダウンリストを使用して行えます。構造によっては追加のパラメータ必要になる場合があります、これは Transform Properties パネルと MetaBall tools パネルの両方に配置されています。

Ball (点、ゼロ次元構造)

これは最もシンプルなメタで、拡張的な設定は全くありません。そこがポイントでもあり、等方性のフィールドを生成し、しなやかな球面が得られます (Blender で Meta Ball または Ball in と呼ばれる由縁がこれです)。

Tube (直線、単次元構造)

このメタが生成する表面は、指定の長さの直線によって生成されるフィールドがもとになります。円筒形の表面になり、端部は丸く閉じます。追加パラメータは1つあります:

- dx: 線の長さ (したがって、チューブの長さ – デフォルトは **1.0**)。

Plane (長方形の平面、二次元構造)

このメタが生成する表面は、長方形の平面によって生成されるフィールドがもとになります。これは平行六面体の表面になり、一定の厚さもち、境界は丸みを帯びます。追加パラメータは2つあります:

- dx: 四角形の長さ (デフォルトでは **1.0**).
- dy: 四角形の幅 (デフォルトでは **1.0**).

デフォルトでは、平面が正方形であることを覚えておきましょう。

Ellipsoid (楕円体の容積物、三次元構造)

このメタが生成する表面は、楕円体ボリュームによって生成されるフィールドがもとになります。これをもとに楕円体の表面ができます。追加パラメータは3つあります:

- dx: 楕円体の長さ (デフォルトでは **1.0**).
- dy: 楕円の幅 (デフォルトでは **1.0**).
- dz: 楕円の高さ (デフォルトでは **1.0**).

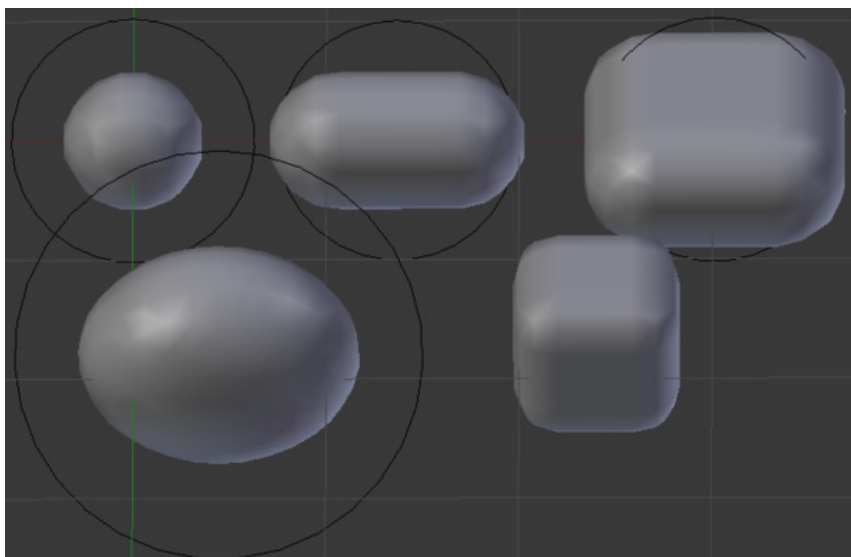
デフォルトでは、ボリュームは球で、Ball オプションと同様の球状のメタが生成されることに注意しましょう...

Cube (平行六面体の容積物、三次元構造)

このメタが生成する表面は、平行六面体のボリュームによって生成されたフィールドがもとになります。平行六面体の表面になり、エッジは丸みを帯びます。ご想像のとおり、3つの追加パラメータがあります:

- dx: 立方体の長さ (デフォルトでは **1.0**).
- dy: 立方体の幅 (デフォルトでは **1.0**).
- dz: 立方体の高さ (デフォルトでは **1.0**).

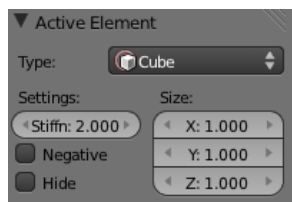
デフォルトでは、ボリュームはキューブであることに注意しましょう。



the 5 meta primitives.

メタの編集

エディットモードのときに、Active Element/パネルが表示されます。これらの設定は、選択状態のmeta要素にのみ適用されます。



the active element panel.

メタの形状


Typeメニューでは、上記のように、メタオブジェクトの形状を変更できます。

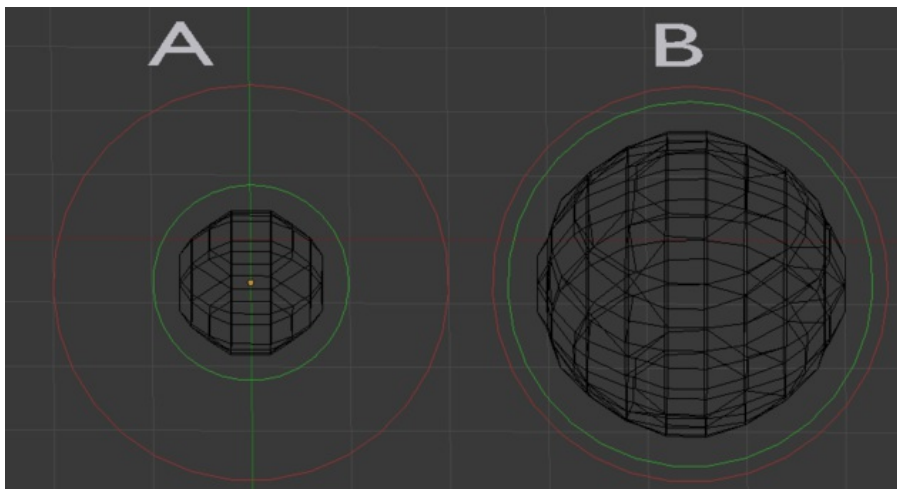
Stiffness

Thresholdと一緒に使用すると、Stiffnessが影響する範囲を制御します。同じオブジェクト内 (あるいは、同じ [オブジェクトファミリー](#)) ではしきい値はすべてのメタで共通ですが、剛性は各メタに固有のものであります。

内側の緑色の丸を拡大縮小すると、Stiffnessの値を変更します。Stiffnessは、メタオブジェクトがどの程度充填されているかを定義します。これは本質的に、メタが他のメタにどの程度敏感に影響されるかを定義します。剛性が低くなるほど、メタの変形が離れた所から開始されるようになります。高い値ほど、メタがマージし始めるまでにもう一歩近づく必要があることを意味します。

Metaオブジェクトが別のメタの“範囲”内に入ると、二つは互いに相互作用を開始します。必ずしも交差する必要はなく、しきい値とStiffnessの設定に依存しますが、設定をいじる必要はほとんどありません。Stiffnessは [緑のリンク](#) でマテリアライズされます。

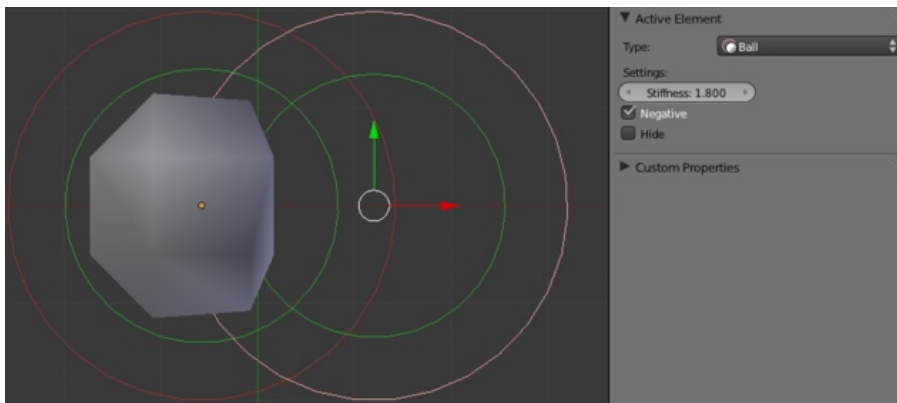
範囲は **0.0** から **10.0** までです。しかし、Stiffnessが見えるためには、しきい値よりもわずかに大きくなければなりません。また、視覚的にStiffnessのリンクを調整することもでき、RMB  を使用してそれを選択し、SでScaleモードを有効にします。



Stiffness.

(Stiffness)図の、“A”とラベルされたメタボールは、“B”とラベルされたものよりも小さいStiffness値を持っています。ご覧のように [緑のリンク](#) の半径が、双方の間で異なっています。

Negative Influence



Negative.

正の影響力の反対の効果は 負の影響力です:オブジェクトがお互いを押し退けます。(Negative) 図ではメタボールとメタ平面をお見せしています。最初は正、ふたつめは負です。負のメタが見えていないことに注目しましょう:周りの円だけが表示されています。オブジェクトが負であることを、

Blenderがこのように表現しているわけです。

球を平面に移動させると、平面のメッシュが“内側に陥没する”か、または内側に崩壊します。平面を移動させて球から離すと、平面のメッシュは自体を復元します。

メタを負にするには、編集モードでメタを選択し、*active element* パネルで *negative* にチェックをいれるだけです。

要素を隠す

Objectモードと同様に、選択しているメタ(s)を非表示にしたり、隠していたものをまた出現させることができます。これはビューを少しすっきりさせたいときに非常に便利です... 赤と緑のリング2つは、選択サークル(オブジェクトモードでの...)と同様に、編集モードでも常に表示されたままでということに注意してください

現在の選択しているものを非表示にするには、H、MetaBall toolsにあるHideトグルボタン、または Metaball » Hide MetaElems » Hide Selected メニューオプションを使いましょう。

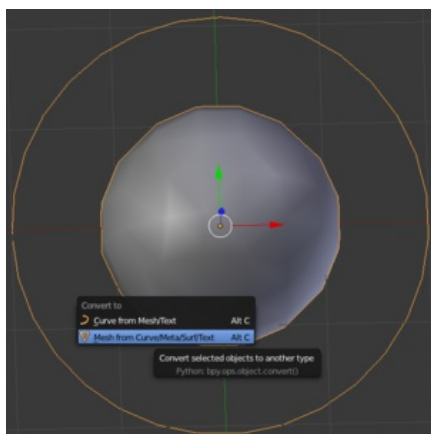
現在選択しているもの以外のすべてを非表示にするには、⇧ ShiftHを押すか、または Metaball » Hide MetaElems » Hide Deselectedメニューを使用しましょう。

隠していたものを再表示するには、AltH、または Metaball » Hide MetaElems メニューを使用します。また、Hideボタンのトグル解除で切り替えることもできます (MetaBall tools panel)。

要素の削除

メタ用の消去メニューというものはなく、選択したメタ情報を削除したい場合は、そのままポップアップを承認するだけです。単純明快!

Conversion



the convert menu

メタからメッシュへの変換だけを行うことができ、しかしこのときには、元のメタオブジェクトを保持するオプションもあります (すなわち、“それ自体”を変換するのではなく、新しくメッシュを作成してから...)。新しいメッシュに使用される resolution は Wiresizeのもので、Render sizeのものではないことに注意してください。

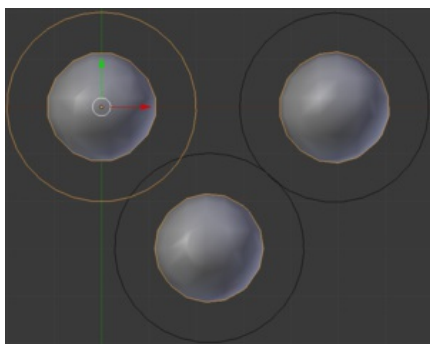
メタを変換するには、オブジェクトモードでAltCを押し、そして *mesh*を選択しましょう

Object Families

ObjectモードでのMetaオブジェクトは、他のオブジェクトタイプとは異なる動作を持っています – それらをいわゆる“ファミリー”に“再編成”することができます。

“ファミリー”とは、複数のメタオブジェクトを再グループ化する方法で、生成するものがあたかも同じオブジェクト内に複数のメタ情報を持っているかのようになります。

ファミリーは、オブジェクト名の左側の部分で定義されます (ドットの前の部分)。ほとんどのパネルで、オブジェクトの名前は唯一“OB”フィールド内のみであり、“MB”フィールドにはないことに注意しましょう。MBは、メタデータブロックの名前です... たとえば、“MetaPlane.001”のファミリーの一部が“MetaPlane”といった具合です。同じ“ファミリー”の各メタオブジェクトは、後述のように、互いに関連付けられます。



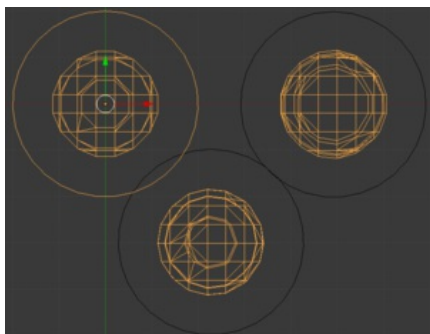
Meta ball base.

メタのファミリーは、ベースメタオブジェクトによって制御されます。これにはオブジェクト名の右の部分を除いた識別名がつきます。たとえば、“MetaThing”, “MetaThing.001”, “MetaThing.002”, “MetaThing.003”, “MetaThing.004”、と呼ばれる5つのメタがある場合、ベースメタオブジェクトは、“MetaThing”です。

ベースメタオブジェクトは、基礎、分割数、しきい値、および変形を決定づけます。また、マテリアルとテクスチャの領域も持っています。ベースメタは、事実上、グループ内の他のメタの親 (あるいは“オーナー”という単語を使用したほうがベターかもしれない) です (つまり、他のメタがベースメタに“含まれる”または加入しているかのよう)。

例

(Meta ball base)図には、“B”とラベルされたベースメタを示しています。他の2つのメタオブジェクトは子です。子の選択リングは常に黒で、グループのメッシュはオレンジ色です。メタはグループ化されているので、統一されたメッシュを形成し、グループ内の任意のメタメッシュを選択することで常に選択されるようになります。たとえば、(Meta ball base)の例で、下側の球 (親) だけを選択すると、親メッシュとすべての子メッシュの両方が強調表示されていることがわかります。



Scaling the “base”.

ベースメタオブジェクトは、グループの **polygonalization** (mesh structure) を制御し、また同様に、子の(非ベースの)メタの polygonalization を制御します。ベースメタを変形すると、子の polygonalization も変化します。しかし、子を変形させても、polygonalization は変わりません。

ヒント

この“polygonization”の議論は、様々なメッシュが互いに向かってまたは離れるほうへ変形しないという意味ではありません (タオブジェクトは、それらが同じファミリーのメンバーであるかどうかにかかわらず、常に通常の方法でお互いに影響を与えます)。むしろ、base オブジェクトが変形される時にだけ、基礎となるメッシュ構造が変化することを意味します。たとえば、ベースを拡大縮小すると、子のメッシュ構造も変化します。(Scaling the “base”)図では、ベースが縮小されていて、この効果で子それぞれのメッシュ構造がスケーリングされています。ご覧のように、子のメッシュの分割数は増加し、ベースのほうは減少しています。子がサイズを変えることはありませんでした!

Empties

(エンティ)

"エンティ" は空のオブジェクトです。実際のジオメトリを持ちませんが多用途のハンドルとして使えます。

Settings

(設定)

Plain Axes (座標軸)

それぞれが +X、-X、+Y、-Y、+Z、-Z軸を向く6つのラインとして描画します。

Arrows (十字)

X軸、Y軸、Z軸、の正方向を指す、それぞれにラベルの付いた矢印として描画します。

Single Arrow (矢印)

+Z軸を指す単一の矢印として描画します。

Circle (円)

XZ平面の円として描画します。

Cube (立方体)

XYZ軸に平行な立方体のフレームとして描画します。

Sphere (球)

3つの円によって定義された暗黙の球体として描画します。初めは各円は X、Y、Z軸に平行です。

Cone (円錐)

+Y軸を指す円錐として描画します。

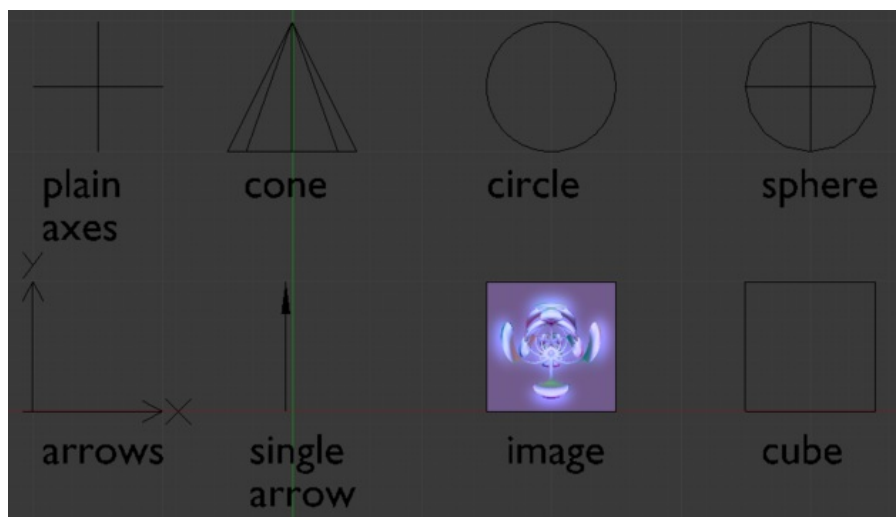
Image (画像)

エンティに画像を表示させることができます。青写真またはモデリングの元になるキャラクターシートを含む、参考用の画像の生成に、背景画像のかわりに使用することができます。画像は3D表示モードにかかわらず表示されます。設定は [背景画像の設定](#) と同じです。

"備考: アルファ画像を使えますがオブジェクト描画順に既知の制限があり、非選択のとき他オブジェクトの上にアルファが描画されないことがあります。"

Size

エンティのローカルサイズを制御します。これは、自身のスケールを変更するのではなく、単に形状のサイズを変更するだけです。



エンティの8種の描画種類

Usage and functions

(用途と機能)

変換のハンドルとして機能し、編集やレンダリングはされません。エンティは、重要で便利なオブジェクトです。使用方法のいくつかの例は、次のとおりです:

オブジェクトグループの親オブジェクト

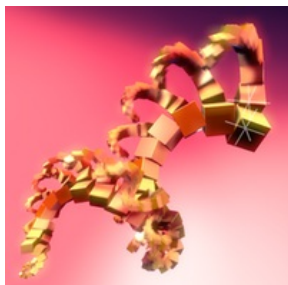
- 任意の数の他のオブジェクトの親にできます。レンダリングに影響させずにオブジェクトグループの制御が簡単に行えます。

拘束のターゲット

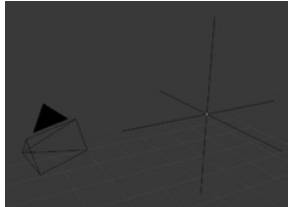
- エンティは通常の拘束かボーン拘束のターゲットとしても使えます。
- これを利用すると制御できることがはるかに増えます。たとえば **Track to** 拘束を使ってカメラをエンティに向かせる簡単なリグを作れます。

配列のオフセット

- エンティは Array (配列複製) モディファイアの補正に使えます。複雑な変形が単独のオブジェクトを動かすだけでできるようになります。



配列操作に使われるエンプティの例



Track to 拘束の制御に使われるエンプティの例

その他の一般的な用途

- プレースホルダ
- リギングの制御
- 被写界深度 (DOF) の距離
- 画像の参照

Group Instances

(グループインスタンス)

グループは [グループ化と親子関係](#) ですべて解説されています。

シーンにグループがすでにあれば、Add (追加) » Group Instance (グループのインスタンス) » Group (グループ) の追加メニューからグループのインスタンスを作ることができます。

このグループプロキシ(代行、proxy)は補助的なエンティオブジェクトで制御/所有され、したがって編集できません(編集モードを持ちません)。実際のところグループのオブジェクトのコピーにはすべて、エンティの子であるかのような動作が見られます。したがって、オブジェクトのすべての編集オプションを使えます(移動/回転/拡大縮小など)。

グループとメタ

グループプロキシ内のメタオブジェクトにはバグがあるようです。メタのメッシュは見えません。円だけが描かれます...

Visualization

(視覚化)

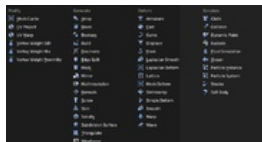
グループプロキシは黒で描かれます(もちろんソリッドやテクスチャ 描画モードでない限り)。グループの「本当の」メンバーの選択内容はその「プロキシ」に反映されます。「プロキシ」自身が選択されるとエンティはピンクに変わり、それ以外は紫になります。

グループプロキシはオプションを1つだけ持ちます。これは [エンティの視覚化](#) のオプションと同じものです。

Modifiers

Mode: Any mode

Panel: Modifiers



Modifiers menu

モディファイアとは非破壊的にオブジェクトに作用する自動処理です。モディファイアを使うと、手作業で行うには面倒な効果(たとえばsubdivision surfaces)を自動的に施してくれます。しかも、オブジェクトそのものを元から変更してしまふことはありません。モディファイアはオブジェクトがどう表示されるか、どうレンダリングされるかということを変えるものですが、実際にオブジェクトの構造を変えるわけではないのです。モディファイアは1つのオブジェクトにいくつも付加して**モディファイアスタック**とすることもできますし、モディファイアの効果を永久的に定着させたい場合はApply(適用)することもできます。

モディファイアには4つのタイプがあります:

修飾 (Modify)

Modifyに属するモディファイアはTransformツールに少し似ています。しかしオブジェクトの形状に直接影響を与えるというよりも、頂点グループなどのデータに対して作用するものです。

[UV投影 \(UV Project\)](#)

メッシュにUV座標を投影します。

[頂点ウェイト \(Vertex Weight\)](#)

様々な方法で頂点グループに編集を加えます。

生成 (Generate)

Generateに属するモディファイアはオブジェクト全体の外観を変化させたり新たな構造を付加したりする構築的なツールです。

[配列複製 \(Array\)](#)

元のメッシュを反復した形状を生み出します。

[ベベル \(Bevel\)](#)

メッシュオブジェクトに対してベベルを付けます。

[ブーリアン \(Boolean\)](#)

他のメッシュとの結合部分、差分、共通部分となる形状を生み出します。

[ビルドアニメ \(Build\)](#)

メッシュが少しずつできあがる様子をアニメーションにします。

[ポリゴン数削減 \(Decimate\)](#)

メッシュのポリゴン数を削減します。

[エッジ分離 \(Edge Split\)](#)

メッシュの辺をシャープにします。

[マスク \(Mask\)](#)

メッシュの特定の部分を隠します。

[ミラー \(Mirror\)](#)

オブジェクトの軸に沿ってメッシュを鏡面コピーします。その結果メッシュはシンメトリカルな形状となり、しかも形を変えるために編集するのは全体の半分、1/4、または1/8で済みます。

[多重解像度 \(Multiresolution\)](#)

数種類の再分割レベルを使い分けてメッシュを制作できます。

[スクリュー \(Screw\)](#)

単純な断面図形状から、らせん形状を生み出します。メッシュ編集のScrewlに似ています。

[ソリッド化 \(Solidify\)](#)

メッシュの面に厚みを付加します。

[再分割 \(Subdivision Surface\)](#)

メッシュ形状を補間して表面を滑らかにします。

[Skin](#)

自動的に生成される構造です。

変形 (Deform)

Deformに属するモディファイアはオブジェクトの形状をゆがめて変形させます。これはメッシュはもちろん、多くの場合テキスト、カーブ、サーフィス、ラティスに対しても使えます。

[アーマチュア \(Armature\)](#)

ボーンを使ってオブジェクトを変形、アニメーションさせます。

[金型 \(Cast\)](#)

メッシュ、サーフィス、ラティスを、球体、円柱または立方体に近づくように変形します。

[カーブ \(Curve\)](#)

カーブオブジェクトをガイドとしてオブジェクトを曲げます。

[置換 \(Displace\)](#)

テクスチャに基づいてオブジェクトを変形します。

[フック \(Hook\)](#)

頂点や制御点を外部から操作するためのフックを付加します。

[格子 \(Lattice\)](#)

ラティスオブジェクトを使ってオブジェクトを変形します。

[メッシュ変形 \(Mesh Deform\)](#)

他のオブジェクトを(ラティスのような)「メッシュを変形のためのケージ」として使ってオブジェクトを変形します。

[収縮包装 \(Shrinkwrap\)](#)

オブジェクトで、ターゲットとなるメッシュオブジェクトをびったり包みます。

[Simple Deform](#)

いくつかの高度な変形を施します。

[平滑化 \(Smooth\)](#)

メッシュを滑らかにします。メッシュ編集のSmoothツールに似ています。

[湾曲 \(Warp\)](#)

指定した2点の間でオブジェクトを引き伸ばしてゆがめます。

[波 \(Wave\)](#)

オブジェクトを波打たせます。

シミュレート (Simulate)

Simulateに属するモディファイアはシミュレーションを起動します。多くの場合、これらのモディファイアはParticle SystemまたはPhysicsが有効になったときに自動的にモディファイアスタックに追加され、モディファイアパネルの役目はモディファイアスタックの中での位置(順序)を規定することだけです。その情報がシミュレーションに使われます。基本的にこれらのシミュレーションの設定は別のパネルで行います。

[クロス \(Cloth\)](#)

布の性質をシミュレーションします。これはメッシュをクロスとして設定したときにモディファイアスタックに追加されます。

[衝突 \(Collision\)](#)

オブジェクト同士の衝突をシミュレートします。

[爆発 \(Explode\)](#)

パーティクルシステムを使ってメッシュを爆破します。

[流体 \(Fluid\)](#)

オブジェクトを流体シミュレーションの一部にします。メッシュの流体の設定をしたときにモディファイアスタックに追加されます。

[パーティクルインスタンス \(Particle Instance\)](#)

オブジェクトがパーティクルのように振舞いますが、それをメッシュ形状を使って行います。

[パーティクルシステム \(Particle System\)](#)

パーティクルシステムを表すモディファイアです。オブジェクトにパーティクルシステムを追加したときにモディファイアスタックに挿入されます。

[煙 \(Smoke\)](#)

リアルな煙をシミュレートします。

[柔体 \(Soft Body\)](#)

オブジェクトが柔らかく、弾力を持ちます。メッシュをソフトボディに設定するとモディファイアが追加されます。

[ダイナミックペイント \(Dynamic Paint\)](#)

オブジェクトやパーティクルが他のオブジェクトにマテリアルを塗りつけるようになります。

[海 \(Ocean\)](#)

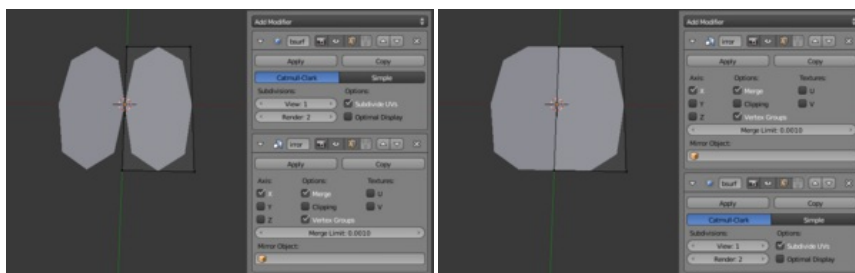
簡単にリアルでアニメートする海を作り出せます。

Modifier

(モディファイア)

モディファイアとは、ある「工程」あるいは「アルゴリズム」のオブジェクトへの適用方法を定義したものです。インタラクティブで非破壊的に、どんな順序でも利用できます。この種の機能はよく「モディファイアスタック」と呼ばれ、他の 3D アプリケーションでも見られるものです。

モディファイアは Modifiers (モディファイア) メニューから追加します。一部のツールやスクリプト、例えば Decimate と Solidify は以前の位置から移されてモディファイアに変わりました。モディファイアスタックにモディファイアを適用する順序は結果に影響を及ぼします。幸いにもモディファイアは、使いやすい上下の矢印アイコンをクリックして簡単に並べ替えられます。例えば図「スタック順序」では [SubSurf](#) と [Mirror](#) モディファイアを入れ替えています。



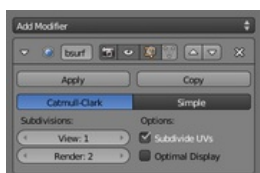
スタック順序

最初の例では Mirror モディファイアがスタックの最後にあります。結果は二つの面のように見えます。もうひとつの例ではこのモディファイアはスタックの最初にあり、結果はひとつにつながった面になっています。

結果が前と変わったことがわかります。スタックの順序は最終的な結果を定義する上で非常に重要だということです。

Interface

(インタフェース)



パネル配置 (Subsurf の例)

各モディファイアは Blender のさまざまな部分から集められたものなので、それぞれに独自の設定や個別の考慮事項があります。ただしインターフェイスの基本部品は共通しています (図「パネル配置 (Subsurf の例)」)。

一番上にあるのが パネルヘッダー です。各アイコンはモディファイアのさまざまな設定を表します (左から右へ):

1. 矢印 — モディファイアを折りたたみ、ヘッダのみを表示します。
2. モディファイアのアイコンとこのモディファイアの名前のボックス — デフォルトではモディファイア自体の名前になります。同じタイプの他のモディファイアとの間で一意になっています。
3. カメラ — レンダリング時にモディファイアの効果を表示します。
4. 目 — 3Dビューでモディファイアの効果を表示します。
5. 箱 — 編集モードでモディファイアの影響が見えるようになります。このボタンはモディファイアの種類によっては使用できない場合があります。
6. 三角形 — 編集モードでケージを編集できるようにします。このアイコンは ケージ モードを表示させます。
7. 上矢印 — スタック内でモディファイアを上へ移動します。
8. 下矢印 — スタック内でモディファイアを下へ移動します。
9. クロス — スタックからモディファイアを削除します。

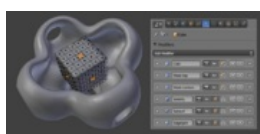
ヘッダの下に2つのボタンがあります:

1. Apply (適用) — モディファイアを実体化させます。
2. Copy (コピー) — スタックの一番下にモディファイアのコピーを作成します。

そしてこれらのボタンから下は、モディファイア個別の設定用サブパネルです。

Stack

(スタック)



この例では細分化された単純な立方体が、モディファイアスタックを使ってより複雑なオブジェクトの内部にトランスフォームされています。[\(blend\)](#)

モディファイアを追加すると、それを スタックに追加することになります。一度追加すれば(常にスタックの最下部に追加されます)、好きなように並べ替えられます。

特定のオブジェクトタイプにしか適用できないモディファイアもあります。これは モディファイア パネルにある Add Modifier(追加) ボタンを選別するパネルでわかります。適用できるモディファイアだけが、このリストボックスのボタンに表示されます。メッシュ オブジェクトでは利用可能なモディファイアはすべて追加できます。一方で例えば Lattice(ラティス) オブジェクトでは、一部のみを持つことができます。

UV投影 (UV Project) モディファイア

Mode: Any mode

Panel: Modifiers (Generate)

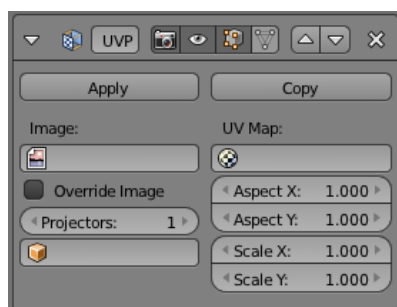


Projecting the Blender logo onto Suzanne.

UV投影はスライド映写のようなものです。モディファイアを適用したオブジェクトに対して、最大10個のオブジェクトの-Z方向からUVマップを放射できます。そして、映写機の光が当たったかのようにオブジェクトの表面に座標を適用します。オプションとしてオブジェクトのテクスチャも上書きすることができます。

サンプルファイル

オプション



UV layer

変更されるUVレイヤー。デフォルトではアクティブなレイヤーが使われます。

Image

モディファイアで使うイメージ。必須ではありません。UV座標自体のみを投影することもできます。このイメージをどう使うかはOverride Imageで決まります。

Override Image

- 有効にすると、一つ一つの面がそれぞれイメージに置き換えられ、イメージは繰り返されます。それはふつうは好ましくありません。
- 無効にすると、指定したイメージが存在する面だけに効果が限定されます。

Projectors

<Objects>

最大10個の投影オブジェクトまでサポートされています。それぞれの面は、その法線になるべく沿っているプロジェクターを選んで投影を受けます。投影は-Z軸の方向になされます(つまりカメラやランプを真下に向けた状態)。投影機がカメラの場合は、遠近法カメラか平行カメラかによって投影も変わります。

Aspect X/Y

Scale X/Y

簡単なイメージの操作です。

使い道

UV Projectはスポットライトをより多様にするのに最適です。また、デカールの繰り返しを防ぐのにも使えます。

Imageのプロパティはいつも使われるようなものではありません。かわりに、モディファイアを適用するオブジェクトのマテリアルとして割り当てられているテクスチャを使います。Texture > Image Mapping > Extension to Clipを設定するとイメージが繰り返されないようにできます。

頂点ウェイト (WeightVGroup) モディファイア

Mode: Any mode

Panel: Modifiers (Modifiers properties)

解説

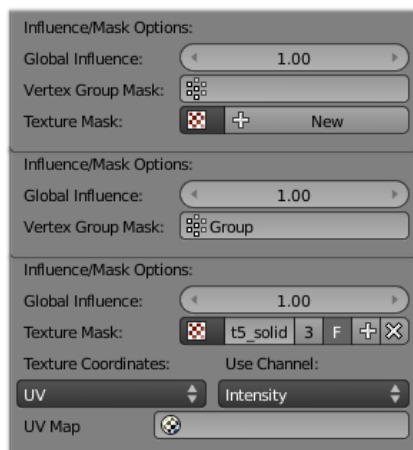
頂点ウェイトモディファイアはオブジェクトの頂点グループに対して働き、そのウェイト、および／または、どの頂点がグループに属するかを変化させます。

これらのモディファイアはウェイトを $[0.0, 1.0]$ の範囲に収めます。よってこの範囲外の数値は失われます。

現在3種類のモディファイアがあります。

- [頂点ウェイト編集 \(Vertex Weight Edit\)](#)
- [頂点ウェイト合成 \(Vertex Weight Mix\)](#)
- [頂点ウェイト近接 \(Vertex Weight Proximity\)](#)

共通設定



The influence/masking part of WeightVGroup modifiers.

いくつかの設定は3種のモディファイアで共通しています。モディファイアの影響の度合いをコントロールする項目です。

Global Influence

全体的なモディファイアの影響力 (**0.0**ならウェイトは変更されず、**1.0**なら標準的な影響力になります)。



これはウェイトだけにかかわる数値であることに注意してください。0.0でもグループに対する頂点の追加／除去は行われます。

Vertex Group Mask

補助的な頂点グループ。このグループのそれぞれの頂点のウェイトがGlobal Influenceの値に掛け合わされます。このグループに属さない頂点に関しては、マスキングウェイト(つまりモディファイアの影響)はゼロになります。

Texture

補助的なテクスチャ。個々の頂点におけるテクスチャの値がGlobal Influenceに掛け合わされます。標準のテクスチャIDで操作します。セットされるとさらに設定項目が現れます。

Texture Coordinates

テクスチャのマッピング法。4種類あります。

- Local ローカル座標系を使います。
- Global グローバル座標系を使います。
- Object 指定したオブジェクトの座標系を使います。
- UV UVレイヤーの座標系を使います。

Use Channel

テクスチャのどのチャンネルを使うかを指定します(強度、RGB、HSV、アルファ - これらの選択肢の意味はおそらく見ればわかるでしょう)。

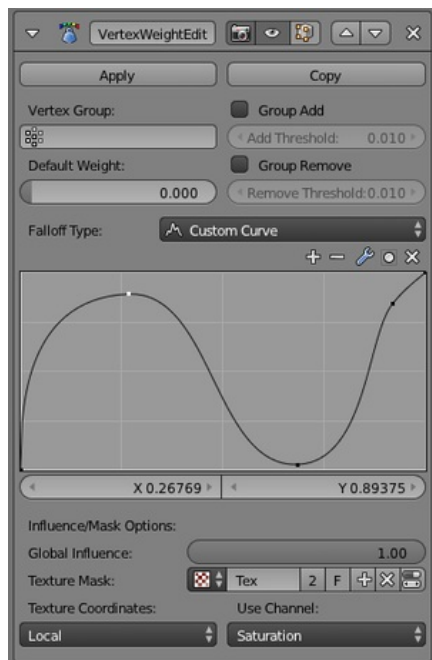
Object

Objectマッピングに使うオブジェクト。

UV Layer

UVマッピングに使うレイヤー。

頂点ウェイト編集 (Vertex Weight Edit) モディファイア



The WeightVGEEdit modifier panel.

これはひとつの頂点グループのウェイトを編集するためのモディファイアです。

個々の頂点に対して以下の操作がなされます。

1. [任意]用意された関数またはカスタムカーブに基づいてウェイト値のマッピング(関数の計算)をします。
2. 計算された値、そして補助頂点グループおよびテクスチャに基づいて影響度を変化させます(0なら元のウェイトのまま、1.0なら完全に置き換えられます)。
3. 計算されたウェイトが頂点に適用されます。そして/または、しきい値に応じて頂点がグループに追加/除去されます。

オプション

Vertex Group

影響を受ける頂点グループ。

Default Weight

グループに属さない頂点に与えられるデフォルトのウェイト。

Falloff Type

マッピングのタイプ:

- Linear 再マッピングせず線形的にマッピングされます。
- Custom Curve カーブマッピングを有効にします。カーブコントロールが現れます。
- Sharp, Smooth, Root, Sphere 一般的なマッピング関数です。
- Random 完全にランダムにマッピングします。
- Median Step 0.5を境として0.0か1.0に振り分けます。

Group Add

最終的なウェイト値がAdd Thresholdよりも高くなった頂点はグループに追加されます。

Group Remove

最終的なウェイト値がRem Thresholdよりも低くなった頂点はグループから除去されます。

頂点ウェイト合成 (Vertex Weight Mix) モディファイア



The WeightVGMix modifier panel.

このモディファイアは様々な方法で第一の頂点グループに第二の頂点グループのウェイト(または単純な値)を合成します。

また、どの頂点に影響を与えるかもコントロールできます(すべて、第一/第二のグループに属するもの、など)。



このモディファイアでは頂点がグループに追加されることがあります(除去されることはありません)。下記参照。

オプション

Vertex Group A

影響を受ける頂点グループ。

Default Weight A

グループに属さない頂点に与えられるデフォルトのウェイト。

Vertex Group B

合成に使われる頂点グループ。

Default Weight B

グループに属さない頂点に与えられるデフォルトのウェイト。

Mix Mode

ウェイトを合成する方法。7種類あります。

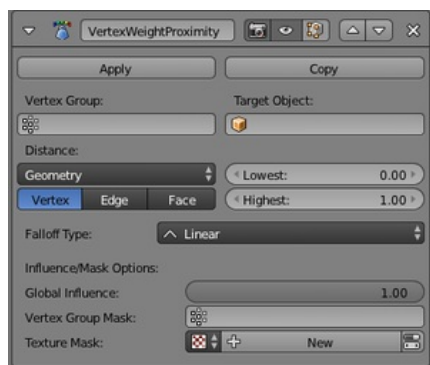
- Replace weights 第二グループのウェイトで置き換えます。
- Add to weights ウェイトを足し合わせます。
- Subtract from weights 第一グループのウェイトから第二グループのウェイトを引き算します。
- Multiply weights ウェイトを掛け算します。
- Divide weights 第一グループのウェイトを第二グループのウェイトで割り算します。
- Difference 差の絶対値を計算します。
- Average ウェイトを平均します。

Mix Set

どの頂点に結果を適用するかを決めます。5種類あります。

- All vertices すべての頂点に結果を適用します。結果としてグループに頂点が追加されることがあります。
- Vertices from group A 第一グループに属する頂点だけに影響を与えます。
- Vertices from group B 第二グループに属する頂点だけに影響を与えます。結果としてグループに頂点が追加されることがあります。
- Vertices from one group 少なくとも一方のグループに属する頂点に影響を与えます。結果としてグループに頂点が追加されることがあります。
- Vertices from both groups 両方のグループに属する頂点だけに影響を与えます。

頂点ウェイト近接 (Vertex Weight Proximity) モディファイア



The WeightVGProximity modifier panel.

このモディファイアは、オブジェクト(またはその頂点)と他のターゲットオブジェクト(またはそのジオメトリ)の距離に応じて頂点グループのウェイトを変化させます。

オプション

Vertex Group

ウェイトを変化させる頂点グループ。

Target Object

このオブジェクトまでの距離にもとづいて処理をします。

Proximity mode

- Object Distance オブジェクト同士の距離に基づいて計算したウェイトを、頂点グループのすべての頂点に割り当てます。
- Geometry Distance グループに属する個々の頂点と、ターゲットオブジェクトまたはそのジオメトリとの距離に基づいてウェイトを決めます。

Geometry Distanceモードはさらに三つのオプションを持っています。これらを有効にするとターゲットオブジェクトの中心点ではなくジオメトリ(頂点、辺、面)が使われます(三つとも有効にした場合は最も短い距離が使われます)。ターゲットオブジェクトがジオメトリを持たない場合(空のオブジェクト、またはカメラなど)は、自動的にデフォルトのObject Distanceに戻ります。

Vertex

個々の頂点のウェイトを、ターゲットオブジェクトの最も近い頂点との距離に基づいて決めます。

Edge

個々の頂点のウェイトを、ターゲットオブジェクトの最も近い辺との距離に基づいて決めます。

Face

個々の頂点のウェイトを、ターゲットオブジェクトの最も近い面との距離に基づいて決めます。

Lowest Dist

ウェイトが**0.0**になる距離。この距離をHighest Distよりも上にする、効果が逆転します。

Highest Dist

ウェイトが**1.0**になる距離。この距離をLowest Distよりも下にする、効果が逆転します。

Falloff Type

マッピング関数。前節の[頂点ウェイト編集](#)を参照。

例

ターゲットオブジェクトからの距離を使う

最初の例として、Waveモディファイアを動的にコントロールしてみましょう。

十分な数の頂点(たとえば**100×100**ぐらい)を持ったGridメッシュを作成し、**10 BU**(BU: Blender単位)四方のサイズにします。Edit Modeに入り、Object DataプロパティのVertex Groupsパネルで頂点グループを追加します。そしてすべての頂点を(たとえばウェイト**1.0**)で登録し、Object Modeに戻ります。

ModifiersプロパティでVertex Weight Proximityモディファイアを追加します。モードはObject Distanceです。先ほど作った頂点グループを指定、またターゲットオブジェクト(私はランブにしました)も指定します。

おそらく、Vertex Weight Proximityモディファイアによる線形マッピングを調整しなければならないでしょう。Lowest DistとHighest Distを、それぞれ最も低いウェイトにしたい頂点までの距離と、最も高いウェイトにしたい頂点までの距離にあわせて設定します。

それではWaveモディファイアを追加しましょう。好きなようにセッティングして、コントロールには同じ頂点グループを指定します。

ターゲットオブジェクトをアニメートし、グリッドの上方を移動させます。すると、オブジェクトのまわりにだけ波が現れるのがわかるでしょう! Vertex Weight EditモディファイアをWaveモディファイアの前に追加し、Custom Curveマッピングでより大きい/狭い「影響力の勾配」にすることもできます。

[\[video link\]](#)

[サンプルファイル](#), TEST_1 scene.

ターゲットオブジェクトのジオメトリからの距離を使う

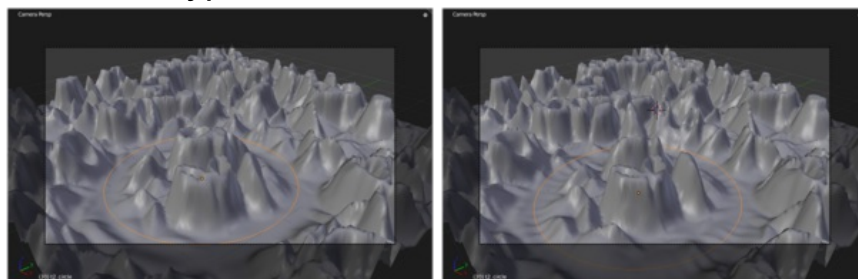
これを、Displaceを使って解説しましょう。

10×10 BUで**100×100**頂点のグリッドを作成し、Edit Modeですべての頂点を頂点グループに登録します。前節と同じ手順です。一つ目のモディファイアとしてSubsurfでさらに再分割してもいいでしょう。

そしてグリッドの上方**0.25 BU**の位置に円形のカーブ形状を追加し、すこし(**4.0**ぐらいに)拡大します。

グリッドを再度選択し、Vertex Weight Proximityモディファイアを追加します。Geometry DistanceモードにしてEdgeを有効にします(Vertexのみを有効にして、かつカーブのU解像度が低いと、波打ったパターンが現れます。図(*Wavy patterns*)参照)。

Wavy patterns.



Distance from edges.

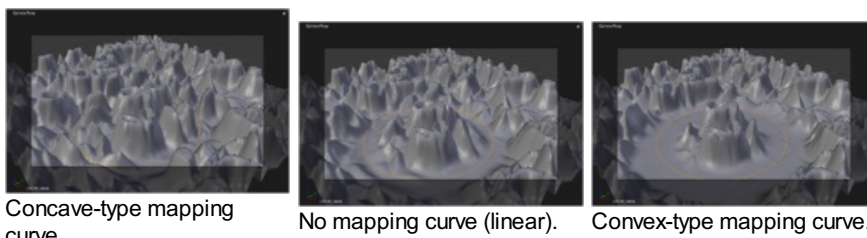
Distance from vertices.

ウェイトのマッピング範囲は、Lowest Distを**0.2**、Highest Distを**2.0**にします。

Displaceモディファイアを三つ目に追加し、好きなテクスチャを指定します。さて、われわれはグリッドの中で円カーブに距離に近い頂点は置換されないようにしたいわけです。その部分ではウェイトがゼロになるので、置換のMidlevelを**0.0**にします。そしてグリッドの頂点グループを指定しておしまいです。グリッド上の山々は円の近くだけ平らにへこむようになるでしょう。

前節の例と同じく、Vertex Weight EditモディファイアをDisplaceの前に入れて、Custom Curveマッピングで影響の与え方を変えることもできます。

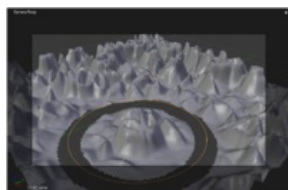
Curve Map variations.



Concave-type mapping curve.

No mapping curve (linear).

Convex-type mapping curve.

Vertices with a computed weight below **0.1** removed from the vertex group.

さらにMaskモディファイアを最後に追加し、Vertex Weight EditのgGroup Removeオプションを有効、しきい値は**0.1**にすると、谷底の部分が消滅します。

[\[video link\]](#)

[サンプルファイル](#), TEST_2 scene.

テクスチャとマッピングカーブを使う

今度は奇妙でエイリアン的な波を作ってみましょう(またWaveモディファイアを使った例ですが、今回は派手ですよ。頂点グループに対する影響がよく分かるでしょう)。

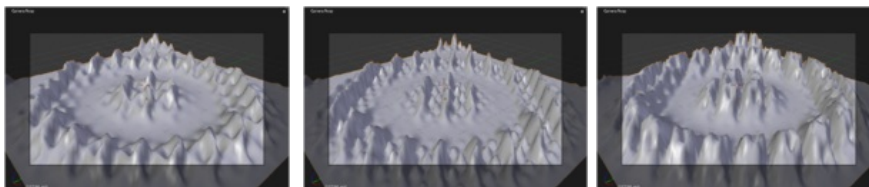
これまで同様**100×100**のグリッドを作ります。今回は、頂点グループ自体は作りますが頂点の登録はしません。それは動的に行うことにします。

そしてVertex Weight Mixモディファイアを追加し、Default Weight Aを**0.0**、Default Weight Bを**1.0**、Mix ModeはReplace weightsのまま、Mix SetはAll verticesにします。これですべての頂点が影響を受けます。初期状態ではどの頂点もグループに属していないのでウェイトは**0.0**で、影響を受けると**1.0**に置き換えられ、グループに追加されることになります。

それではマスキングテクスチャを作りましょう。私はデフォルトのMagicにしました。このテクスチャは、第二グループのウェイト(**1.0**)が第一グループのウェイト(**0.0**)にどの程度影響するかをコントロールします。つまり今回の場合、このテクスチャの値がそのままウェイトになるわけです。

テクスチャのマッピング座標系はデフォルトのLocalのままにして、あとはいろんなチャンネルと好きなようにいじってください。

Texture channel variations.



Using intensity.

Using Red.

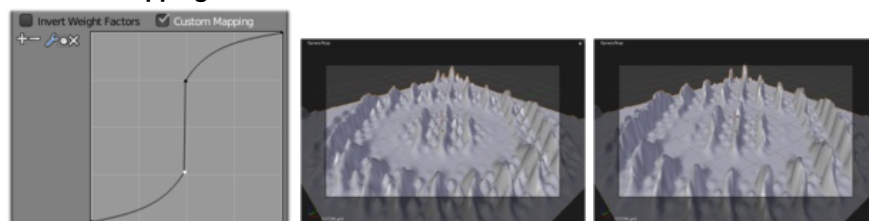
Using Saturation.

それからWaveモディファイアを追加するのを忘れてはいけません。頂点グループも指定してください。

このようにして、生成されたウェイト値をそのまま波に使うこともできますが、カーブマッピングでウェイトを変えたければ、おなじみのVertex Weight Editモディファイアを追加してCustom Curveマッピングを選択しましょう。

デフォルトでは一対一の線形マッピングが行われます。つまり、何もしないのと同じです！ 図(A customized mapping curve)のようにカーブを変更してみてください。[0.0, 0.5]を[0.0, 0.25]に、[0.5, 1.0]を[0.75, 1.0]にマッピングしてあります。これですべての頂点のウェイトが**0.25**以下か**0.75**以上になり、波の中に「壁」が作られます。

Custom mapping curve.



A customized mapping curve.

Custom Mapping disabled.

Custom Mapping enabled.

[\[video link\]](#)

[サンプルファイル](#), TEST_4 scene.

関連項目

- The [Development page](#).

Buildモディファイア

Mode: Object mode

Panel: Modifiers

解説

Buildモディファイアは時間が進むごとにメッシュオブジェクトの面が一つずつ現れるようにします。マテリアルが通常のものではなくhaloの場合は、面ではなく頂点が次々に現れます。

デフォルトでは、面(または頂点)はメモリ内に記録されている順に現れます。メモリ内の順序はデフォルトでは作られた順ですが、Edit modeでSort Faces (CtrlAltF)を使うと変えることができます。

オプション



Build modifier

Start

Buildプロセスの始まるフレーム。

Length

Buildが完了するまでにかかるフレーム数。

Randomize

面が現れる順番をランダム化します。

Seed

ランダム化(乱数)の種。この値を変えると、別の「ランダムな順序」になります。逆に、同じオブジェクトと同じ種を使うと、面の現れる順序はいつも同じです。

Decimateモディファイア

Mode: Object mode

Panel: Modifiers

解説

Decimateモディファイアを使うと、形状の変化を最小限にして頂点や面を減らせます。これは少ない頂点で慎重に作られたメッシュに対して使うのは適切ではありません。そのようなメッシュではすべての頂点や面が形状を保つのに必要だからです。しかし、複雑な過程で作られたメッシュ、たとえばproportional editingや度重なる改良、あるいはSubSurfされた形状からの変換などを経たメッシュの場合、おそらく多くの頂点は本当に必要ではないでしょう。

Decimateモディファイアは非破壊的にポリゴン数を減らすための便利な方法です。これは、モディファイアというシステムの利点を実証しています。通常は元のメッシュデータを永久に変更してしまうような処理が、モディファイアを使えばインタラクティブかつ安全に行えるということを見せてくれます。

他のモディファイアの多くとは違い、Decimate モディファイアの結果はEdit modeで表示させることはできません。

Decimateモディファイアが扱えるのは三角ポリゴンのみです。したがって四角ポリゴンは暗黙のうちに二つに分割されて処理されます。

オプション



decimate modifier

Ratio

Decimateしたあとの面の割合。数値は**0.0**(0%。つまりすべての面が削除されます)から**1.0**(100%。メッシュは元のまま、ただし四角ポリゴンは三角ポリゴンに分割されます)までです。割合を**1.0**から**0.0**へ下げるにしたがって、メッシュはより削減されていき、やがて元のメッシュとは似つかないものになるでしょう。

Face Count (表示のみ)

この欄には削減後の面の数が表示されます。

例

単純な平面

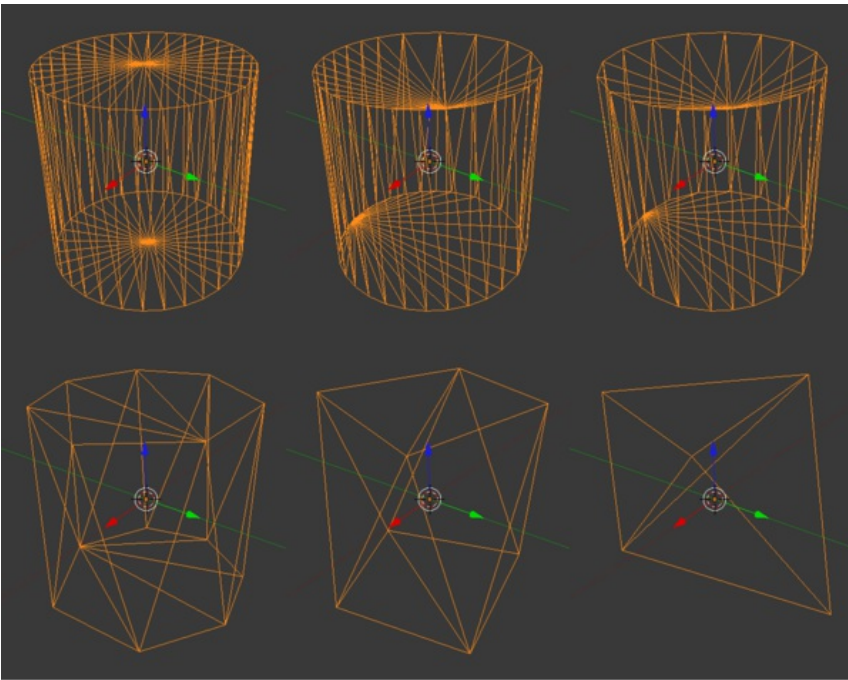
Planeオブジェクトと、何も変形していない4x4のGridオブジェクトを比べてみましょう。どちらもレンダリングの結果はまったく同じです。しかしPlaneは1つの面と4つの頂点を持つのにに対して、Gridは9つの面と16の頂点を持ちます。よって不要な頂点や面が多いわけです。Decimateモディファイアはこのような不要な面を削除してくれます。

Decimateされた円柱

簡単な例として、デフォルトの**32**セグメントの円柱を使ってみましょう。これは最初**96**個の面を持っています。そこにDecimateモディファイアを適用してみると、面の数が増えます！ これはこのモディファイアが四角ポリゴンをすべて三角ポリゴンに変換するためで、そうすると面の数は増えることになります。四角ポリゴンは2つの三角ポリゴンに分解されるからです。

Decimateモディファイアの主な目的は、頂点や面を削減してメッシュのリソースを減らすことですが、同時に元の形状を維持しなければいけません。

以下の画像では、面の割合が**100%**から**5%**(Ratioが**0.05**)へとだんだん下がっていきます。注目してほしいのはRatioが**0.6**(**60%**)に下がると面の数が**128**から**88**になり、しかし依然として元の円柱によく似ていることです。これで不要な面を**40**個削除したことになります。



1.0 (100%). Faces: 128; 0.8 (80%). Faces: 102; 0.6 (60%). Faces: 88
0.2 (20%). Faces: 24; 0.1 (10%). Faces: 12; 0.05 (5%). Faces: 6

Ratioが**0.1**になると、円柱は立方体に近くなるのがわかるでしょう。そして**0.05**になるともう立方体にも見えません！

狙い通りの面の数や見た目になったら、モディファイアをApplyすることもできます。三角ポリゴンを四角ポリゴンに変換してよりリソースを削減したければ、すべての頂点を選択し(A)、それからAltJを押してください。

EdgeSplitモディファイア

Mode: Any mode

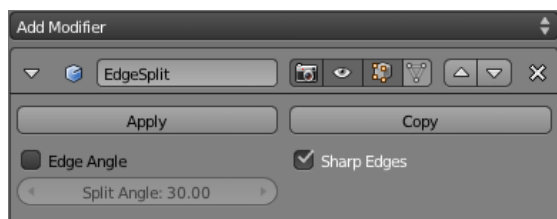
Panel: Modifiers (Editing context, F9)

解説

EdgeSplitモディファイアはメッシュの辺を分離します。辺は角度(つまりその辺で接している2つの面の角度)によって指定することも、辺にシャープにするマークをつけて指定することもできます。

辺を分離すると、その辺上にある頂点の法線の計算に影響し、辺がくっきりとシャープになります。よってこのモディファイアはAutosmoothボタンと同じように使うこともできます。Autosmoothは指定した角度以上に角度の付いた辺をシャープにします。また、EdgeSplitモディファイアではどの辺をシャープに、どの辺を滑らかにするかを手動でコントロールすることもできます(さらに別の方法については [Mesh Smoothing](#) を参照)。角度による指定と手動の指定を両方有効にすることもできます。EdgeSplitモディファイアで生成された結果はスクリプトとして出力できます。これはゲーム製作者にとって便利でしょう。

オプション



EdgeSplit modifier.

From Edge Angle

このボタンを有効にすると、辺の角度がSplit Angleで指定した値よりも大きい場合は辺が分離されます。

- 辺の角度とは、その辺で接している2つの面の角度のことです。
- 2つ以上の面が辺を共有している場合は、その辺は常に分離されません。
- 辺を共有する面が2つより少ない場合は、その辺は決して分離されません。

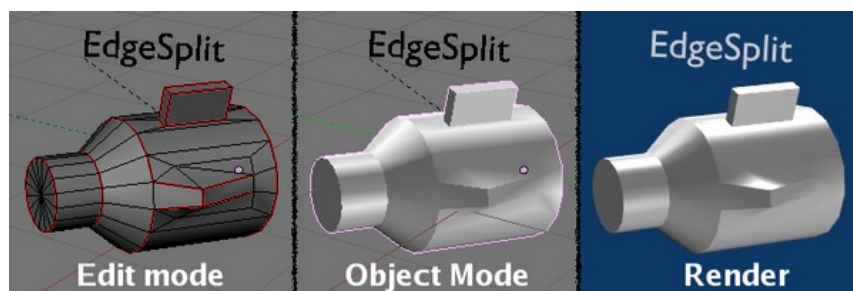
Split Angle

この値よりも辺の角度が大きいと、From Edge Angleが有効の場合に辺が分離されます。数値は0°(すべての辺が分離される)から180°(どの辺も分離されない)です。

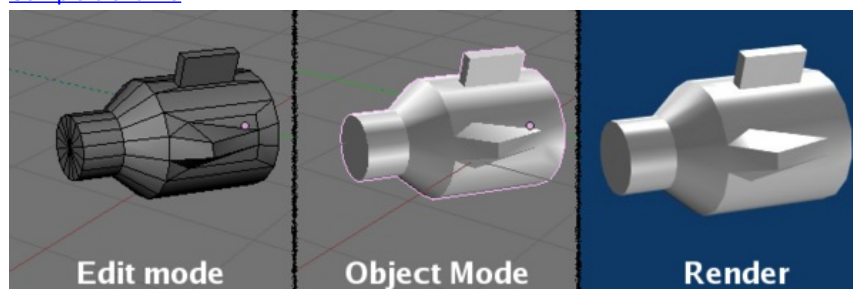
From Marked As Sharp

このボタンが有効の場合、「シャープ」マークが付いている辺が分離されます。マークするにはEdge Specials » Mark Sharp (Edit modeでCtrlE)を使ってください。

例



EdgeSplit modifier output with From Marked As Sharp selected.

[Sample blend file](#)

EdgeSplit modifier output with From Edge Angle selected.

[Sample blend file](#)

Maskモディファイア

Mode: Any mode

Panel: Modifiers

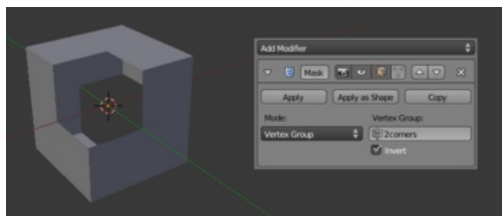
解説

Maskモディファイアはオブジェクトメッシュの特定の部分を見えなくします。その結果、その部分は存在しないかのように振舞います。

オプション

Mode

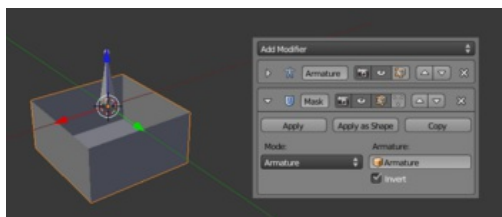
Maskモディファイアは二つのモードでメッシュを隠します。ドロップダウンリストから選択してください。



Vertex Group

Vertex Group

Vertex Groupが選択されていると、Maskモディファイアは指定された頂点グループに基づいてメッシュをマスクします。頂点グループの名前を入力すると、そのグループの頂点が存在する部分はマスクされる(これは通常「見える」ということです)ようになります。そしてグループに属していない頂点は見えなくなります。メッシュに頂点ウェイトが割り当てられていても問題ありませんが、ウェイトの数値は完全に無視されます。Maskモディファイアはその頂点がグループに属しているかどうかだけを考慮し、ウェイトは考慮しません。たとえばウェイトが**0.5**だからといって半分だけマスクされるということはありません。仮にウェイトが**0.0**だとしても、グループに属してさえいればマスクされます。



Armature

Armature

Pose Modeや、アーマチュアを編集しているときに便利です。アーマチュアオブジェクトの名前をテキスト欄に入力してください。Pose Modeでボーンを動かしているとき、アクティブなボーンに関わっていない頂点がマスクされます。Inverseボタンを有効にすれば、そのボーンがメッシュにどう影響するかを見ることができます。

Inverse

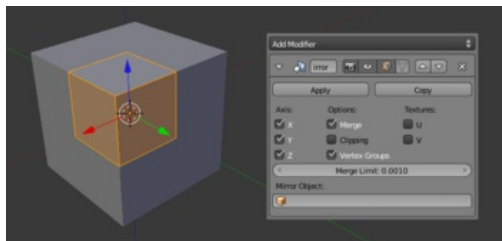
通常、Maskモディファイアがメッシュに適用されると、その影響下にある部分は見えたままで、それ以外の部分が見えなくなります。Inverseボタンはその振る舞いを逆にして、通常見えなかった部分は見えるようになり、見えていた部分は見えなくなります。

Mirrorモディファイア

Mode: Any mode

Panel: Modifiers

解説



The corner of a cube mirrored across three axes to form... well... a cube.

Mirrorモディファイアは、メッシュの中心を通るローカルなX、Y、Z軸に沿って自動的に鏡像を作り出します（鏡面は選んだ軸以外の二つの軸で規定されることになります）。他のオブジェクトの中心および軸を鏡として使うこともできます。鏡面から指定した許容値内の距離にある頂点を鏡面で結合することもできます。元のオブジェクトの頂点が鏡面の反対側に通り抜けることができないようにすることもできます。そしてもう一つ重要なのは、頂点グループやUV座標も反転してコピーすることができます。

オプション



Mirror modifier

Axis

指定した軸 (X, Y, or Z軸) に沿って鏡面が作られます（つまりこの軸が鏡面に対して垂直な軸となります）。X軸を指定した場合、元のメッシュにおけるプラスのX値が鏡像ではマイナスになるということです。

複数の軸を選択することもできます。よりたくさんの鏡像が生成され、鏡自体も完全に鏡面コピーされます（つまり、1つの軸を選んだ状態では鏡は1つ、2つの軸では4つ、3つすべての軸を選択すると鏡は8つになります）。

Options

Merge

鏡面上の頂点を結合します。後述のMerge Limitを参照。

Clipping

頂点が鏡面を通り抜けるのを防ぎます。これはEdit modeのときだけであることに注意してください（Object modeで移動や拡大縮小などをした場合は頂点が境界を通り抜けてしまいます）。

Clippingが有効になっていても、頂点がMerge Limitよりも鏡面から離れたところにあれば、結合されることはありません。Merge Limitの範囲に入ると二つの頂点はくっついて、鏡面上から動かなくなります。鏡面からの距離が異なる複数の頂点を移動しているときは、一つ一つ個別にくっついていきます。

頂点が鏡面にくっついた状態でLMBを押して移動を決定してしまうと、鏡面からひきはがすにはClippingを無効にする必要があります。

Vertex Groups

このボタンが有効だと、Mirrorモディファイアは頂点グループの鏡像を作り出します。非常にすばらしい機能ですが、これには厳しい条件があります。

- まず、鏡像を作りたい頂点グループは慣例に沿った右／左の命名規則で名づけられていなくてはなりません（“.R”、“.right”、“.L”などの接尾語）。
- 次に、鏡像となるグループをあらかじめ作っておく必要があります（つまり名前は同じで接尾語が左右逆のグループ名）。しかもグループは空（頂点が一つも割り当てられていない）でなければ動作しません。

通常は、コピーされた頂点はももとのグループにそのまま属します。しかしこのオプションを有効にすると、上述のルールに合致するグループに関しては、元のオブジェクトの頂点は元のグループに、鏡面コピーされた頂点は鏡像グループに属します。これはアーマチュアを使うときに非常に便利です。たとえば、オブジェクトの半分だけをモデリングして、慎重にアーマチュアの半分をリギングし、そしてMirrorモディファイアを使ってオブジェクトのもう半分を生成します。ArmatureモディファイアはMirrorモディファイアよりも下の段になるように注意してください。最後に、複数の軸を選択している場合ですが、元のオブジェクトから直接コピーされた頂点は鏡像グループに属します。コピーのコピー（つまり2回反射している）である頂点は、またもとのグループに戻ります。このような繰り返しです。

Textures

UとVのオプションはそれぞれUとVのテクスチャ座標を鏡面コピーします。値は**0.5**を境に反転されます。すなわち、両方のボタンを有効にした場合、元の頂点のUV座標が**(0.3, 0.85)**なら、鏡面コピーされた頂点のUV座標は**(0.7, 0.15)**になります。

Merge Limit

元の頂点とコピーされた頂点を結合する最大距離です。これ以下の距離まで鏡面に近づくと両者は結合され、元のメッシュとコピーのメッシュが

リンクされます。

Mirror Object

鏡像生成時に参照するオブジェクトの名前(通常はこの欄は空にします)。このオブジェクトの中心と軸が鏡面として使われます。もちろんその位置や回転を(Ipoカーブなどを使って)アニメーションさせれば、鏡像効果もそれに従います。

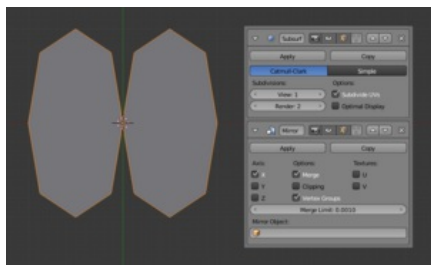
ヒント

モデリング作業の多くはシメトリカルなオブジェクトの制作を含んでいます。しかしながら、長年の間、こういったオブジェクトの両面を簡単に作るには賢明なBlenderアーティストたちが見つけた方法を使うしかありませんでした。一般的なテクニックのひとつは、モデルの片側を作り、AltDでリンク複製することです。このとき指定した軸に沿って反転できます。そうすると、編集作業がリアルタイムに反映される完璧な鏡面コピーができあがります。

Mirrorモディファイアはもうひとつの、よりシンプルな方法を提供してくれます。ひととおりモデリングが終わればApplyを押して実体のあるメッシュを生成することもできるし、さらなる編集作業のためにそのままにしておくこともできます。

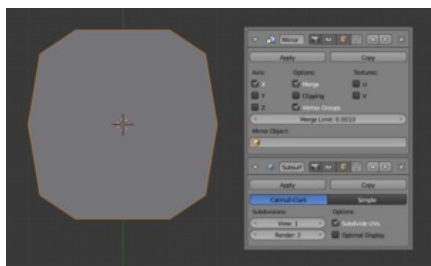
MirrorモディファイアとSubdivision Surfaceモディファイア

MirrorモディファイアとSubsurfをあわせて使う場合、モディファイアの順序が重要です。



Subsurf modifier before Mirror modifier

これはSubsurfモディファイアを先に、Mirrorモディファイアを後に適用した場合です。メッシュは鏡面の部分で裂けてしまっています。



Mirror modifier before Subsurf modifier

これはMirrorモディファイアが先で、Subsurfモディファイアが後の場合です。この順序だと、鏡面の部分がぴったりくっついています。たいていはこちらのほうが望み通りの結果でしょう。


Mirrorするために整列させる

Mirrorモディファイアを適用するために、鏡面の軸となるべき辺や面にオブジェクトの中心がちょうどくるようにしなければならない、ということはよくあります。これを目で見ながらやると難しい場合もあるでしょう。正確な位置を割り出す良い方法は、鏡面に垂直な辺を見つけることです。その辺の二つの頂点を選択し、[Cursor to Selection](#)で解説されている⇧ ShiftSを使います。これで3Dカーソルは二点のちょうど中間に移動します。最後に、Editingコンテキスト(F9)で、MeshパネルからCenter Cursorを選択すると、オブジェクトの中心点が3Dカーソルの位置に移動します。これで鏡面コピーはうまくいくでしょう。

もしくは、空のオブジェクトを別に用意して、鏡面の中心としても良いでしょう。

Multiresolutionモディファイア

Mode: Object mode

Panel: Properties Window -> Context Button Modifiers 

Multiresolutionモディファイアを使うと、3D View、[Template:Sculpt Mode](#)、Blender Renderの各状況に応じて違った度合いでメッシュを再分割できます。

オプション

Catmull-Clark / Simple

再分割のタイプを設定します。Simpleは形状は変えずに単純に分割するだけです。Catmull-Clarkは滑らかな表面を作り出します。これはもとの形状よりも小さくなります。

Preview

3D Viewでの再分割レベルを指定します。

Sculpt

Sculpt Modeでの再分割レベルを指定します。

Render

レンダリング時の再分割レベルを指定します。

Subdivide

再分割レベルを一段階追加します。

Delete Higher

現在よりも高いレベルの再分割を消去します。

Reshape

頂点の座標を他のメッシュからコピーします。これを使うには、まずトポロジーや頂点の座標が適合している他のメッシュを選択し、それからShift選択で目的のメッシュを選択してReshapeをクリックします。このメッシュはもう一つのメッシュと同じ形になるでしょう。

Apply Base

再分割後のメッシュに近い形になるように元のメッシュを変形します。

Optimal Display

再分割によって増えた辺の描画は省略します。


Save External

変更を外部.btxファイルに保存します。

Remesh Modifier

(リメッシュモディファイア)

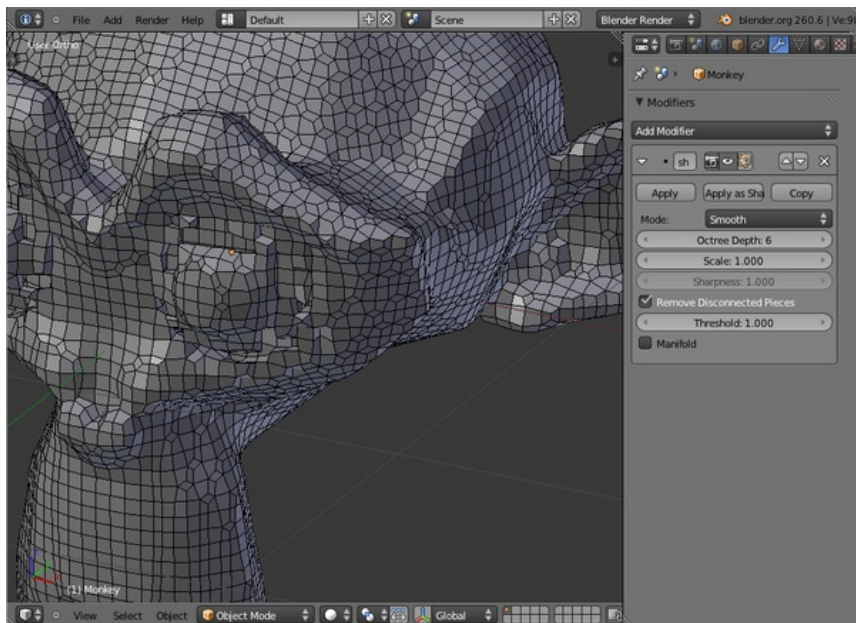
モード: オブジェクトモード

パネル: Properties Window -> Context Button Modifiers 

Description

(解説)

remesh モディファイアは入力サーフェスをもとに新たなメッシュトポロジーを生成するためのツールです。出力は入力サーフェスの曲率を守りますが、トポロジーは四角面だけを含みます。



Usage

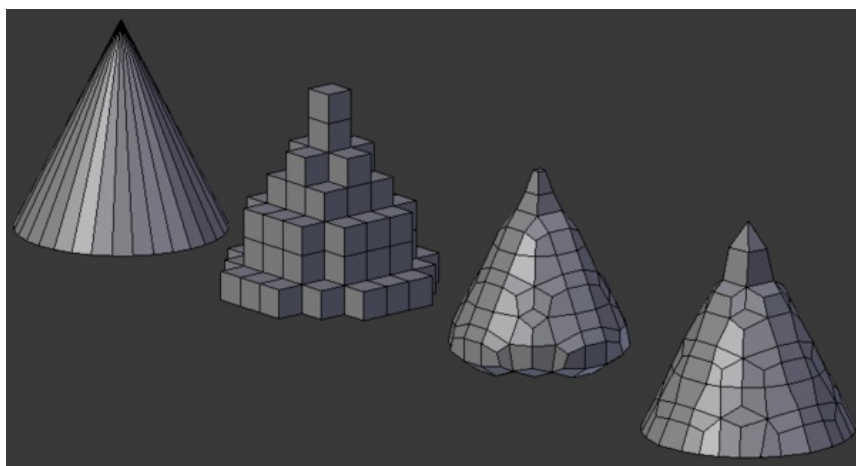
(使い方)

モディファイアパネルで remesh モディファイアを追加します (このモディファイアはメッシュだけに使えます)。入力メッシュには厚さが必要です。入力が完全に平らな場合は、remesh モディファイアの上に solidify (厚みづけ) モディファイアを追加してください。

Mode

(モード)

remesh モディファイアには3つの基本モードがあります。Blocks、Smooth、そして Sharp です。



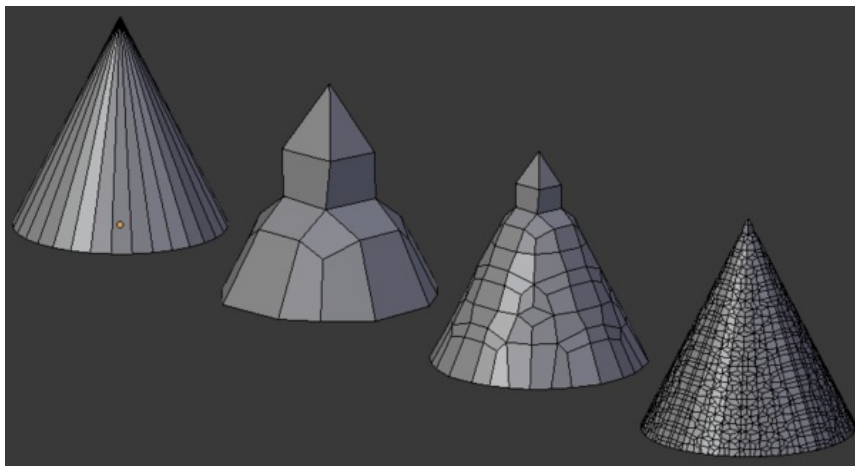
異なる remesh モードを使った円錐の例。左から元の円錐、Blocks、Smooth、Sharp

なお、出力トポロジーは3つのモードのどれもほとんど同じです。変わるのは平滑化 (smoothing) です。Blocks モードでは平滑化はまったく行われません。Smooth と Sharp モードはメッシュがすでに滑らかな場所では同じ出力を生成しますが、Sharp モードは細部に渡って鋭利さをもっとうまく残します。この例では Sharp モードのとき、円錐の円形の底と円錐の頂点が正しく再現されています。

Octree Depth and Scale

(八分木の深さと拡大縮小)

Octree Depth(八分木の深さ)は出力の解像度を設定します。低い値は入力に比較して大きな面を生成し、高い値は高密度の出力を生成します。Scale(拡大縮小)をセットすると、結果をさらに微調整できます。値を下げるほど効率よく出力解像度を下げることができます。

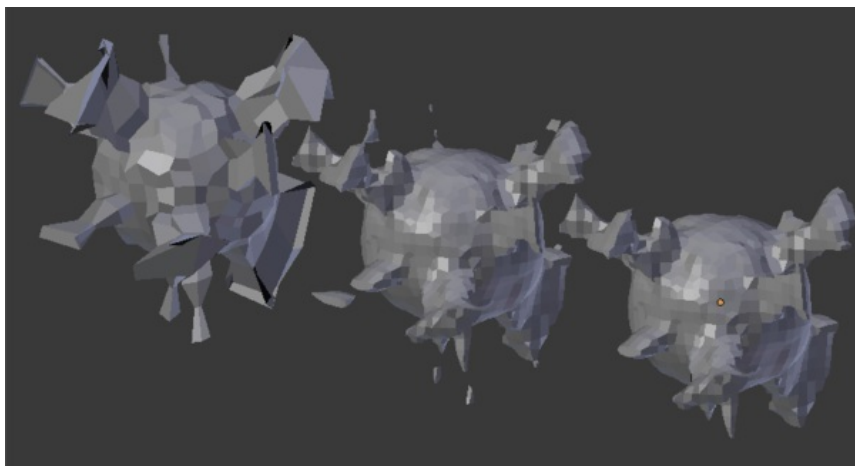


入力メッシュおよび低～高解像度で出力されたメッシュ

Disconnected Pieces

(とぎれた要素)

出力からとぎれた小さな要素を取り除くには、Remove Disconnected(接続されていない部分を削除)を有効にして threshold(しきい値)をセットし、取り除く必要のあるとぎれた要素の大きさを調整します。



入力メッシュ(左)は相当不規則なので、remesh モディファイアの初めの出力(中央)には小さなとぎれた要素があります。Remove Disconnected Pieces を有効化すると(右)こうした面が削除されます。

Sharpness

(鋭利さ)

Sharp モードでは Sharpness 値を設定して、入力の鋭利な辺に出力をどれだけ近づけるか制御します(不規則さを排除するには低い値を使ってください)。

Demo Videos

(実演ビデオ)

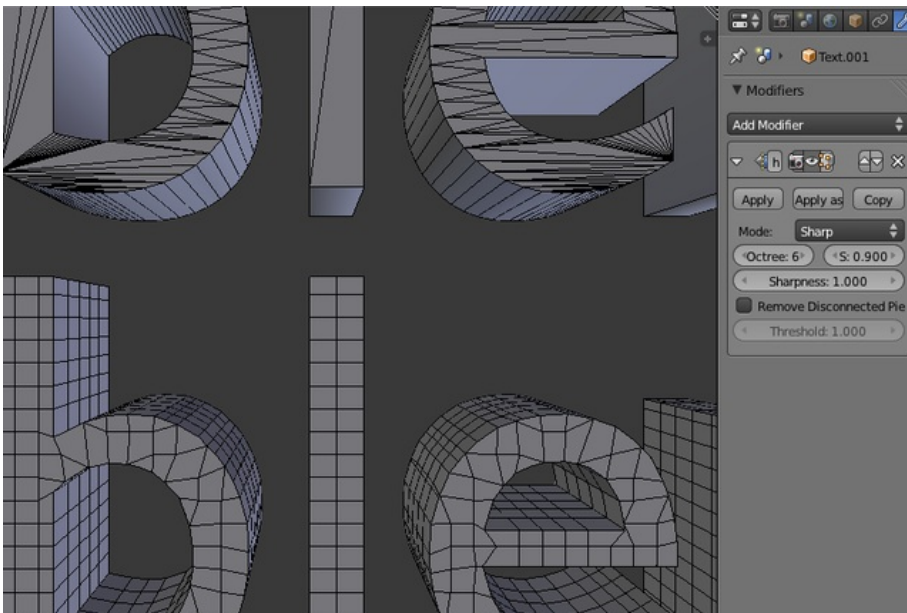
[\[video link\]](#)

[\[video link\]](#)

[\[video link\]](#)

[\[video link\]](#)

[\[video link\]](#) [\[video link\]](#)



トポロジー改良のためテキストに用いられた Remesh モディファイア

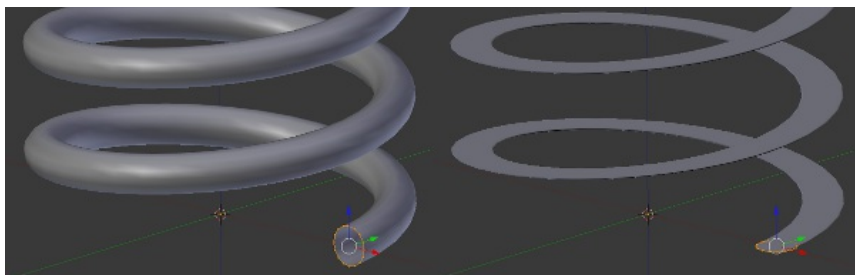
Screwモディファイア

Mode: Any mode

Panel: Modifiers

解説

ScrewモディファイアはTool ShelfのScrewツールに似ています。断面図を規定するメッシュオブジェクトやカーブオブジェクトを使って、らせんのような形状を生み出します。



Properly aligning the profile object is important

断面図は二次元の形状で、らせんの軸上ではなくオブジェクトの中心から離れた位置に適切に配置しなければいけません。

オプション



Screw modifier

Axis

らせんが構築される軸。

Screw

繰り返されるらせんの一つ分の高さ。

AxisOb

ここに入力したオブジェクトのAxisが使われます。

Object Screw

AxisObのオブジェクトを使って Screwの値を定義します。

Angle

一つのらせんで進む角度。

Steps

一つのらせんに使われるステップ数 ({{Literal|3D View}}内での表示)。

Render Steps

上と同様で、レンダリング時のステップ数。数値を大きくすると品質が上がります。

Calc Order

法線の問題を避けるために辺の順序を計算します。これはメッシュの場合だけで、カーブには必要ありません。

Flip

法線の方向を反転します。

Iterations

らせんを繰り返す回数。

Solidifyモディファイア

Mode: Any mode

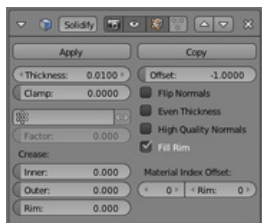
Panel: Modifiers

解説

Solidifyモディファイアはどんなメッシュにも厚みを与えることができます。

これはもともとCampbell Barton(別号Ideasman)によって書かれたスクリプトです。Solidifyは[Blender 2.44に同梱されていましたが](#)、Blender 2.5ではモディファイアに変化しました。

オプション



Solidify modifier

Thickness

与えられる厚み。

Offset

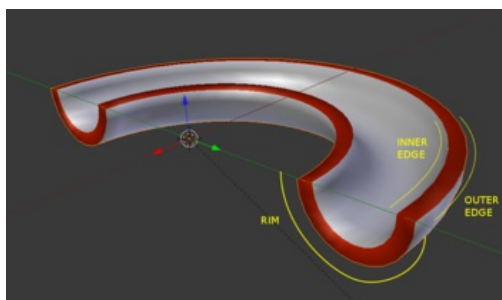
-1から1までの数値で、元のメッシュの内側に厚みが付くか外側に厚みが付くかを指定します。0にすると両側に均等に厚みが付きます。

Vertex Group

指定した頂点グループのみに厚みを加えます。

Invert

前項の制限を逆にして、指定した頂点グループのみ除外します。



Rim and edges. In this example, the object was assigned a second material used to color the rim red.

Crease

Inner

内壁の端の辺のcrease値。

Outer

外壁の端の辺のcrease値。

Rim

縁の部分の辺(内側と外側の間に横木状に連なる辺)のcrease値。

Even Thickness

鋭い角の部分でも厚みが同じになるように調整します。品質が上がる場合もありますが計算時間が増します。

High Quality Normals

より同じ厚みになるように法線を計算します。品質が上がる場合もありますがこれも計算時間が増します。

Fill Rim

内側と外側の間の縁を面で埋めます。

Rim Material

オブジェクトに設定されている第2マテリアルを縁に適用します。

再分割 (Subdivision Surface) モディファイア

Mode: Any Mode

Panel: Modifiers

Hotkey: F1F2

解説

Subdivision Surfaceはメッシュの面を再分割して滑らかな外観を生み出すことによって、少ない頂点のメッシュから複雑な表面を作ることができます。これにより膨大なデータを保存したり管理したりせずに高精度のメッシュモデリングができ、オブジェクトに有機的な見た目を与えることができます。どのようなメッシュに対しても、Blenderは必要に応じて滑らかな再分割の計算をします。モデリング時でもレンダリング時でも、またシンプルな仮のSubdivision Surface (略してSubsurf) でも、より高性能なCatmull-Clark法によるものでも可能です。

オプション



Modifier's panel

Subsurfは[モディファイア](#)です。これをメッシュに付加するには、Add Modifierを押して、リストからSubdivision Surfaceを選んでください。

Catmull-Clark/Simple

再分割のアルゴリズムを選択します。

- Simple:単純に面を分割し、滑らかにはしません (Edit modeでの W → Subdivide に似ています)。これはほぼ必要ないでしょう。
- Catmull-Clark:デフォルトの設定。再分割とスムーズ化を行います。

Subdivisions

3D Viewおよびレンダリング時の分割度を指定します。

Note

これら二つのセッティングによって、3D Viewで作業するときには軽量で近似的なモデルを扱い、レンダリング時にはより品質の高い設定で出力できます

Options

- Optimal Display:分割後にできる辺すべてではなく、もとの辺のワイヤフレームだけを表示します。
- Subdivide UVs:これを有効にすると、UVマップもSubsurfされます (つまりこのモディファイアによって再分割された面に対して仮想的な座標が与えられます)。

編集中に再分割の結果 ("isolines") を表示しておくには、Editing Cageモードを有効にする必要があります (モディファイアの順序を上下するボタンのすぐ隣のボタンです)。これにより元の頂点の位置ではなくSubsurf後の位置に頂点があるかのようにつかむことができます。

Weighted creases (重み付き折り目) for subdivision surfaces

Mode: Edit Mode (Mesh)

Panel: 3D View → Transform Properties

Menu: Mesh → Edges → Crease Subsurf

解説


Weighted edge creasesを使うと、Subsurfされるメッシュの辺をシャープにしたり滑らかにしたりできます。

オプション

crease weightを変えたい辺を選択して、Transform Properties (N)のスライダーを操作してください。数値を高くすると辺が「強く」なり、スムーズ化に抵抗する度合いが大きくなります。Subsurfされた形状は縦横に走る辺から定義されているので、ある面に属する辺を鋭くしたいときは面を取り囲むすべての辺の重みを高くする必要があります。

三角面化(Triangulate)モディファイアー

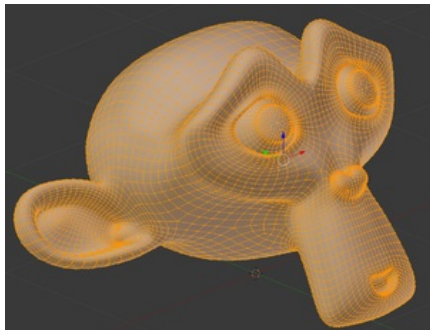
Mode: Object mode

Panel: Properties Window -> Context Button Modifiers 

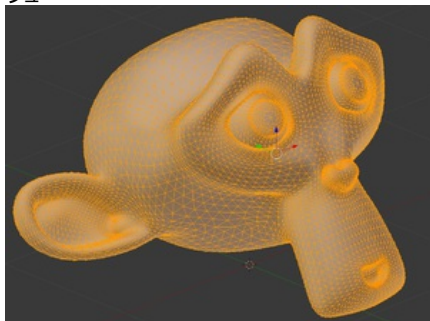
説明

三角面化(Triangulate)モディファイアーは、メッシュ内のすべての面(四角形またはNゴン)を三角形の面に変換します。

このモディファイアーは編集モード(Edit Mode)での[Ctrl]+[T]キーの三角面化(Triangulate)と全く同じ機能を実行します。



三角面化(Triangulate)モディファイアー前のメッシュ



三角面化(Triangulate)モディファイアー後のメッシュ

オプション

四角形分割方法(Quad Method)

高品質(Beauty)

四角形を綺麗な三角形に分割します。遅い方法です。

定値(Fixed)

四角形を1つめと3つめの頂点で分割します。

別の定値(Fixed Alternate)

四角形を2つめと4つめの頂点で分割します。

最短の対角線(Shortest Diagonal)

四角形を頂点間の距離を元に分割します。

Nゴン分割方法(Ngon Method)

高品質(Beauty)

新しい三角形を綺麗に組み換えます。遅い方法です。

クリップ(Clip)

Nゴンを ear clip アルゴリズムを使用して分割します。

アーマチュアモディファイア

Mode: Object mode

Panel: Modifiers

Description

アーマチュアモディファイアが使用されるのは、キャラクターのポーズをアニメーション化するための骨格システムを構築したり、その他にもポーズが必要なあらゆる用途です。頂点グループとマルチモディファイアオプションを使用すると、単一の(メッシュ)オブジェクトをアニメーション化するとき複数のアーマチュアを使用することができます。

オブジェクトにアーマチュアシステムを追加することによって、そのオブジェクトを正確に変形させることができるため、形状を手作業でアニメーション化する必要がなくなります。アーマチュアモディファイアは、アーマチュアオブジェクトの名前を指定することによって、ボーンによるオブジェクトの単純な変形ができ、これは(旧来の)“親/子”システムを使用せずに行うことができます。

アーマチュアの使用方法の詳細については、[こちらの章](#)を参照してください。

オプション



Armature modifier

Object

この修飾子によって使用されるアーマチュアオブジェクトの名前。

Preserve Volume

滑らかな、よりよい回転の補間を取得するには四元数を使用してください。

Vertex Group

オブジェクトの頂点グループの名前で、その重みづけを使用してこのアーマチュアモディファイアの結果への影響を決定します。他のアーマチュアの結果と混成させるときに使用します。マルチモディファイアが起動されているときには、同じオブジェクト上にこれらのモディファイアのうち少なくとも二つを持たせないと意味がなくなります。

MultiModifier

前の(アーマチュア?)モディファイアと同じデータを入力として使用します。“非変形”のデータすべてに基づいて、複数のアーマチュアを同じオブジェクトの変形に使用できます(つまり、この最初の結果を変形させるための第二アーマチュア修飾子を避けられます...)。アーマチュアモディファイアの結果は一緒に混合されて、“mixing guides”としてVGroup頂点グループの重みづけに使用されます。。

Bind To

メッシュにアーマチュアを結びつける方法。

Vertex Groups

頂点のグループが定義する変形を有効化/無効化する(すなわち、名前を指定されたボーンは、同じ名前のグループに属する頂点だけを変形させる)。

Bone Envelopes

が変形させる定義を有効/無効します(すなわち、ボーンが自分の隣の頂点を変形する)。

Invert

直前に設定した頂点のグループの定義によって、設定されている影響を反転します(すなわち、このグループの重みの値を元に戻します)。

キャストモディファイア

Mode: Any mode

Panel: Modifiers

Description

このモディファイアはメッシュ、カーブ、サーフェスまたは格子の形状を、いくつかの事前定義された形状 (球, 円柱, 直方体)のいずれかに移し変えます。

Editing コンテキスト内の To Sphere ツールと同等であり (Mesh → Transform → To Sphere Alt⇧ ShiftS)、他のプログラムのほうは "Spherify" または "Spherize", と呼ばれていますが、上に書いたように、球の成形に限定されるものではありません。

Hint

[Smooth modifier](#) は Castとは良い仲間で、表面をなめらかにして見栄えを良くしたりあるいは作品のシェーディングを修正するときにもキャストした形状が必要になることがあります。

Important

パフォーマンスを発揮させるためには、この修飾子は、ローカル座標でのみ動作させましょう。変更されたオブジェクトが正しく表示されない場合、とりわけシリンダーへ成形する場合、オブジェクトの回転 (CtrlA)を適用する必要があります。

オプション



cast modifier

Cast Type

キャストの型 (ターゲットの形状)を選択するメニュー: Sphere, Cylinder or Cuboid.

X, Y, Z

X、Y、Z軸方向で修飾子を有効/無効にするトグルボタン (シリンダーキャストのタイプではXとYのみ)。

Factor

オリジナルとキャストの頂点の位置の間のブレンドを制御する係数。It's a 線形補間します: **0.0** では元の座標を示し (修飾子は効果がないということでもある), **1.0** ではターゲット形状そのものと成形します。値 [0.0, 1.0] を上げたり下げたりするとメッシュを変形させ、時に面白い方法で行います。

Radius

ゼロ以外の場合、この半径は、影響力の球を定義します。その外側の頂点は、修飾子の影響を受けません。

Size

投影形状の代替サイズ。ゼロの場合、初期形状と制御オブジェクトによって定義されます。

From radius

有効化すると、半径から サイズ を計算し、結果を滑らかにします。

Vertex Group

頂点グループの名前であり、頂点への影響をグループ内だけに制限します。頂点の重みづけを塗り分けることにより、選択的で、リアルタイムな成形をすることができます。

Control Object

効果をコントロールするオブジェクトの名前。このオブジェクトの中心の位置は、投影の中心を定義します。また、その大きさと回転によって投影頂点を変換します。Hint: このコントロール用オブジェクトをアニメーション (キーフレーム) にすると、修飾されている側のオブジェクトもまたアニメーションにされます。

カーブモディファイア

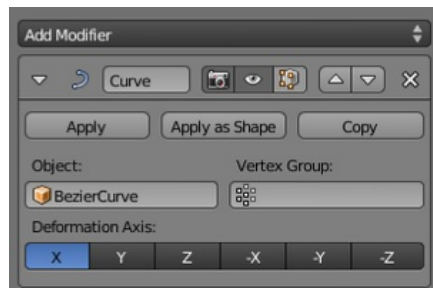
Mode: Object mode

Panel: Modifiers

Description

カーブ モディファイアはちょうど例外を追加した [predecessor](#) のように機能し、曲線と変形されるオブジェクトとの間に親/子関係の必要性はなく、そして効果はリアルタイムですべてのオブジェクトタイプに適用できます。

オプション



Curve modifier

Object

変形オブジェクトに影響を与える曲線オブジェクトの名前。

Vertex Group

変形オブジェクト内の頂点グループの名前。モディファイアは、このグループに割り当てられている頂点だけに影響を与えます。

Deformation Axis

X, Y, Z, -X, -Y, -Z: カーブはこの軸に沿って変形します。

置換 (Displace) モディファイア

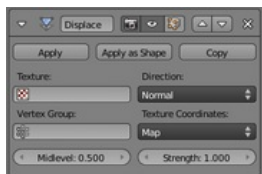
Mode: Any mode

Panel: Modifiers

解説

Displaceモディファイアはメッシュの頂点をテクスチャの強度 (intensity) に応じて置換 (移動) します。手続き型テクスチャでもイメージテクスチャでも使えます。ローカル座標軸や頂点の法線に沿って置換することもできるし、テクスチャのRGBそれぞれの要素をつかってローカルX軸、Y軸、Z軸の置換を同時に行うこともできます。

オプション



Displace modifier

Texture

置換に使うテクスチャの名前。
この欄が空だとモディファイアは無効になります。

Vertex Group

モディファイアの影響を受ける頂点グループの名前。
If VGroupが空だとモディファイアはすべての頂点に等しく影響します。

Midlevel

置換の基準になる値を設定します。テクスチャの各点の値が、この値よりも小さければ指定された置換方向に対してマイナスの向きに置換され、この値よりも大きければプラスの方向に置換されます。このような式になります。置換の高さ = テクスチャの値 - Midlevel
Blenderではカラー値や輝度などは**0.0**から**1.0**で表します。**0**から**255**ではないので注意してください。

Direction

置換される方向。
以下から選べます。

- X - ローカルX軸に沿って置換します。
- Y - ローカルY軸に沿って置換します。
- Z - ローカルZ軸に沿って置換します。
- RGB -> XYZ - テクスチャのRGBそれぞれの要素に基づいて、ローカルXYZ軸それぞれに沿って置換します。
- Normal - 頂点の法線方向に置換します。

Texture Coordinates

各頂点がテクスチャから値を受け取る時のテクスチャ座標。
以下から選べます。

- UV - 面のUV座標に従います。

UV Layer

使用するUVレイヤー。
オブジェクトにUV座標が設定されていない場合はローカル座標が使われます。UV Layer欄が空で、しかしUVレイヤーそのものは存在する場合 (たとえばUVレイヤーを追加した直後などはそうなります)、アクティブなUVレイヤーが設定されます。

Note

UV座標は面ごとに設定されるので、現在のところ頂点のUVテクスチャ座標はその頂点を使う面のうちで最初に見つかった面から設定されます。他の面は無視されます。このことから、UV座標が連続していない場合は不自然な結果になる可能性があります。

- Object - 以下の欄で指定したオブジェクトの座標系をテクスチャ座標系に使います。

Object

ここで設定したオブジェクトの座標系を使います。よってこのオブジェクトを動かすとテクスチャのマッピング座標も変化します。また、置換を施すほうのオブジェクトを移動させた場合も座標系は変化することに注意してください。オブジェクトを移動してもテクスチャを同じ座標系のままで保ちたいなら、座標の指定に使っているオブジェクトも同じ方向に同じ割合で移動する必要があります。この欄が空の場合はローカル座標が使われます。

- Global - オブジェクトのグローバル座標に基づいてテクスチャ座標を設定します。
- Local - オブジェクトのローカル座標に基づいてテクスチャ座標を設定します。

Strength


置換の強さ。Midlevelに基づいてオフセットが計算されたあと、Strengthを掛け算したものが最終的な置換の値になります。このような式です。頂点の移動量 = 置換の値 × Strength。
負の値を使うとモディファイアの効果を反転できます。

以下も参照してください

- Blender artists post: [Displace modifier tutorial](#) (September 2006)

ラティスマディファイア

Mode: Object mode

Panel: Properties Window -> Context Button Modifiers 

The ラティスマディファイアは、Lattice オブジェクトの形状に応じてベースオブジェクトを変形します。

オプション



Lattice modifier.


Object

Lattice オブジェクトへ変形させるベースオブジェクト。

Vertex Group

オプションな頂点グループの名前で、モディファイアが効果を及ぼすベースメッシュの部位を制限することができます。

Hints

プロパティウィンドウにある、ラティス用のオブジェクトデータコンテキストでは、ラティスオブジェクトの属性を制御できます .

ラティスは、非レンダラブルな3次元グリッドの頂点から構成されています。これらの主な用途は、それらの制御の基礎となるオブジェクトに追加的な変形能力を与えることです (モディファイア *経由* でも、またその親でもどちらでも)。これらの“犠牲者”となるオブジェクトは、メッシュ、カーブ、サーフェス、テキスト、格子、さらには粒子までも。

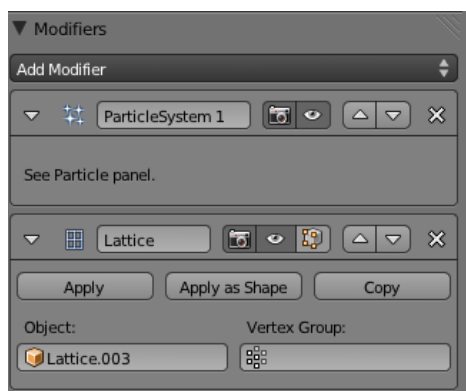
なぜ エディットモードでメッシュ自体を変形させるのではなく、格子を使ってメッシュを変形させるのでしょうか？ その理由はいくつかあります：

- まず第一に: それが簡単だからです。メッシュに膨大な数の頂点があるとしたら、それらをスケーリングしたり、つかんだり、移動させたりすることは大変な作業になるかもしれません。その代わりにナイスでシンプルな格子を使用すれば、あなたの仕事は簡易になり、いくつかの頂点を移動するだけですみます。
- 出来栄が良くなります。変形で得られる見栄えはかなり良くなります!
- 速いからです! 同じ格子を、複数のメッシュの変形に使用できます。各オブジェクトにラティスマディファイアを与えるだけで、どの点においても同じ格子になります。
- 良い習慣がつくからです。ラティスを使えばいろいろなバージョンのメッシュを取得でき、余分な作業とリソースの消費は最小限ですみます。これにより、シーンのデザインを最適化し、モデリングの作業量を最小限に抑えることにつながります。ラティスは、メッシュの表面のテクスチャ座標には影響しません。メッシュオブジェクトへの微妙な変更は、このやり方で容易に円滑化されるので、メッシュ自体を変更しないようにしましょう。

Example/Tutorial(s)

[チュートリアル](#)のセクションには、2.4バージョンのチュートリアル例があります。2.5のチュートリアルでは、[フォークを形作る](#)方法を解説しています。

パーティクルとラティス



Particles following a lattice.

モディファイアの順序が正しければ、パーティクルはラティスに従います。最初がパーティクルで、その後がラティスです!

メッシュ変形モディファイア

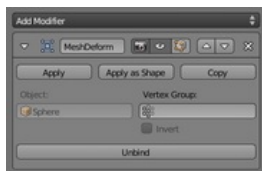
Mode: All modes

Panel: Modifiers

概要

Mesh Deformモディファイアは任意のクローズドメッシュに使用でき(ラティスマディファイアとちがって直方体形状に限らず、任意の閉じた図形)、別のメッシュ周りの変形ケージとして動作します。

オプション



Mesh Deform modifier

メッシュ変形 モディファイアは簡単に使えて合理的ですが、変形メッシュケージを変形オブジェクトに正確にマッピングするときなどは、必要な計算を行うのが非常に遅くなることがあります。

Object

変形メッシュのケージとして使用するメッシュオブジェクトの名前。

Vertex Group

変形メッシュケージの影響を受ける、オプションな頂点グループ。

Invert

以前の設定で定義された頂点のグループによって設定された影響を反転します(すなわち、このグループの重みの値を元に戻します)。

Bind

Bind ボタンはメッシュ変形 モディファイアに、変形メッシュケージを変形オブジェクトへと実際にリンクさせるよう指示するものです。こうすることにより、変形メッシュケージの形状を変えると、変形オブジェクトの形状を実際に変化させるようになります。メッシュ変形 モディファイアの依存する設定と、変形メッシュケージおよび/または変形オブジェクトの複雑さに注意してください。この操作を完了させるまでに長い時間がかかることがあります。これにより、その処理が完了するまでは、Blenderがユーザーのアクションに応答しなくなることもつながらります。Blenderがメモリを使い切ってしまったり、クラッシュするおそれもあります。

Unbind

変形オブジェクトが変形メッシュケージに関連付けられているとき、Unbind ボタンを選択することで、Bind と取り替えて、後から関連付けを解除することができます。

Unbind をクリックすると、*deform mesh cage* はその時点の形状を維持し、その元々の開始の形状に自分自身を戻してリセットすることはありません。元の形状が必要な場合は、それを変更する前にコピーを保存しておく必要があります。しかし変形オブジェクトは、変形メッシュケージにbindされる前の元の形状にリセットされます。

Precision

精密 数値スライダのフィールドでは、ケージ上の点を移動させるときに、変形メッシュケージが変形オブジェクトを変更する精度を制御します。Precision フィールドの値の範囲は、デフォルトでは5で、2 から 10の範囲で指定できます。この値を高く引き上げると、バインディングのための計算をメッシュ変形 モディファイアが完了するまでにかかる時間は大幅に増えますが、変形オブジェクトのケージマッピングはより正確になります。この計算時間の上昇により、必要なものを算出するまでBlenderが応答を停止することもあります。同様に、Blenderが反応しないからといってPrecision 値を高く上げ、そのうえ非常に複雑な変形メッシュのケージおよび/または変形オブジェクトに Bind させようとすると、大量のメモリを消費して、極端な場合Blenderをクラッシュさせます。安全を期すために、先に進む前にあなたのblendファイルを保存しておきましょう! ケージをバインドした後はこの設定は使用できなくなります。

Dynamic

Dynamic ボタンが指し示すのは、メッシュ変形 モディファイアが考慮すべき、変形オブジェクトに関わる変形および変更のうち、変形メッシュケージの直接的な結果以外のものです。

With Dynamic ボタンを有効にすると、他のメッシュの機能が変わり(他のモディファイアおよびシェイプキーと同様)、変形メッシュケージを変形オブジェクトにバインドする際に考慮されるようになり、変形の質を高めます。バインド時のメモリや処理時間を節約するために、デフォルトでは無効になっています...

Precisionと同様に、ケージがバインドされた後はこの設定は使用できません。

See Also

- ラティスマディファイア

Shrinkwrap Modifier

Mode: All modes

Panel: Modifiers

概要

シュリンクラップ モディファイアは、オブジェクトを他のオブジェクトの表面に“縮める”ことができます。オブジェクトの各頂点を移動させて、指定したメッシュに最も近い表面上の位置へと編集します (利用可能な3つの方法のいずれかを使用)。これは、メッシュ、格子、曲線、曲面およびテキストに適用することができます。

変形モディファイアのほとんどにおいて、影響を受ける“頂点”は“計算上の”ものです。すなわち、モディファイアが計算された時点でのオブジェクトの実際の幾何学形状であって、オリジナルの *頂点*/コントロールポイントではありません。

ちょっとしたビュー非依存の [retopo tool](#) (in Blender 2.49)、シュリンクラップは頂点を法線に沿って投影させたり、または最寄りの表面のポイントに移動させます。しかしretopoにあったような精度の問題を起こすことはなく、それは画像の空間の代わりにオブジェクトスペースで動作するからです。また、標的位置から“距離を保つ”こともできます。

Note
シュリンクラップ モディファイアがすごく有用なのはわかったけど、でもエンティやオブジェクトの位置の移動もさせたいという方は... [シュリンクラップ拘束](#)を参照してください!

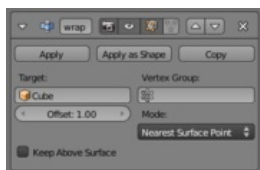
オプション

Target
ターゲットをシュリンクさせ、メッシュのまわりに shrink/wrap around させる。

Vertex Group
このウェイトペイントは、その時に修正されるメッシュの頂点グループのためのもので、各頂点をその標的位置に変位させるかどうか、あるいはその量をコントロールします。頂点がこのグループのメンバーでない場合は、変位されません (重量 0 と同じ)。

Offset
計算された標的位置から離しておく必要がある距離。Blender 単位で指定。

Subsurf Levels
このオプションでは、ラップを計算する前に、変更されるオブジェクト(またはそのターゲット?)に(一時的な)キャットマルクラーク 細分割を適用しておく必要があります... そうしないと、まったく効果を持たなくなってしまうようです...



Nearest Surface Point

Mode
このドロップダウンリストでは、変更されるオブジェクトの各頂点において、ターゲットの表面上で最も近い点を決定するときに使用するメソッドを指定します。一部のオプションでは、いくつかの特別な、固有のコントロールがパネルに追加されます。

Nearest Surface Point
これは、シュリンクターゲットの表面上で最も近い点を選択します。表面上に特別なオプションを追加し、計算された頂点を常に“床面”の上に保ちます。これは オフセット がnullではない場合にのみ意味があります。



Project

Projection
頂点を、シュリンクターゲットに触れるまで、選択した軸に沿って投影します。シュリンクターゲットに接触しない頂点は元の位置に残されます。これが意味するのは、このオプションの設定および2つのオブジェクトの相対位置に依って、変更をうけるオブジェクトが変形ないままになることもあるということです。これはバグではなく、単に設定を“いじる”か (特に、Negative/Positive のもの)、または周りのオブジェクトのいずれかを移動させましょう...

このメソッドは、ときどき予期しない結果をもたらすので、習得が最も困難です... 追加オプションもかなりの数が増えます:

X, Y, Z

変更をうけるオブジェクトは、ローカル軸に沿って投影が行われます。これらのオプションは相互に組み合わせることができ、投影の“中央値軸”の影響をうけます。

Negative, Positive

収縮の方向(複数可)を選択することで、選択した軸に沿わせることができます。複数の シュリンクラップ 修飾子を使用して、負と

正の軸を組み合わせることができます。

Cull Faces

これは、ターゲットの表面の“フロントサイド”(“バックサイド”でも同様)上の任意の投影を防止することができます。どちらの“側の”表面なのかはその法線によって決定されます(フロントは法線の“起点”となっている“ほうの”サイドになります)。

Auxiliary Target

投影時の追加オブジェクト。



Nearest Vertex

Nearest Vertex

これは、シュリンクターゲットのうち最も近い頂点を選択します。余分なオプションの追加はありません。

Simple Deform Modifier

Mode: All modes

Panel: Modifiers

概要



Simple Deform

Simple Deform modifier は、いくつかの簡単な変形を、オブジェクトに容易に適用することができます (メッシュ、格子、曲線、曲面およびテキストがサポートされています)。

ほとんどの変形モディファイアにおいて、変形機能は "計算済みの頂点" に適用されます。すなわち、オリジナルの *頂点* コントロールポイントではなく、修飾子が計算された時点でのオブジェクトの実際のジオメトリに。これが意味するのは、Subdivision Surface modifier (メッシュ用) を最初に挿入するとき、またはプレビューの解像度 の設定 (for curves/surfaces/texts) を上げることで、変形の効果の詳細レベルを上げることができるということです。

別のオブジェクトを使用して、変形の軸と原点を定義することが可能で、まったく異なるさまざまなエフェクトを適用できます。

オプション

Mode

このドロップダウンリストでは、適用する変形機能を定義します。4つのうちから使用可能です:

- Twist – Z軸を中心として回転します。
- Bend – メッシュをZ軸へ曲げる。
- Taper – Z軸に沿って直線的に拡大縮小する。
- Stretch – Z軸に沿ってオブジェクトを伸縮します(負の係数だと押し縮める)。

Vertex Group

各頂点の変形に影響されるかどうか、およびその程度を示す頂点グループの名前。

Origin

変形の原点を定義するオブジェクトの名前 (通常は空)。このオブジェクトは次のいずれかです:

- 回転させて軸を制御する (それ自体のZ軸を、このときに "ガイド" として使用する)。
- 平行移動させて変形の原点を制御する。
- スケーリングによって変形係数を変更する。

Note

起点を制御するオブジェクト (Origin フィールドにあるもの) が変形オブジェクトの子であるときは、これはBlenderのデータシステムに周期的な依存関係を生成してしまいます (the DAG – "dependency graph"?)。回避策は、エンプティを作成し、それに両方のオブジェクトを添付することです。

Factor

変形量。マイナスに設定することもできます。

Limits

これらの設定では、変形の下限と上限を設定することができます (**0.0** to **1.0** までの、比例値となります)。言うまでもなく、上限値は下限値より小さくなることはありません。

Lock X Axis/Lock Y Axis (Taper and Stretch modes only)

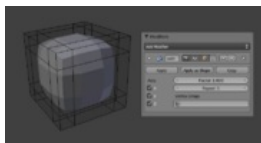
XおよびまたはY座標への変更を許可するかどうかを制御します。したがって、オブジェクトのY座標はそのままに保ちながら、X座標を押しつぶすこともできます。

平滑化 (Smooth) モディファイア

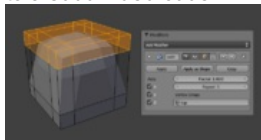
Mode: Any mode

Panel: Modifiers

解説



Smooth modifier applied to a subdivided cube



As above, with a vertex group selected

このモディファイアは隣り合っている面の角度を緩やかにすることで、メッシュを滑らかにします。EditingコンテキストのSmoothと同様です。つまり再分割をせずに滑らかに整えるので頂点の数は変わりません。

使い道は滑らかにするだけにとどまりません。効果を制御する数値は[0.0, 1.0]の範囲を超えて設定できます。そのような設定をすると、元のメッシュに応じて面白い効果が表れます。

オプション

X, Y, Z

XYZそれぞれの軸方向に関してモディファイアの有効／無効を切り替えます。

Factor

平滑化の量を決める数値です。滑らかにする場合の範囲は0.0から1.0です(0.0:無効、0.5:Smoothボタンと同じ、1.0:最大の効果)。また、この範囲外の数値(0.0より下か0.5より上)を設定するとメッシュをゆがめる効果になります。

Repeat

平滑化を行う回数。Smoothボタンを何度も押すのと同じ効果です。

Vertex Group

ここで指定した頂点グループのみに効果を与えます。ウェイトペイントをしながらリアルタイムで平滑化することもできます。

ワープモディファイア

この変形モディファイアを使用すると、メッシュのパーツを新しい場所のほうへ歪曲させることができます。"from" と "to" の場所の選択に2つのオブジェクトを使う、非常に柔軟な方法を使用します。オプションはカーブフォールオフ、テクスチャおよび頂点グループを使用します。

ワープモディファイアは、最初は少しトリッキーですが、それ自体の動作の仕方が、理解を助けてくれるでしょう。モディファイアは、オブジェクトの中心点で指定された2つのポイントが必要とします。"from"ポイントが指定するのは、"to" ポイントに引っ張られていく区域にある点です。エディットモードにおいて [Proportional Editing](#) を使用するときと似ています。

Options

From:

湾曲の起点となるオブジェクトの変換を指定します。

To:

湾曲先のオブジェクトの変換を指定します。

Preserve Volume

変換の1つを入れ替える時にボリュームの保存を有効にします。

Vertex Group

特定の頂点グループへの変形を制限する。

Strength

効果をどれくらい強くするかを設定します。

Radius

変形ハンドルで湾曲させることができる変換対象からの距離を設定します。

Falloff Type

変換の中心地から遠ざかるにしたがって、半径の値によってワープの強度を変える方法に設定します。減衰型の説明については、[Proportional Editing](#) を参照してください。

Texture

転置の変化の作成によって強度が相殺されるテクスチャを指定します。

Texture Coordinates

テクスチャのついた湾曲を使用するときに、テクスチャをメッシュに適用する方法を設定します。

Object

オブジェクトに設定するときに使用するオブジェクトを指定します。

UV Layer

UVに設定をするときのUVレイヤを指定します。

ウェーブモディファイア

Mode: Object mode

Panel: Modifiers

概要

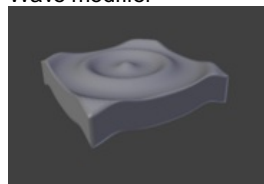
ウェーブモディファイアは、オブジェクトの頂点/コントロールポイントのZ座標に海のようなモーションを追加します。このモディファイアは、メッシュ、格子、曲線、曲面やテキストに使用でき、非メッシュオブジェクトではいくつか制限があります:

- 法線をアクティブ化または VGroup に名前を入力すると、それだけでモディファイアを非アクティブにします。
- さらに悪いことに、テクスチャ座標として UV を選択すると、いっぺんにBlenderをクラッシュさせます!

オプション



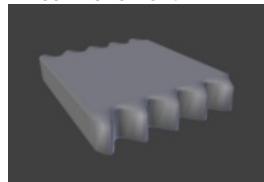
Wave modifier



Circular wave front



Linear wave front



Motion enabled for X and
Normals enabled for Y

Motion

X, Y, Cyclic: 波の効果によるZ方向への頂点/コントロールポイントの変形は、与えられた出発点を起点として円形の波先へとオブジェクトに沿って伝搬します (XとYの両方が活性化されているとき)。また、rectilinear wave frontsと併用し、X or Y ボタンを活性化すると、対応する軸に平行になります。Cyclic は、単一のパルスではなく、周期的に波を繰り返します。

Normals

メッシュ用途のみ。サーフェスの法線に沿ってメッシュを置き換える (オブジェクトのZ軸の代わりに)。

Time

時間パラメータを制御する設定。

Offset

フレームのオフセット時間。波が開始するフレーム (Speed が 正の場合)、または終了するフレーム (Speed が 負の場合)。波の始動および事前開始をさせるには、負のフレーム番号を使用してください。

Life

フレームでのアニメーションの持続時間。ゼロに設定すると、永遠にアニメーションをループさせます。

Damping

フレームの追加の数で、この中では波は、寿命に到達した後に高さの値からゼロへと、ゆっくりと減衰されます。減衰はすべてのさざ波において発生し、寿命を超える最初のフレームで始まります。波紋は、Damping framesを経て消えます。

Position

波の中心となるX座標とY座標で、オブジェクトのローカル座標。Falloff は、波がフェードアウトする速度を制御し、同時に上記の座標からどれだけ離れた位置まで波が到達するかも制御します。注意してほしいのは、Start Position Object を選択すると上記で選択した座標は実際にキャンセルされますが、しかし Falloff の値は保持されるということです。

Start Position Object

波の開始位置の基準として別のオブジェクトを使用してください。無効にするには空白のままにします。その後、このオブジェクトの位置をアニメーション化することで、時間に沿って波の原点を変更させられることも覚えておきましょう。

Vertex Group

メッシュ用のみ。波の作用がメッシュのどの部分に影響するか、およびその広がり制御に使用する、頂点グループの名前 (vertex weightsを使用する)。

Texture

オブジェクトの変位レベルを制御するには、このテクスチャを使用してください。アニメーションテクスチャは非常に興味深い結果をここにもたらしません。

Texture Coordinates

このメニューでは、変位につかうテクスチャの座標を選択できます:

Local

オブジェクトのローカル座標。

Global

グローバル座標。

Object

すぐ下に追加フィールドをつけ加えて、テクスチャ座標の取得元のオブジェクトの名前を入力できるようにする。

UV

余分な UV Layer のドロップダウンリストを追加し、使用するUVレイヤを選択できるようにする。 **Warning:** 非メッシュオブジェクトでこのオプションをアクティブにしてはいけません。Blenderをクラッシュさせることもあります。

Speed

フレームごとのBU (for "Blender Units") での、rippleの速度。

Height

BUでの、rippleの高さや振幅。

Width

BUで、波幅の半分。2つの隣り合うrippleの頂点の間隔 (Cycl が有効になっている場合)。これは、リップルの振幅に間接的な影響を持っています - パルスが互いに近すぎる場合、波が 0 Z-positionに到達しない場合もあり、そのためこのケースではBlenderによって波全体が実際に低くされ、最小値がゼロになるので、結果的に、最大値は期待振幅よりも低くなります。 [技術的な詳細](#) は以下を参照してください。

Narrowness

各パルスの実際の幅、値を高くするとパルスの幅は狭くなる。単一のパルスの外見上の領域の実際の幅は、 $4/Narrow$ の式で与えられます。つまり、Narrowness が 1 場合パルスは 4 units 広く、and if Narrowness が 4 の場合パルスは 1 unit だけ幅が広がります。

Warning

上記のすべての値は、実際の寸法を取得するオブジェクトに対応する Scale 値で乗算する必要があります。例えば、Scale Z の値が 2 で、波の高さの値が 1 であれば、最終的な波は高さ 2 BUで与えられます!

Technical Details and Hints

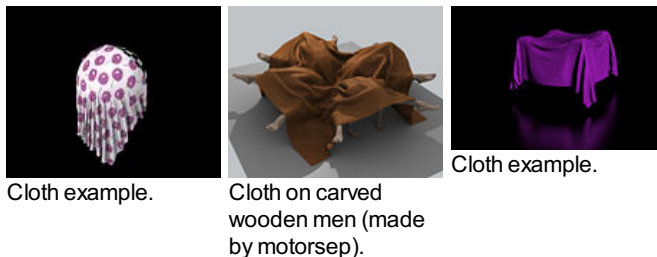
上記の値の関係をここで説明します:



Wave front characteristics.

魅力的な波の効果を得るために、海の波および正弦波に近いものにするには、次のリップルと間の距離とリップル幅を等しくします。つまりは $Narrow$ の値は $2/Width$ と等しくしなければいけません。例えば $Width=1$ であれば $Narrow$ は 2 と設定します。

クロス(Cloth)シミュレーション



クロス(布)シミュレーションはCGの表現の中でも難関の一つです。布は現実の世界ではありふれた素材のようで、実際はその内部や環境との間で非常に複雑な相互作用が働いているのです。何年もの開発の末にBlenderは非常に強力なクロスシミュレーターを手に入れ、衣服や旗の再現ができるようになりました。クロスはあなたのコントロールするオブジェクトや風や力場および他の空力モデルと関わって、その影響を受けます。

解説

クロスとして設定されれば、どんなメッシュでもクロスとして振舞います。Physicsサブコンテキストに三つのパネルで構成されたClothパネルがあります。クロスは開いたメッシュでも閉じたメッシュでもよく、また質量がゼロでもいいし、単位面積あたりの質量を持たせることもできます。

クロスは一般的にグリッドまたは立方体としてモデル化されますが、たとえばティディベアでもいいのです。しかしながら、閉じたメッシュであれば[ソフトボディ](#)を使うほうがよりうまくシミュレートできます。クロスはべらべらした布地のシミュレーションに向いています。

オブジェクトがクロスとして設定されると、クロス[モディファイア](#)がオブジェクトのモディファイアスタックに追加されます。モディファイアは他のモディファイア、たとえばArmatureやSmoothと相互作用します。この場合、メッシュの最終的な形状はモディファイアスタックにおけるモディファイアの順序に則って計算されます。たとえば、クロスを滑らかにするのならその形状が計算されたあとにしなくてはなりません。

よってクロスは二つの場所で編集することになります。F7のPhysicsボタンでクロスの様々な性質を調整し、モディファイアスタックでは画面での表示や他のモディファイアとの関係を調整します。

クロスモディファイアをApplyして、クロスをそのフレームでの形状に固定することもできます。そうするとモディファイアは削除されます。たとえば、平らなクロスをテーブルにかがせて、シミュレーションを行い、それからモディファイアをApplyします。この場合はモデリングのためにシミュレーターを使ったということになるでしょう。

シミュレーションの結果はキャッシュに保存されます。よって一度計算されたメッシュ形状のアニメーションは再計算する必要はありません。シミュレーションの設定を変えた場合は、キャッシュの消去と再計算を自由にできます。一度目のシミュレーションは完全に自動的に行われ、バイクや余計なステップは含まれません。

各フレームでのクロス形状の計算はバックグラウンドで自動的に行われるので、その計算中も作業を続けられます。しかしこれは非常にCPUを使う仕事で、あなたのコンピュータのパワーとシミュレーションの複雑さにかかっています。CPUのパワーがたくさんいる計算の場合は描画の遅れに気づくでしょう。

先回りしないように

クロスシミュレーションをセットアップし、しかしまだシミュレーションの終わりまで計算されていない場合、先のほうのフレームにジャンプするとクロス形状が正しく計算されないかもしれません。それ以前のフレームがまだ計算されていないからです。

作業の流れ

クロスに関する作業は以下のような手順になります。

1. クロスの初期形状をモデリングします。
2. PropertiesのPhysicsタブで、オブジェクトをクロスとして設定します。
3. クロスと相互作用するdeflectionオブジェクトをモデリングします。Deflectionモディファイアがモディファイアスタックの最後、つまり他のモディファイアによる変形等よりも後になるように注意してください。
4. クロスに照明をあて、マテリアルやテクスチャの設定をします。必要ならUV展開をします。
5. 必要ならパーティクルを追加します。たとえば布の表面から蒸気が出るような場合。
6. シミュレーションを実行し、満足な結果になるように設定を調整します。timelineウインドウのVCRコントローラが役に立つでしょう。
7. 場合によっては、シミュレーションのある時点でのクロス形状を初期形状として再度シミュレーションします。
8. フレームごとに細かな手直しをします。

クロスシミュレーションの生成

このセクションでは望みの結果になるようにどう設定すればいいかを紹介합니다。まず、クロスを有効にして、再現したいクロスの種類に応じてセットアップします。プリセットから手始めとなるものを選びます。

設定してみるとおわかりのように、布地が重くなるほど、硬くなり、伸びにくく、空気の影響が少なくなります。

クロスパネル

Presets

いくつかのクロスの例が用意されています。新しく付け加えることもできます。

Quality

1フレームごとのシミュレーションのステップです。数値を高くすると品質は上がりますが、計算が遅くなります。

素材

Mass

クロス素材の質量。

Structural

全体的な硬さ。

Bending

しわの係数。数値が高いほど大きなひだができます。

減衰

Spring

クロス velocities の減衰。高いほど滑らかで、あまり揺れなくなります。

Air

空気は通常、物が落下する速度を和らげる働きがあります。

ピン止め



Cloth in action.

クロスをピン止めるのにまず必要なのは [頂点グループ](#) です。これにはいくつか方法があり、ウェイトペイントを使ってピン止めたい部分にペイントするのもやり方の一つです ([ウェイトペイント](#) を参照)。

頂点グループを設定したら、あとは簡単です。ClothパネルのPinning of clothボタンを押し、使用する頂点グループとその点での硬さを設定します。

Stiffness

ターゲットの点での硬さ。数値はデフォルトの1のままで良いでしょう。

衝突

大抵の場合、クロスはたんに3D空間に垂れ下がっているだけではなく、環境内の他のオブジェクトと衝突します。適切なシミュレーションをするためにいくつか設定・注意しなければならないことがあります。


1. クロスオブジェクトをCollision(衝突シミュレーション)に参加するように設定します。
2. オプションとして(しかし推奨)、クロス自身とも衝突するように設定します。
3. 他のオブジェクトはメッシュオブジェクトでなければいけません。
4. 他のオブジェクトは移動したり、さらに他のオブジェクトの影響(アーマチュアやシェイプキーなど)で変形したりしてもかまいません。
5. 他のオブジェクトはクロスをdeflect(ゆがめる)する設定をします。
6. .blendファイルを保存してください。そうするとシミュレーションの結果も保存されます。
7. それではシミュレーションをBakeしましょう。アニメーション範囲の各フレームでの形状が計算されます。
8. その後、各フレームでの形状を編集できます。
9. 環境や他のオブジェクトを再調整して、特定のフレーム以降を再シミュレーションすることもできます。

衝突の設定

Cloth Collisions panel.

今度はクロスオブジェクトを衝突シミュレーションに参加させなければいけません。右のような Cloth Collisionパネルを探してください。

Enable Collisions

この項目を LMB  クリックすると衝突が有効になります。

Quality

シミュレーションの全体的な品質。数値を高くすると計算に時間がかかりますが、物体がクロスを突き抜けてしまったりすることが少なくなります。

Distance

他のオブジェクトがこの距離よりも近づくと、クロスを押しのけます。

Repel

衝突しそうになったときの斥力。

Repel Distance

斥力が働く最大距離。最小値よりも大きくする必要があります。

Friction

摩擦効果。メッシュオブジェクトとぶつかったときの滑りにくさです。たとえば、絹は綿よりも摩擦が低いです。

クロス自身との衝突

現実の布は自分自身をすり抜けたりはしません。よって通常は自己衝突の設定が必要でしょう。

Enable Self Collisions

これをクリックすると、クロスは自分自身を突き抜けなくなります。これはシミュレーションの時間が増えますが、結果がよりリアルになります。遠くから見た旗などにはこの設定は必要ないでしょうが、人物のケープやブラウスを近くから見るときには有効にするべきでしょう。

Quality

数値を高くすると、衝突の計算がより多く繰り返され、品質が上がります。少なくともCollision Qualityと同じかそれ以上にする必要があります。





Distance

何か問題があったらMin Distanceの数値を変えてみるというかもしれません。たいていは0.75あたりが最適ですが、動きが速い場合は1.0にしたほうがいいでしょう。0.5まで下げるのは非常に危険です(すり抜けてしまうことが多々あるでしょう)が、計算のスピードアップにもなります。

サンプルファイル: [Cloth selfcollisions](#).

共有されたレイヤー

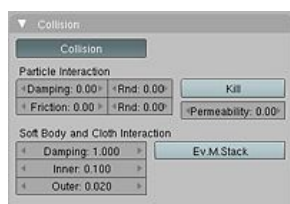
たとえば、レイヤー2と3にズボンが、レイヤー1と2に人物が存在している場合を考えてみてください。ズボンにはすでにクロスの設定がしてあります。クロスにとって人物が「見え」て要る必要があります。そうすれば、人物の足が曲がる時、その動きはクロスを押しのけます。大事なことは、他のシミュレーションと同じく、相互作用は共有されているレイヤーのみで起こるということです。この例の場合、双方のオブジェクトはレイヤー2を共有しています。

オブジェクトのレイヤーを変えるには、3D ViewのObject ModeでオブジェクトをRMB クリックして選択します。そしてMでMove Layersポップアップを表示させると、そのオブジェクトが存在しているレイヤーがわかります。オブジェクトを1つのレイヤーに移動するにはレイヤーのボタンをLMB クリックしてください。複数のレイヤーに移動するには⇧ Shift LMB で選びます。あるレイヤーからオブジェクトを取り除きたい場合は、⇧ Shift LMB で再度クリックします。

メッシュオブジェクトの衝突

衝突に使いたいのがメッシュオブジェクトではないとき、たとえばNURBSサーフィスやテキストオブジェクトの場合、メッシュに変換する必要があります。Object Modeでオブジェクトを選択し、3D viewのヘッダからObject → Convert Object Type (AltC)を選び、ポップアップメニューからMeshを選びます。

クロスとオブジェクトの衝突



Collision settings.

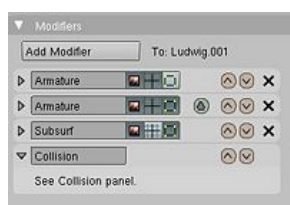
クロスは他のオブジェクトによってゆがまないといけません。クロスをゆがめるには、オブジェクトをクロスと衝突させるように設定する必要があります。衝突するオブジェクト(クロスオブジェクトのほうではありません)のdeflectionを有効にしてください。

ButtonsウィンドウのObjectコンテキスト、その中のPhysicsサブコンテキストで、右図のようなCollisionパネルを見つけてください。このパネルで設定をすると、共有しているレイヤー内の他のオブジェクト(パーティクル、ソフトボディ、クロス)に対しても衝突やdeflectionが起こることに注意してください。

注意

CollisionパネルはPhysicsサブコンテキスト内に三種類あります。一つ目はFieldsパネルのそばにあるもので、これがいまここで必要なパネルです。二つ目はSoft Bodyグループにあるタブで、これはソフトボディに関係するものです(よってクロスに関する限り見る必要はありません)。そして最後にCloth panelのそばのタブです。これは、さきほど設定に使いました。

メッシュオブジェクトのモディファイアスタック



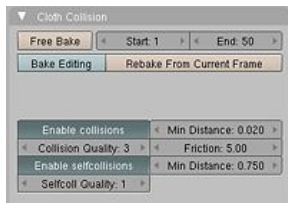
Collision stack.

オブジェクトの形状がクロスを変形させるためには、そのフレームでのオブジェクトの「本当の」形状がわからなければシミュレーションできません。この本当の形状とは基本となる形状がシェイプキーやアーマチュアによって変形された状態のもので、したがって、Collisionモディファイアはこれらのモディファイアよりもあとに位置しなければいけません。右図は人物のメッシュオブジェクトのモディファイアです(クロスオブジェクトではありません)。

クロスのキャッシュ

キャッシュの設定は他の物理シミュレーションと同様です。詳しくは[パーティクルキャッシュ](#)を見てください。

衝突をバイクする



シミュレーションにかかわるフレームでの他のオブジェクトの設定(アーマチュアによるメッシュの変形も含めて)が終わったら、衝突の計算を始められます。クロスを選択し、ObjectコンテキストのPhysicsサブコンテキストで、StartとEndを設定してシミュレーションするフレーム範囲を決め、Bakeボタンを押してください。

バイクしたシミュレーションを消去するまではStartやEndは変えられません。シミュレーションが終わると、開放、編集、再バイクのボタンが現れるでしょう。

ここですぐに気づくことがいくつかあるでしょう。まず、以前よりもバイクが非常に遅いということ、それからおそらく右図のように箱を突き抜けてしまう部分があるでしょう。

シミュレーションのキャッシュを編集する

キャッシュには各フレームでのメッシュ形状が含まれています。シミュレーションをバイクしたあとBake Editingボタンを押すと、キャッシュされた形状を編集できます。編集したいフレームに移動し、⇨ TabでEdit modeに移動します。そしてメッシュ編集ツールをなんでも使って頂点を動かしてください。Edit modeから出ると、アニメーションのフレームとしてそのメッシュ形状が記録されます。そのフレーム以降を再シミュレートしたい場合はRebake from next Frameを押してアニメーションを再生してください。Blenderは編集後のメッシュ形状を使って続きをシミュレートしてくれます。

メッシュを編集して、オブジェクトが突き抜けてしまった部分などを手直ししましょう。

メッシュの頂点を追加、消去、押し出し、取り外し、などとすると、1フレーム目から現在のフレームまでメッシュが置き換えられます。よって、Edit modeから出るときにはキャッシュの保護を解除して消去してください。そうすればちゃんとつながりのあるシミュレーションになります。

トラブルシューティング

衝突検出に関して問題にぶつかったときは、修正する方法は二つあります。

- 手っ取り早いのはCloth CollisionパネルのMin Distanceを大きくすることです。これが一番手軽に「つきぬけ」を修正できます。しかしながら、正確さは低下するので見た目はあまりよくないでしょう。このやり方をすると、クロスは空気の上にふんわりと乗っているような感じになります。
- もう一つの方法はQuality (Clothパネルにあります)を上げることです。そうするとより細かいステップでシミュレーションが行われ、すばい動きの衝突も高い精度で再現できるようになります。また、Collision Qualityを上げることもできます。衝突の計算を反復する回数が増えます。
- これらの方法で解決できないときは、キャッシュまたはバイクされた結果をEdit modeで編集できます。
- クロスがオブジェクトに切り裂かれてしまうという場合は、ClothパネルのStructStiffを非常に高く、1000くらいにしてみてください。

Subsurfモディファイア

バイクやキャッシュは異なるsubsurfレベルに応じてそれぞれ行われるので、3D Viewとレンダリング時のsubsurfレベルは同じにしておいてください。

例

クロスシミュレーションをするには、もちろん布が必要です。よってデフォルトの立方体を消去して平面を追加してください。私はY軸に沿って拡大してみました、そうしなくてもかまいません。きれいにべらべらふにやふにやとさせるにはメッシュを何度か再分割する必要があります。私は8回再分割しました。⇨ Tabを押してEdit modeに入り、W → Subdivide multiで8に設定してみました。

では、このオブジェクトをクロスにするために、Objectコンテキスト(F7) → Physicsサブコンテキストに移動しましょう。スクロールしてClothパネルを見つけ、Clothボタンを押してください。するといろいろな設定値が現れるでしょう。しかし今回はこれらのほとんどは扱いません。

これでクロスをアニメーションさせる準備はおしまいです。しかし、AltAを押してみても、あなたのかわいい布切れはまったくおもしろくない感じに落下するだけでしょ。これをなんとかするのが次の2つのセクションで説明するピン止めと衝突です。

成形のためにシミュレーションを使う

どのフレームでも、ClothモディファイアをApplyしてそのときのメッシュを固定化することができます。そして再度クロスを有効にし、最初と最後のフレームを設定してシミュレーションすることもできます。

旗を例にしてみましょう。単純なグリッドをポールにピン止めて、50フレームほどシミュレーションしてみると、旗は自然なポジションに落ち着いて止まります。その状態でクロスモディファイアをApplyします。そしてシーンの中ではためかせたくなったら再度クロスを有効にして、旗がカメラに写っている時間をシミュレーションのフレーム範囲に設定すればいいのです。

クросを滑らかにする

さて、前のセクションから続けて読まれているなら、あなたのクロスは少しカクカクしているかもしれません。それを滑らかにするには、Editingコンテキスト (`{Shortcut{f9}}`) のModifiers/パネルで、SmoothやSubsurfモディファイアを使ってください。そして、同じコンテキストでLinks and Materialsを見つけ (頂点グループのために使ったのと同じパネルです)、Set Smoothを押してください。

それではAltAを押してみてください。すばらしい結果になるはずです。

アーマチュアとクロス

アーマチュアやよって変形されるクロス、および他のオブジェクトとの衝突のサンプルファイル: [Regression blend file](#)

クросをソフトボディとして使う



Using cloth for softbodies.

クロスはソフトボディを再現することにも使えます。これが主な目的ではないのは確かですが、ちゃんと働きます。この例では標準のRubberプリセットを使っています。設定は変えず、AltAを押すだけです。

写真の例のサンプルファイル: [Using Cloth for softbodies](#)

クロスと風



Flag with wind applied.

風と衝突のシミュレーションのサンプルファイル: [Cloth flag with wind and selfcollisions.](#)

衝突 (Collision)

パーティクル、**ソフトボディ**および**クロス**はメッシュオブジェクトと衝突させることができます。また**Boids**はCollisionオブジェクトを避けようとします。

- オブジェクト同士が影響しあうには少なくとも1つのレイヤーを共有している必要があります。
- パーティクルに対する影響を特定のオブジェクトグループに制限することもできます ([Field Weights panel](#))。
- ソフトボディに対するdeflection(ゆがませる)は難しいです。しばしば突き抜けてしまいます。
- Hairパーティクルはdeflectionを無視します(しかしソフトボディとしてアニメーションさせることはできます)

オブジェクトのdeflectionの設定を変えると、パーティクル、ソフトボディ、クロスなどのシミュレーションを(Free Cacheでキャッシュを消去して)再計算しなくてはなりません。これは自動的にには行われません。選択されたすべてのオブジェクトのキャッシュを消去するにはCtrlB → Free cache selectedです。

Mode: Object Mode

Panel: Object context → Physics sub-context → Collision

オプション

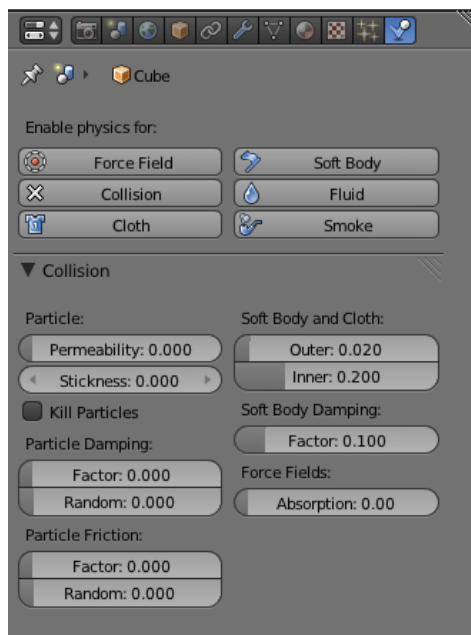


Image 1: Collision panel in the Physics sub-context.

Permeability

メッシュを通過するパーティクルのかけらの割合。IpoのPermチャンネルでアニメーションさせることもできます。

Stickiness

オブジェクトにくっつくパーティクルの量

Kill Particles

衝突時にパーティクルを消します。

Damping Factor

衝突時の減衰(パーティクルの速度とは独立しています)。

Random damping

減衰率をランダムにします。

Friction Factor

オブジェクトの表面を動くときの摩擦。

Random friction

摩擦をランダムにします。

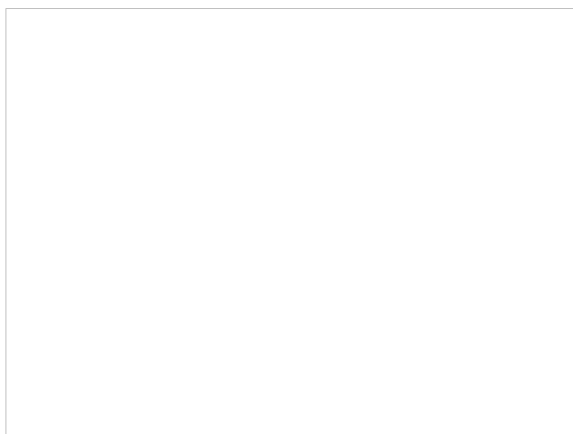


Image 1b: A softbody vertex colliding with a plane.

ソフトボディおよびクロス衝突

Outer

外側の衝突領域の大きさ。

Inner

内側の衝突領域の大きさ (詰め物の厚み)。

外側と内側は法線で決定されます。図Image 1bの水色の矢印です。

Damping Factor

衝突時の減衰。

ソフトボディの衝突を完璧に行うのは難しいです。ぶつかるオブジェクトが速すぎれば、ソフトボディを突き抜けてしまいます。[ソフトボディ](#)のセクションも見てください。

力場の相互作用

Absorption

deflectorはdeflectエフェクターに作用させることもできます。衝突またはdeflectorオブジェクトを設定して特定のエフェクターの力場をゆがませることができます。Absorptionの値を調整してください。100%にすると力はそのオブジェクトをまったく通り抜けなくなります。たとえばこの値を10%、43%、3%に設定したオブジェクトが縦に並んでいた場合、最終的に通り抜ける力は約50%($100 \times (1-0.1) \times (1-0.43) \times (1-0.03)$)になります。

例



Image 2: Deflected Particles.

これは下向きに放射しているパーティクルシステムにdupliverされたMetaオブジェクトで、立方体メッシュによってゆがめられています。

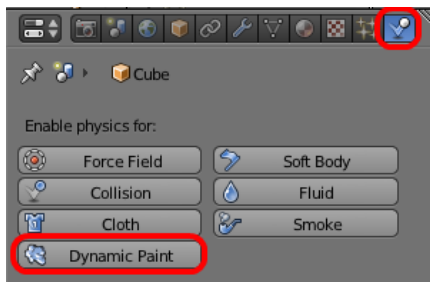
ヒント

- メッシュの法線が正しい側に向いているように気をつけてください。
- Hairパーティクルは力場に直接的に反応します。よって近い距離で力場を使うなら必ずしも衝突シミュレーションは必要ありません。
- Particle modelにおいて、Hairパーティクルはそれ自体を放射しているメッシュを避けようとしています。よって、少なくともモデリングに関しては衝突のような作用を利用できるということです。

ダイナミックペイント

ダイナミックペイント(Dynamic paint)は、オブジェクトをペイントキャンバスやブラシやブラシに変え、頂点カラーや連番画像、ディスプレイメントを作成できる新しいモディファイア & 物理システムです。これにより従来は難しかったたくさんのエフェクトが可能になります。例えば、雪の中の足跡、地面を濡らす雨だれ、壁に貼りつくペイント、徐々に凍っていくオブジェクトなど。

下記のガイドでは、ダイナミックペイントのユーザーインターフェイスと機能全般について非常に基本的なことを説明しています。



ダイナミックペイントの起動

モディファイアの追加

ダイナミックペイントは、プロパティエディタの "Physics" タブから利用できます。

タイプ

モディファイアには二つの違うタイプがあります。

[Canvas](#)

オブジェクトがダイナミックペイントの Brush からペイントされるようになります。

[Brush](#)

オブジェクトが Canvas にペイントするようになります。

注意

BrushとCanvasを同時にONにすることもできます。この場合、同じオブジェクトの "Brush" がその "Canvas" に影響を与える事はありませんが、シーン内の他のオブジェクトに相互作用できます。

その他役に立つページ

- [A step-by step introduction\(英文\)](#)
- [A detailed guide that covers every setting with images and examples\(英文\)](#) (現在更新停止中)

流体 (Fluid) シミュレーション

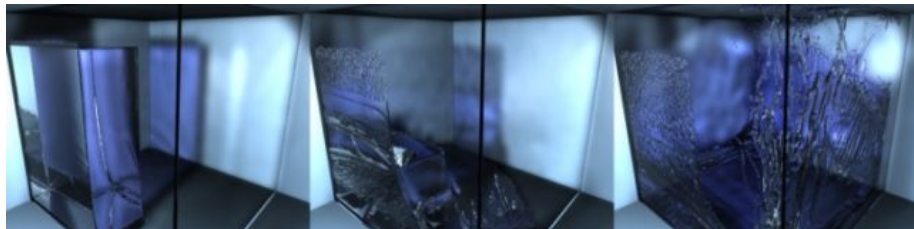
Mode: Object mode / Edit mode (Mesh)

Panel: Physics sub-context → Fluid

解説

Blenderでシーンを制作するとき、指定したオブジェクトを流体シミュレーションに参加させることができます。たとえば流体や障害物としてです。また別のオブジェクトのバウンディングボックスはシミュレーションの行われる領域として使われます (これはシミュレーションドメインと呼ばれます)。シミュレーション全体にかかわるパラメータ (たとえば速度や重力) もこのドメインオブジェクトに対して設定できます。

バイク (BAKE) ボタンを押すと、形状やセッティングの情報がシミュレーターに伝えられ、流体シミュレーションが実行されます。そして各フレームにおけるメッシュとそのプレビューが生成され、ディスクに保存されます。その後、表示しているフレームに応じて適切なデータを読み込まれ、表示またはレンダリングできます。



A breaking dam.

作業の流れ

基本的に、以下のような手順を踏みます。

- [シミュレーションドメイン](#) (シーンの中で流体の存在する領域) を設定する。
- [流体となるオブジェクト](#) を設定し、マテリアル、粘度、初期速度を指定する。
- 場合により、[流体をコントロールするオブジェクト](#) を設定する (発生源や吸収源)。
- 場合により、流体と関係するオブジェクトを設定する。
 - [障害物](#)
 - 流体に浮かぶ [パーティクル](#)
 - 流体を目的の形状に変形させる [コントロール](#)
- 場合により、[流体のプロパティをアニメートする](#)
- [シミュレーションをバイクする](#) (必要に応じて手直しし、再度バイクする)。



ドメインオブジェクトにバイクされます !

流体シミュレーションを計算すると、シミュレーションはドメインオブジェクトに対してバイクされます。

したがって:

- バイクに関するオプションはドメインオブジェクトが選択されているときのみ表示されます。
- バイクに関するオプションはドメインのページの [バイクのセクション](#) で説明されています。

さらに詳しく

Blenderの流体シミュレーションについてより詳しく知るには、以下のページを参照してください。

- シミュレーションについての [ヒント](#)。
- よりリアルなシミュレーションを行うのに役立つ [技術的詳細](#)。
- 制限や問題の解決法を知るには [付録](#)。また追加のリンクも紹介されています。

オーシャンシミュレーション

Blender のオーシャンシミュレーション (ocean simulation) ツールはモディファイアの形を取り、変形する海の表面のシミュレーションと生成、そしてシミュレーションデータのレンダリングに使用される関連テクスチャの作成を行います。オープンソースの Houdini Ocean Toolkit から移植されており、沖の波と泡をシミュレートすることを目的としています。

シミュレーション内部

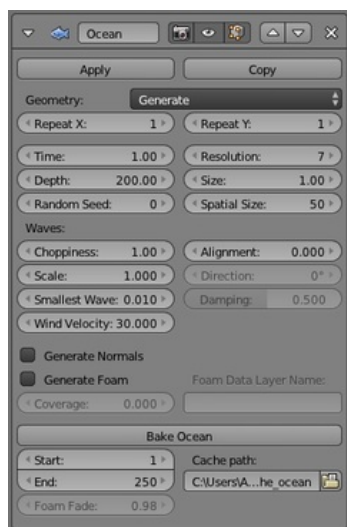
シミュレータ自身にはFFT法を使用し、2Dテクスチャマップに非常によく似たシミュレーション情報の2Dグリッドを内部的に生成しています。シミュレータはディスプレイメント、ノーマル、波頭をの交差(いわゆる泡)を計算するのに使用される特別なデータの3つのタイプのデータを生成可能です。シミュレーション後、これらのマップは海の表面の形状の3Dディスプレイに使用され、Ocean Textureを通じてシェーディングすることもできます。内部シミュレーションエンジンは OpenMP によりマルチスレッド化されており、マルチコアの恩恵を受けられます。

Ocean モディファイア

モード: Object mode

パネル: Modifiers context

解説



Ocean Modifier Panel

Ocean モディファイアは Blender 内でシミュレーションを行うのにメインとなる場所です。モディファイアにはシミュレーションデータが格納され、変形した海の表面 Mesh を作成するのに利用します。また、Ocean モディファイアは泡 (Foam) マップを視覚化するための頂点カラーチャンネルを追加することもできます。

Geometry オプション

Geometry

Ocean モディファイアが Mesh のジオメトリに影響を及ぼす方法:

- **Generate:** シミュレーションデータと正確に一致する、タイル化した Mesh グリッドを生成します。

Mesh サーフェスを生成した時、既存の Mesh オブジェクトはこのオーシャングリッドに完全にオーバーライドされます。UV チャンネルも追加され、UV 空間の[0,0,1,0]をシミュレーショングリッドにマップします。


- **Displace:** 既存の任意のトポロジーの Mesh をディスプレイします

Repeat X, Repeat Y

Mesh サーフェスを生成した時、そのグリッドをX軸とY軸方向にタイル化する回数。これらのタイル化された Mesh 領域の UV は、UV空間の [0,1]より外でも維持されます。

シミュレータのオプション

Time

オーシャンサーフェスが評価される時間。アニメーションするオーシャンを作成するには、キーフレームを挿入 (RMB ) し、この Time 値を動かす必要があります。Time 値が変化するスピードは波のアニメーションのスピードを決めます。

Resolution

シミュレーションエンジンにおける品質 VS スピードを主にコントロールするもので、シミュレーションにより生成される内部2Dグリッドの解像度を決めます。内部グリッドはこの Resolution 値の二乗であるため、Resolution値が16の場合、作成されるシミュレーションデータサイズは256×256となります。高い Resolution値では計算が遅くなりますが、利用可能なディテールが細かくなります。

注意: Geometryオプションで 'Generate' を選択した場合、この Resolution 値は内部シミュレーションデータの解像度に相当する、生成 Mesh サーフェスの解像度も決定します。

Spatial Size

シミュレーションが行われるオーシャンサーフェス領域の幅(メートル)。また、これは生成される Mesh もしくはディスプレイ領域のサイズ

(Blender単位)も決定します。もちろん、Oceanモディファイアの付いたオブジェクトを Object Mode でスケールし、シーン内の外見上のサイズを調整することもできます。

Depth

シミュレーションする領域の下の、海底までの深さ

Wave オプション

Choppiness

波の頂上の波立ち。Choppinessが0で、オーシャンサーフェスはZ軸方向のみ上下に移動しますが、Choppinessが高い場合、波はX軸とY軸の横方向にも移動し、波の頂上を鋭くします。

Scale

波の振幅の全体的な大きさをコントロールします。0より上もしくは下で、波のおおよその高さもしくは深さになります。単に Ocean オブジェクトのZ軸方向をスケールするだけではなく、シミュレーションの全ての面や、X軸とY軸の移動量、付随する泡とノーマルもスケールします。

Alignment

風によって生じる波の形状の方向。0で風と波がランダムで方向も均一になります。高い Alignment 値で風がもっと一定の方向に吹くようになり、波も一つの方向に圧縮され、整列して現れるようになります。

Direction

Alignment使用時、波が整列する方向を角度で指定します。

Damping

Alignment使用時、内部反射する波が弱まる量です。波の動きの指向性を高くする効果があります(波の形状ではありません)。Damping が0.0で各波は全ての方向で互いに反射しあい、Damping が1.0でこれらの内部反射する波は消滅し、風の方向に移動する波のみが残ります。

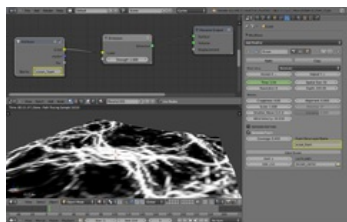
Smallest Wave

生成する波の最小サイズ。動作はローパスフィルタに似ており、高周期の波のディテールが失われます。

Wind Velocity

風のスピード(m/秒)。低い速度だと、小さな表面の波に限定されます。

Sim Data Generation オプション



Foam Vertex Color を名前を付けたデータレイヤとともに使用

デフォルトでは、シミュレータはディスプレイメントデータのみを生成します。最小の作業量で最速のフィードバックが得られるからです。さらに追加でレンダリング用のシミュレーションデータを生成できます。

Generate Normals

追加のノーマルマップデータをシミュレートします。これはノーマルにマップした時、バンプマップとしてオーシャンテクスチャから使用でき、さらにベイキング時にはノーマルマップの連番画像を生成可能です。

Generate Foam

追加の泡 (Foam) データをシミュレートします。これは

オーシャンテクスチャをテクスチャリングに(恐らくマスクとして)使用する際に取り出すことができます。また、ベイキング時には連番画像の泡マップの生成が可能になります。

Coverage

泡が波を覆う量を調整します。負の値で泡の量が減少し(頂上部分のみ残り)、正の値で追加されます。

Foam Data Layer Name

頂点データレイヤの名前で、Ocean モディファイアにより、泡マップを頂点カラーとして格納するのに使用されます。これはレンダー内で泡データにアクセスするのに必要です。

ベイキング

オーシャンデータを毎回シミュレートする代わりに、オーシャンデータをディスクにバイク(Bake)します。シミュレーションがバイクされた時、シミュレータエンジンは完全にスキップされ、モディファイア/テクスチャはすべての情報をバイクされたファイルから取り出します。

ベイキングは以下の理由により有利です。

- データを再計算するより格納したデータを使用の方が早い
- オーシャンデータを外部レンダーでレンダリングできる
- もっと上級の泡マップが可能

データファイル

シミュレーションデータはディスク内に連番の OpenEXR 画像マップとして格納され、それぞれディスプレイメント用、ノーマル用、泡用(生成された時のみ)になります。バイクされたファイルからのデータ読み込みでは、バイク連番画像のフレームの一つがディスクから読み込まれた時、メモリにキャッシュされます。つまり、読み込み済フレームへのアクセスはディスクアクセスのオーバーヘッドがなく、高速だということです。

これらのバイクされたファイルは素の OpenEXR であるため、対応している他のアプリケーションやレンダラで開いたり、レンダリングすることもできます。

泡のベイキング

また、ベイキングでは上級の泡機能が使用できます。その場でシミュレートする場合、オーシャンシミュレータは現在のフレームからのみデータを取り出します。泡マップの場合、これは与えられたフレームの波の頂上の先端を表現します。現実では、後の泡は波の相互作用により生成され、波の表面のてっぺんに少しの間残ってから消えます。ベイキングでは、前回のフレームからの泡を累積し、表面上に残しておくことで、この挙動を近似することができます。

[\[video link\]](#)

Baking オプション

Start, End

シミュレーションをバイクするフレーム(指定フレーム含む)。バイクされた範囲外のフレームにアクセスした場合、バイクの Start と End フレームが繰り返されます。

Cache Path

バイクされた EXR ファイルを格納するフォルダ。連番画像は disp_####.exr、normal_####.exr、foam_####.exr の形式になり、####の部分には4ケタのフレーム番号が入ります。もし Cache Path に指定されたフォルダが存在しない場合は作成されます。

[\[video link\]](#)

Blender で画像マップをシミュレートとバイクし、3Delightでレンダリング。

来歴

コアのシミュレータは Drew Whitehouse氏により、[Houdini Ocean Toolkit](#)用に開発されたものです。これが Hamed Zaghaghi氏により、C に移植され、短編映画の Lighthouse 制作中、ProMotion Studios/[Red Cartel](#) の支援により、Blender 2.4シリーズ用のパッチに統合されたものです。

この作業では、Matt Ebb がそのコアシミュレータを Blender 2.5に再統合し、機能の追加、修正、最適化を、'[Save the Ocean Sim](#)' プロジェクトの支援を受けて行いました。

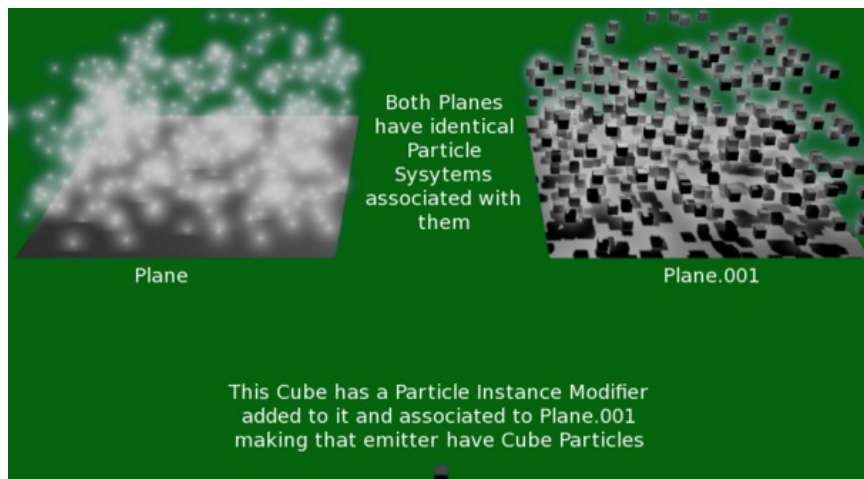
パーティクルインスタンス(Particle Instance)モディファイア

Mode: Any mode

Panel: Modifiers (Editing context, F9)

解説

オブジェクトにパーティクルインスタンスモディファイアを追加すると、そのオブジェクトはパーティクルシステムを使っているオブジェクトのパーティクル形状とすることができます。つまり、このモディファイアを使うには、パーティクルシステムを使っている他のオブジェクトが必要です。さもなくばパーティクルインスタンスモディファイアは一見何もしていないように見えるでしょう。



Particle system on left has no ParticleInstance modified object associated with it. The one on the right is associated with cube shown by using a ParticleInstance modifier on the cube.

概要

パーティクルおよびパーティクルインスタンスに関連する用語を簡単に説明します。

- パーティクルシステム(Particle system) : パーティクルを放射(emit) / 生成(generate)するように設定されたオブジェクト(メッシュ)。
- 通常パーティクル(Normal particle) : 子パーティクルではないパーティクル。
- 子パーティクル(Children/child particle) : すでに存在している通常パーティクルによって生み出されたパーティクル。基本的に子パーティクルのほうが計算するのは速いです。
- 胎児パーティクル(Unborn particle) : まだ放射される時間ではないので、放射されていないパーティクル。たとえば放射されるフレームより前のフレームではこの状態です。
- 生存パーティクル(Alive particle) : 放射されていて、死んでいる状態ではないパーティクル。たとえば寿命が来る前の状態です。
- 死亡パーティクル(Dead particle) : 放射され、その後寿命に達して、死んでいる状態に入ったパーティクル。

オプション

パーティクルインスタンスモディファイアは、他のオブジェクトのパーティクルシステムと相互に依存しているため、生成される効果の見た目や振る舞いはパーティクルシステムの設定によって大きく変わります。パーティクルインスタンスモディファイアを使っていて思ったとおりの結果が得られない場合は、このモディファイアだけではなくパーティクルシステムの設定もチェックしてみてください。

Object

ここにオブジェクト名を指定して、パーティクルインスタンスモディファイアと他のオブジェクト(ふつうはパーティクルシステムを持っているオブジェクト)を関連付けます。つまり、ここで指定したオブジェクトがパーティクルを放射している場合、そのパーティクルの形状はパーティクルインスタンスモディファイアが適用されているメッシュの現在の形状になります。

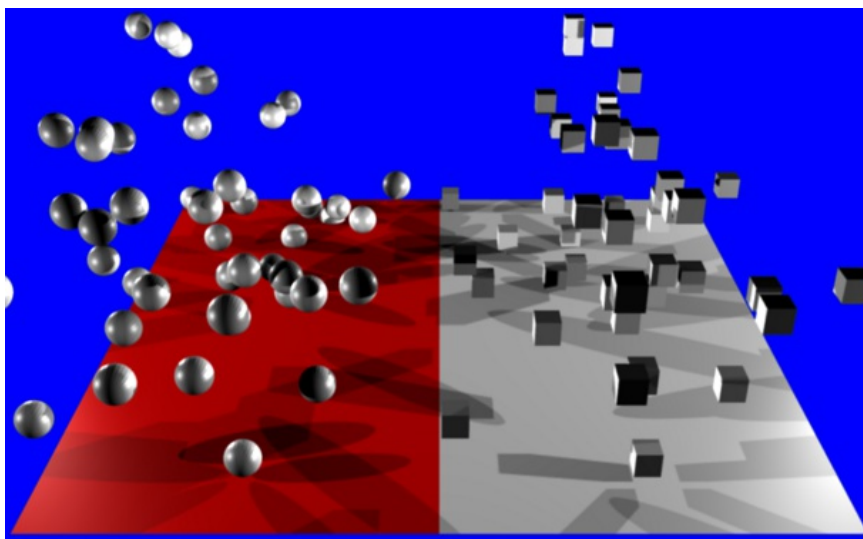
たとえば球体がパーティクルインスタンスモディファイアを付加されていて、この欄に指定した別のオブジェクトがパーティクルを放射しているなら、そのパーティクルは球形になります。

たいていはこの欄にはパーティクルシステムを持ったオブジェクトの名前を指定しますが、例外のオブジェクトが指定できないわけではありません。オブジェクトがパーティクルシステムを持っているかいないかはチェックされません。

Particle System

この欄は、Objectで指定したオブジェクトが複数のパーティクルシステムを持っているときに、目的のシステムの番号を指定します。数値は1から10までです。その範囲のどんな番号でも指定できますが、その番号が有効かどうかはチェックされないので注意してください。存在しないパーティクルシステムの番号を指定した場合は何の変化も起こらず、警告も出ません。

以下の画像は、赤と白の二つのエリアそれぞれにパーティクルシステムを設定している平面メッシュです。左側は球体にパーティクルインスタンスモディファイアを適用している場合、右側は立方体にパーティクルインスタンスモディファイアを適用している場合です。



Render showing a single Plain mesh object assigned to two different vertex groups and each of those vertex groups is assigned a separate and independent particle system, with each particle system being assigned a different ParticleInstance modifier. In the case shown the ParticleInstance modifiers are a sphere and a cube.

[Example Blend file](#)

作成

Normal

有効にすると、パーティクルインスタンスモディファイアを適用されているオブジェクトの形状が、通常パーティクルの存在している位置に描画されます。つまりオブジェクトが球形なら、通常パーティクルが球形になります。

Children

有効にすると、パーティクルインスタンスモディファイアを適用されているオブジェクトの形状が、子パーティクルの存在している位置に描画されます。つまりオブジェクトが球形なら、子パーティクルが球形になります。

Size

インスタンスオブジェクトのサイズを変更します。これを無効にすると、パーティクルはすべて元のオブジェクトのサイズと同じになります。

表示

Unborn

有効にすると、パーティクルインスタンスモディファイアを適用されているオブジェクトの形状が、胎児パーティクルが放射／使用される予定の位置に描画されます。つまりオブジェクトが球形なら、胎児パーティクルが球形になります。

Alive

有効にすると、パーティクルインスタンスモディファイアを適用されているオブジェクトの形状が、生存パーティクルが放射／使用される予定の位置に描画されます。つまりオブジェクトが球形なら、生存パーティクルが球形になります。

Dead

有効にすると、パーティクルインスタンスモディファイアを適用されているオブジェクトの形状が、パーティクルが死亡する予定の位置に描画されません。つまりオブジェクトが球形なら、死亡パーティクルが球形になります。

パスを使う

Create Along Paths

これを有効にすると、パーティクルのパス(軌道)やヘアパーティクルの繊維に沿うようにオブジェクトが変形されます。たとえば、以下の画像では一つのパーティクルが1から4までの通過点(ターゲットのパーティクルシステム)を通過して動き、4で死亡します。

X,Y,X Rotation Axis

パスに沿わせる軸を指定します。

Keep Shape

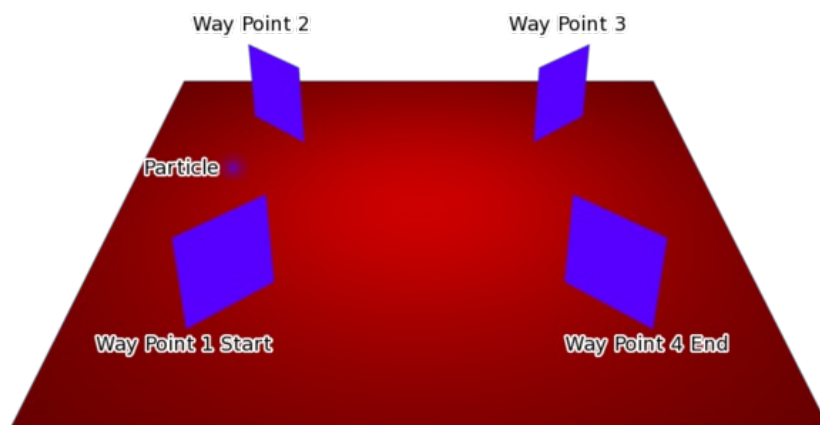
オブジェクトが変形するのを防ぎます。単純にオブジェクトの中心からパスの終端方向に対して整列されます。

Position

パスの何%をオブジェクトが占めるかを指定します。この値をアニメーションさせると、成長していくような効果などが表現できます。

Random

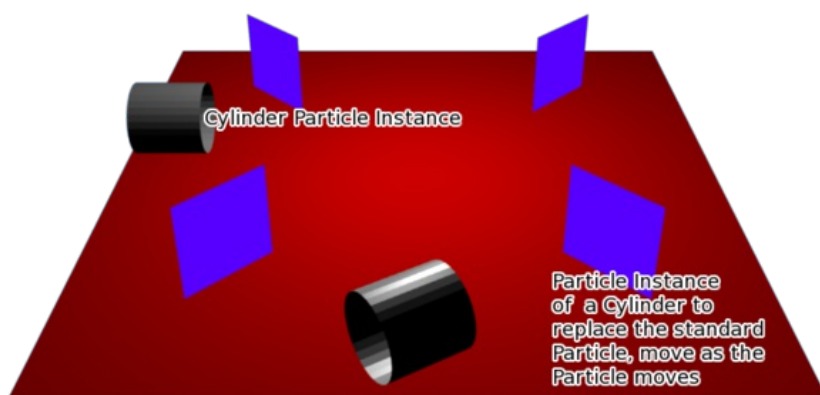
Positionの値をランダムにする度合い。



Keyed particle following way points (showing one particle).

[Example Blend file](#)

パーティクルインスタンスモディファイアが円柱に付加され、通過点のパーティクルシステムに関連付けられると、パーティクルの位置が円柱にコピーされ、円柱はパーティクルの位置についていきます。その過程で円柱の他のプロパティは変わらず、位置だけが変わります。形状や回転は変わりません。下の図を見てください。



Keyed particle following way points showing a mesh object (ParticleInstance modifier) in place of the original particle.

[Example Blend file](#)

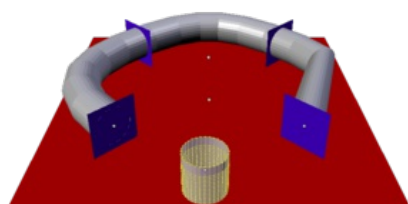
上の二つの画像ではパーティクルインスタンスモディファイアのPathボタンは無効でしたが、これを有効にすると下図のような効果が現れるでしょう。



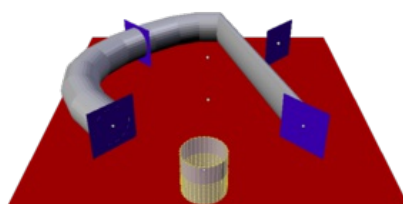
Keyed particle following way points showing a mesh object (ParticleInstance modifier) in place of the original particle, that is also being deformed to fit the travel path of the original particle.

[Example Blend file](#)

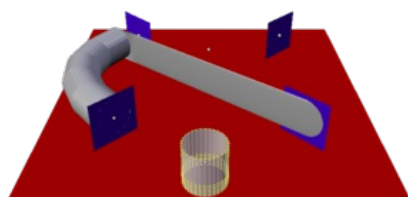
円柱の位置がパーティクルを追跡するのではなく、円柱のメッシュそのものがパスにフィットしようとしています。メッシュ形状のジオメトリは、変形がうまく行われるかどうかに影響します。今回の場合、円柱はたくさんループカットをしてあるので、パスに沿ってきれいに曲げられます。たとえば、円柱のループカットを減らしていくと、変形の結果は以下のように変わります。



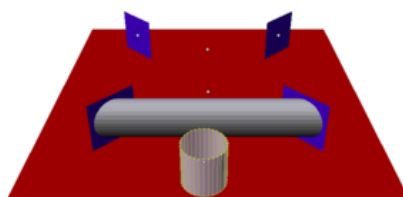
The cylinder has most of its edge loops so most of the path deform is very regular apart from at the very end of the curve.



The cylinder has some of its edge loops removed so the path of the deform starts to become less regular.



Now the deform path is very rough.



At this point there aren't any vertices to bend the cylinder to follow the path, and instead the cylinder just goes directly to the last way point 4.

すべての追加ループを削除すると、円柱の上部と下部の頂点だけが残りに、曲がるための頂点がないということになります。この場合はパスにきれいに沿うことができません。よって通過点1から通過点4に直行してしまいます。パーティクルインスタンスモディファイアのPathボタンはkeyedパーティクルだけでなくヘアパーティクルに対しても使えます。この場合はヘアパーティクルの繊維 (strand) に沿ってオブジェクトが変形します。以下の画像はヘアに対してPathボタンを使った効果です。



Strand with a ParticleInstance modifier associated with it and deforming the cylinder along the hair profile.

[Example Blend file](#)

Note

Strands when they are generated instantly die when created so for the Path button to be of any use, you must also have the Dead button activated. Otherwise the path a mesh took will not be visible!

関連項目

- [パーティクル](#)

パーティクル(Particles)

パーティクルはオブジェクトメッシュから放射される多量の粒子です。典型的には数千という数になります。個々のパーティクルは光の点であったりメッシュであったり、つながれていたり自由に飛び跳ねたりします。パーティクルは様々な影響や力に反応し、また寿命という概念があります。自由に動き回るパーティクルは炎、煙、霧、そして塵や魔法の力などの表現に役立ちます。

[ヘア](#) (Hair)タイプのパーティクルは通常のパーティクルを特殊化したものです。ヘアは繊維 (strands) から成り立っていて、髪や動物の毛、草などを表現できます。

パーティクルはモディファイアのパネルとしても存在しますが、すべての設定はObjectコンテキストのParticleサブコンテキストで行います。

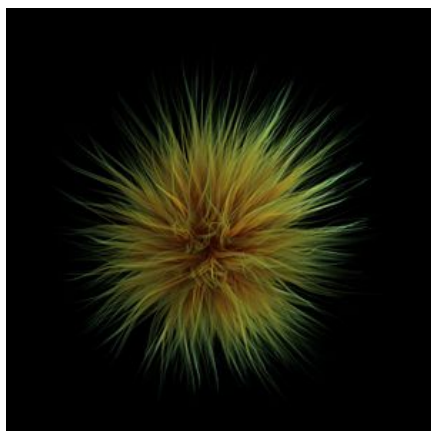


Image 1: Some fur made from particles ([Blend file](#)).

パーティクルは基本的にメッシュ空間に放射されます。その動きは以下のような様々な要素に左右されます。

- メッシュから放出されるとき初期速度。
- エミッター (パーティクルを放射するオブジェクト) の動き。
- 重力や空気抵抗。
- 風、渦、カーブに沿ったガイドなどの力場。
- 他のオブジェクトとの衝突などの相互作用。
- 他のパーティクルとの引力や反発力。
- ソフトボディによる動き (ヘアパーティクルのみ)。
- [ラティス](#)による手動操作。

パーティクルは以下のような形態でレンダリングできます。

- [Halos](#) (炎、煙、雲などに)。
- [メッシュ](#)。そのメッシュそのものをアニメーションすることもできます (魚、蜂など)。この場合、パーティクルはオブジェクトを運ぶ役目ということになります。
- [繊維 \(strands\)](#) (髪、毛、草)。パーティクルの動く道筋が繊維として見えます。この繊維は3D Windowで操作できます (櫛でとかしたり、追加、カット、移動したり)。

どのオブジェクトも、いくつものパーティクルシステムを持つことができます。そしてそれぞれのパーティクルシステムは最大100,000個のパーティクルを含むことができます。パーティクルのタイプによっては (ヘアおよびkeyed) それぞれが最大10,000個の子パーティクルを持つことができます (そのそれぞれが親と似たような動き、放射をします)。具体的な限界はコンピュータのメモリとあなたの忍耐によって決まります。

以前のバージョンとの非互換性

Blender 2.45までで使われていたパーティクルシステムと新しいシステムでは多くの違いがあります。旧システムではできなかったいろいろな事ができるようになりました。新システムは旧システムとは互換性がなく、Blenderは旧システムを変換しようとはしますが完全に上手くいくわけではありません。旧システムは新システムでのエミッターシステム (後述) に似ています。もしもBlender 2.45またはそれ以前のバージョンを使っている場合は、[旧ドキュメントを参照してください](#)。

作業の流れ

通常のパーティクルを使うには以下のような手順を踏みます。

1. パーティクルを放射するメッシュを作成する。
2. そのメッシュに対してパーティクルシステムを1つまたはそれ以上設定する。望みの効果を得るために、複数のパーティクルシステムを相互作用または合成しなければならないことはよくあります。
3. それぞれのパーティクルシステムの設定を整える。
4. ベースメッシュや、シーンに含まれるほかのパーティクルメッシュをアニメートする。
5. パーティクルの軌跡や流れを成形する。
6. ヘアパーティクルの場合はカットしたり櫛でとかしたりして整える。
7. 最終的なレンダリングやシミュレーションを行い、必要に応じて手直します。

パーティクルシステムの作成

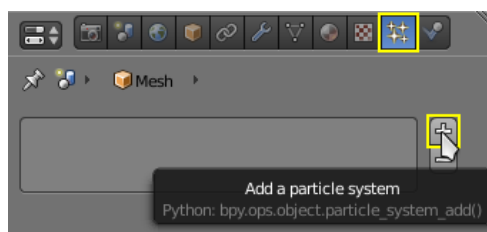


Image 2: Adding a particle system.

オブジェクトに新しいパーティクルシステムを追加するには、Object Settings EditorのParticlesタブで +ボタンを押します。一つのオブジェクトは複数のパーティクルシステムを持つことができます。

それぞれのパーティクルシステムは互いに独立した設定を持っています。それを共通させることもできるので、手作業で設定値をコピーしなくてもすみます。また、複数のオブジェクト間で設定値を共有することもできます。Randomの項目を使ってそれぞれの見た目を少しずつ変えれば、単純なコピーには見えなくなります。

パーティクルシステムの種類

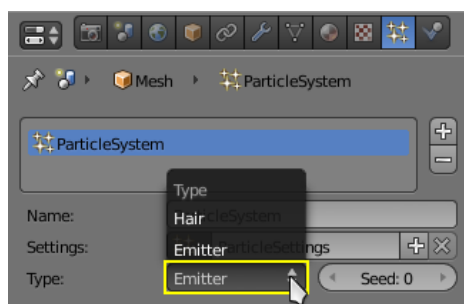


Image 3: Particle system types.

パーティクルシステムを作成したら、Propertyウィンドウにたくさんのパネルやボタンが現れます。しかしあわてないように！ パーティクルシステムは2つのタイプがあり、Typeのドロップダウンリストで切り替えられます。

エミッター (Emitter)

旧パーティクルシステムをさらに拡張したようなものです。設定した開始フレームから終了フレームまでの間、オブジェクトの表面からパーティクルが放射され、それぞれのパーティクルは寿命を持っています。

ヘア (Hair)

このタイプのパーティクルは繊維 (strand) としてレンダリングされ、非常に独特の特徴を持っています。3D Windowでリアルタイムに編集したり、クロスシミュレーションを併用してアニメートしたりできます。

パーティクルシステムのタイプによって設定項目は部分的に異なります。たとえばImage 3ではエミッターだけの設定が表示されています。

共通オプション

それぞれのシステムは共通したコントロールセクションを持っています。ただしその中の項目はシステムによって変わります。

放射 (Emission)	パーティクルの初期状態やシーンへの表れ方を設定します。
キャッシュ (Cache)	リアルタイムな反応性を上げるため、また無駄な再計算を避けるため、パーティクルのデータをメモリ内にキャッシュ、またはディスクに蓄積できます。
速度 (Velocity)	パーティクルの初期速度。
回転 (Rotation)	パーティクルの回転に関する設定。
物理 (Physics)	パーティクルの動き方の種類を決めます。
レンダー (Render)	レンダリングの設定。
表示 (Display)	3D Windowでの表示。
子 (Children)	子パーティクルの生成に関するコントロール。
フィールドウェイト (Field Weights)	外部からの力に関する設定。
力場 (Force Field Settings)	パーティクルを力場にします。
頂点グループ (Vertex Groups)	頂点グループによって様々な影響を与えます。

リンク

- [Tutorials](#)
- [Physics Caching and Baking](#)
- [Particle Rewrite Documentation](#)
- [Thoughts about the particle rewrite code](#)
- [Static Particle Fur Library](#)

煙 (smoke) シミュレーション

開発ノート

Blenderの新しい煙シミュレーションは論文'[turb.php|Wavelet Turbulence for Fluid Simulation](#)'および付随するサンプルコードに基づいています。BlenderにはDaniel Genrichによって実装され、現在はJanne Karhulによってメンテナンスされています。

内部処理

このシミュレーターはボリュメトリック流体ベースモデルを使い、結果はボクセルグリッドとして出力されます。このボクセルデータはBlenderの3Dビュー内でカスタムOpenGLシェーディングによってインタラクティブに表示され、ボクセルデータテクスチャを使ってレンダリングできます。Blenderの煙シミュレーションは[パーティクル](#)のまわりのボクセルをラップします。そのためには、パーティクルを放射するオブジェクトと、煙がレンダリングされる領域を囲む「ドメイン」オブジェクトが必要となります。

Note

ドキュメントのこのパートはバージョン2.58を使っています。

作業の流れ

煙シミュレーションは流体シミュレーションに似ています。シミュレーションにはドメイン(Domain 領域)とフロー(Flow 流れ)オブジェクトが必要です。

- [ドメイン](#)にするオブジェクトを作成します。シミュレーションの領域を規定するメッシュです。
- [フロー](#)にするオブジェクトを作成します。煙を生成する領域を規定するものです。
- [衝突](#)オブジェクトをセットします。シーンの中で煙とかわるオブジェクトです。
- 煙に[マテリアル](#)を適用する。
- シミュレーションを[バイク](#)する。

何か困ったことがあれば[付録](#)を参照してください。

柔体 (Soft Body) シミュレーション



Image 1a: A softbody cloth uncovering a text. [Animation](#) – [Blend file](#)

柔体 (Soft Body) は、柔らかい、または変形する物体のシミュレーションです。Blenderにおいて、このシステムは布でできた物および閉じたメッシュに適しています。べらべらした布切れに対しては、別のエンジンを使った専用の[クロス](#)シミュレーションがより適しています。

シミュレーションはオブジェクトの頂点や制御点に対して力を作用させることで行われます。重力や力場といった外部の力、および頂点同士をつなぎとめようとする内部の力です。このようにして、何かで満たされた物体が力に反応した場合にとる形状をリアルに再現できるのです。

柔体は他のオブジェクトと衝突させることができます (*Collision*)。自分自身と衝突することもできます (*Self Collision*)。

柔体シミュレーションの結果は静的なオブジェクトに変換できます。また、シミュレーションの結果を編集 (*bake edit*) することもできます。たとえば途中のあたりの結果を編集して、そこからまたシミュレーションをすることもできます。

典型的な使い方



Image 1b: A wind cone. The cone is a Soft Body, as the suspension. [Animation](#) – [Blend file](#)

柔体は以下のようなものに適しています。

- 伸び縮みする物体。衝突をする場合もしない場合も。
- 力に反応する旗や布切れ。
- ある種のモデリング作業。たとえばクッションやテーブルクロスを作るとき。
- Blenderには衣服をシミュレートするのに適した別のシステム ([クロス](#)を参照) がありますが、幅の広い袖などの特定の部分には柔体を使うこともできます。
- 髪 (衝突を最小限にすれば)。
- 揺れるロープ、鎖などのアニメーション。

以下のビデオを見れば使い道のアイデアがもっとわかかもしれません: [\[1\]](#), [\[2\]](#)

柔体の作成

柔体シミュレーションは頂点または制御点をもつすべての種類のオブジェクトで行えます。

- メッシュ
- カーブ
- サーフィス
- ラティス

オブジェクトを柔体としてシミュレーションを起動するには:

- PropertiesウィンドウのPhysicsタブ (一番右の、ボールが弾んでいるアイコン) に移動します。
- Soft Bodyのボタンを押します。

するとたくさんの設定項目が現れます。すべての項目のリファレンスは[ここを参照](#)。

- シミュレーションはAltAで始められます。
- Spaceでポーズ、AltAで再開。
- Escで中止します。

シミュレーション品質

Soft Body Solverパネルでシミュレーションの正確さを設定できます。

Min Step

各フレームに対するシミュレーションの最小ステップ数。動きの速い衝突オブジェクトが柔体をすり抜けてしまうようなときは数値を大きくしてください。

Max Step

各フレームに対するシミュレーションの最大ステップ数。通常、シミュレーションのステップ数は(Error Limitに基づいて)動的に変わりますが、この値を変えたほうが良い場合もあります。

Auto-Step

ステップサイズを自動的に変えます。

Error Limit

シミュレーションの全体的な品質。デフォルトは0.1です。これは衝突の作用をどのくらい正確にチェックするかという最も重要な項目です。まずは平均的な辺の長さの半分の値にしてください。そして目に見える不都合、乱れ、大げさすぎる反応などが発生した場合は、値を下げます。Error Limitが同じなら、シミュレーションはどんなときでも大体同じくらいの「ひどさ」で行われます。値を下げると、より最適なステップサイズになります。

Fuzzy

シミュレーションの計算が速くなりますが、正確さは下がります。

Choke

且衝突オブジェクトのメッシュ内に進入した頂点や辺を落ち着かせます(出て行く速度を下げます)。

Diagnostics

Print Performance to Console

シミュレーションの動作をコンソールに出力します。

Estimate Matrix

行列を推定してCOM、ROT、SCALEに分離します。

キャッシュとバイク

柔体や他の物理シミュレーションは統一的なキャッシュおよびバイクのシステムを持っています。詳しくは[パーティクルキャッシュ \(eng\)](#)を参照してください。

アニメーションをプレビューすると、シミュレーションの結果は自動的にディスクにキャッシュされ、次にアニメーションするときには保存したデータを読み取ってすばやく動作します。シミュレーションをバイク(Bake)した場合はそのキャッシュは保護され、再計算が必要になるような変更を加える際はたずねられます。



開始時間と終了時間に注意

シミュレーションはBakeパネルのStartとEndで指定したフレームの範囲だけで行われます。バイク以外のときもです！ よってデフォルトの250フレームよりも長くシミュレーションしたい場合はEndの値を変える必要があります。

- キャッシュ:
 - アニメーションが始まると、それぞれの物理シミュレーションシステムが、その開始時間から終了時間までのフレームをディスクに書き込みます。これらのファイルは.blendファイルと同じ場所に“blendcache”の接頭語をつけて保存されます。
 - 変更が加えられた場合キャッシュは消去されますが、すべての変更の際ではありません。よって手動で開放しなければならない場合もあります。たとえば、カ場の設定を変えたときなどです。再度キャッシュする際には、シミュレーションの開始時間よりも前の(または同じ)フレームからアニメーションさせなければならないことに注意してください。
 - ディスクに対する書き込み権限がないときはキャッシュはされません。
 - キャッシュの開放は各シミュレーションシステムごとのパネルから行えます。またCtrlBで、選択したすべてのオブジェクトのキャッシュを消去できます。
 - .blendファイルの名前がとても長くて、かつオペレーティングシステムでファイル名の長さ制限がある場合、問題が発生する可能性があります。
- バイク:
 - バイクすると、シミュレーションの結果はプロテクトされ、変更はできなくなります。
 - CtrlBで、選択したすべてのオブジェクトのバイク結果を消去できます。また現在のオブジェクトの柔体シミュレーションについてはFree Bakeで開放できます。
 - メッシュ形状を変更してもシミュレーションの再計算はされません。

レンダーファームでレンダリングするときは、すべての物理シミュレーションをバイクして、blendcacheファイルをすべてレンダーファームに送るのが良いでしょう。

リアルタイムなインタラクション

柔体シミュレーションの作業をするときには、Timelineウィンドウが便利だと気づくでしょう。目的のフレームに簡単に移動でき、つねにそのときのフレームのシミュレーションが表示されます。TimelineウィンドウのPlaybackメニューからContinue Physicsのオプションを選択すると、シミュレーション中にオブジェクトにかかわることができます。たとえばオブジェクトを動かして衝突させたり、柔体を揺らしたりできます。とても楽しいですよ！



Continue PhysicsはAltAでアニメーションしているときは働きません。

この機能はTimelineウィンドウでPlayボタンを押したときだけ使えます。

シミュレーションの途中のフレームで柔体オブジェクトを選択し、EditingコンテキストでモディファイアをApplyすると、そのときのオブジェクトの形状が定着します。

ヒント

- 柔体シミュレーションはオブジェクトの頂点が均等に分布していると特に上手く働きます。また衝突を高品質でシミュレートするには十分な数の頂点が必要です。オブジェクトのある部分に頂点を追加すると、その部分の変形のしかたが変わります（硬くなります）。*Image 1b*のアニメーションを参照。
- 衝突の計算は長い時間がかかることがあります。見えない部分は計算しないようにしましょう。
- 衝突の計算を速くするためにしばしば便利なのは、単純で透明で比較的大きな補助オブジェクトを作って、実際にはそれに衝突させることです（*Image 1a*の例を参照）。
- 柔体を使うのは、その必要があるときだけにしましょう。ぴったりした布で表面を覆った物体を単独でアニメーションさせる場合などは意味がありません。柔体の髪に自己衝突を適用することはできますが、これはややこしいことになるでしょう。それについては後の[柔体-衝突 \(eng\)](#)で扱います。
- オブジェクトそのものではなく、それに付加したLatticeやCurve Guideを柔体にすることも試してみましょう。計算が物凄く速くなるかもしれません。

リンク

- [Developer Notes](#)
- [Swinging of a chain](#)
- [Softbodies for Rigged Characters](#)

Introduction

(序文)

ライティングは、モデリング、マテリアル、テクスチャと同じく、レンダリングにおいて非常に重要な話題です。どれほど正確にモデリングされ、テクスチャを与えられたシーンでも、適切なライティング配置なしでは残念な結果となります。一方、シンプルなモデルであっても、巧みなライティングが施されれば非常に現実感のある見た目になります。悲しいことに、ライティングは未熟なアーティストにしばしば見落とされがちです。彼らは通例、現実世界のシーンの多くは単光源(ランプ、太陽など)で照らされているのだから、コンピュータグラフィックスにおいても単光源は同様の振る舞いをするのだと信じ込んでいます。これは誤りです。なぜならば、現実世界では単光源しか存在しなくても、物体からの反射光がシーン全体に広がってやわらかな影を作ります、そして影は決して真っ黒ではなく、影の中にも光が当たる部分があります。

Viewing Restrictions

(表示制限)

シーンにおいて、物体やライティングの色は様々なものから影響を受けます:

- 色の違いを識別する能力(部分色盲はありふれたものです)。
- イメージを表示するメディア(例: 液晶パネル上なのか印刷された光沢紙なのか)。
- イメージの品質(例: JPEG 画像の圧縮設定 **0.4** と **1.0** の違い)。
- イメージを見ている環境(例: CRTモニターを暗い部屋で見ているのか、日当たりの良い部屋で見ているのか)。
- 脳が、ある色と、周囲の物体や背景色との相対的な輝度値の差をどう認識するか。

だから、まったく同じ画像を、Aさんが部屋CのモニターBで見るのと、Dさんが画像のプリントアウトEを地下鉄Fで見るとでは、全然違って見えるかもしれません。

Global Influences

(大域的な作用)

Blender には、ライティングへの影響を調節できる要素があります:

- World (ワールド) の [Ambient Light](#) (周辺光) の色。
- この環境光を物体に当てるための [アンビエントオクルージョン \(eng\)](#) の使用。
- 環境光が物体の [マテリアル](#) に色を与える度合い。
- 物体の色を他のものに放射する [Indirect Lighting](#) (間接照明) の利用。
- 使用するレンダリングエンジン(Blender 内蔵エンジンと [Yafray \(eng\)](#) での違い)。
- シーン内にある [Lamp \(ランプ\)](#)。

現実世界で光が周囲にはねる物理現象のシミュレーションには、Ambient Occlusion/アンビエントオクルージョン(World の設定)、Buffer Shadows/バッファシャドウ(オブジェクトの作る影の近似)、Ray Tracing/レイトレーシング(光源からの光子の経路をたどります)が使われます。さらに、Blender では [Indirect Lighting/間接照明](#) も使えます。レイトレーシング、アンビエントオクルージョン、間接照明はコンピュータの処理負荷の高い処理です。Blender は内蔵のスキャンラインレンダラー(ほんとうに優れたものです)を使って、より速いレンダリングを行えます。この種のレンダリングエンジンは多くの単純化した仮説を前提にし、現実の光の挙動をシミュレートしようとはしないため、高速になります。

Lighting Settings

(ライティングの設定)

上記の大域的な作用を考慮した上で、シーンに光源の光を追加し始めてください。以下は制御できる、もっとも重要なものです:

- 使用する光源の種類(Sun/サン、Spot/スポット、Lamp/ポイント、Hemi/ヘミ など)
- 光源の色
- 光源の位置と向き
- Energy/エネルギーや Falloff/減衰などの光源の設定

そしてマテリアルの [シェーダー](#) がどうやって光に反応するのかというところに戻ります。

この章では、複数の光源を協力させてシーンを照らす方法を含む、上記内容を扱う予定です。Blender のさまざまな光源の種類とその挙動を分析し、長所と短所を挙げます。三点照明などの定番を含むたくさんのライティンググリグについても述べます。

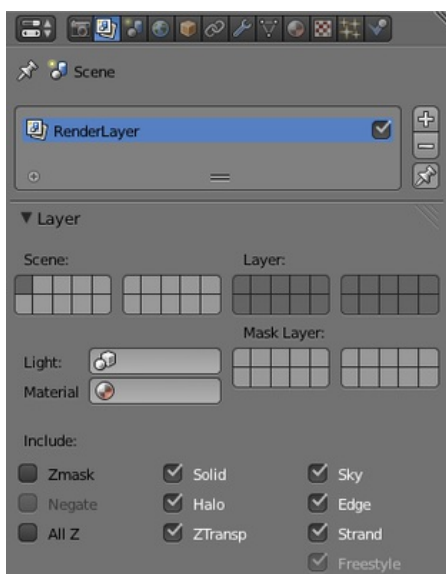
Lighting in the Workflow

(作業工程におけるライティング)

このユーザーマニュアルではライティングをマテリアルより前に置きました。メッシュにマテリアルを割り当てる前に、ライティングを設定するのがよいでしょう。マテリアルシェーダーは光に反応するため、正しいライティングなしにはマテリアルシェーダーは正しく見えなくなり、泣けるほどライティングがひどいとシェーダーと奮闘する羽目になります。この章で例として用いる画像、球や立方体、背景はすべて、マテリアルを一切使っていません。

Overriding Materials to Reset Lighting

(ライティングのリセット用のマテリアルのオーバーライド)



Render Layers パネルにある Material 項目

マテリアルの割り当てをもう始めていて、いまライティングをいじっているところなら、デフォルトの汎用の灰色マテリアルを作ることをお勧めします。Vertex Color(頂点色)、Face Texture(面テクスチャ)、Shadeless(陰影なし)のどの設定も有効でない、おなじみの無地で **(0.8, 0.8, 0.8)** の RGB 値をもつ灰色です。このマテリアルの名前を「Grey」としておきます。

次に Render Layers(レンダーレイヤー)コンテキストを開きます。Material(マテリアル)欄に、いま作った「Grey」を選んでください。このマテリアルがすでに設定済みのマテリアルより優先され、この色ですべてがレンダリングされるようになります。これでライティングの調整にとりかかれます。この欄を空白にするだけで、元のマテリアルに戻すことができます。

Lights

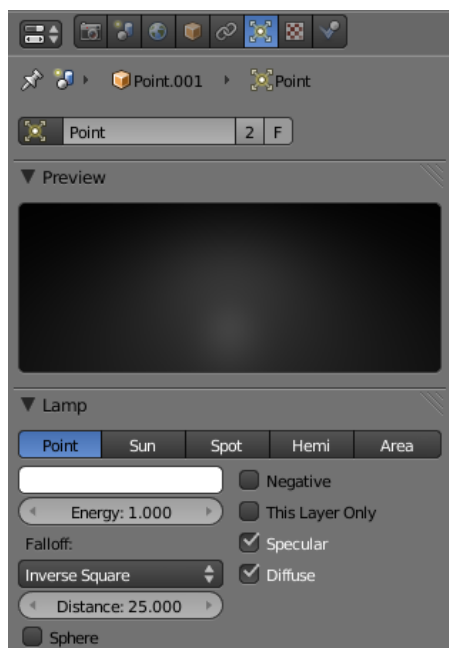
(照明)

すでに述べたように、Blenderには間接照明や環境照明といったような、幾つかのタイプの照明手法があります。しかし、もっともよく用いられるのは「lights」、つまり「lamps(ランプ)」です。この節では、これらの照明についての一般的な情報と設定について説明します(ランプの詳細については[Lamps\(ランプ\)](#)を参照してください):

- [照明のプロパティ](#) – ランプの共通設定。
- [照明の減衰](#)
- [照明テクスチャ](#) – テクスチャをランプに適用する方法
- [照明が影響するもの](#)
- [照明に関する設定](#) – 他のコンテキストにおける関連設定

Lights Common Options

(照明の共通オプション)



Lamp Properties パネル

Blender では5つのタイプの照明があります。それらは以下に示す幾つかのオプションを同様に持ちます。

Object Data

(オブジェクトデータ)

(訳注: 以下は最上部にある、ラベルのないコントロールです)

照明のオブジェクトデータを閲覧

クリックすることで現在のシーン上の全ての照明を参照できます。

名前

現在選択している照明のオブジェクトデータが持っている名前です。名前を編集することで変更できます。

ユーザー数

選択している照明のオブジェクトデータを共有する照明オブジェクトの数を示します。

F

選択している照明のオブジェクトデータを利用している箇所を仮想的に作ります。

Preview

(プレビュー)

設定した照明の簡易的なプレビューです。

Lamp

(ランプ)

Distance (距離)

Dist フィールドでは現在選択している光源の強度が半分になるブレンダーユニット (Blender Unit, 以下 BU) の値を示しています。ここに示された BU 値よりも光源との距離が小さいオブジェクトほど強く照らされ、光源との距離が大きいオブジェクトほど照らされる度合いが弱まります。ある種の設定と光の減衰タイプは Distance フィールドの値がどのように解釈されるかに影響を与えます。つまり、この設定値は常に同様に作用するわけではないのです。詳細は [光の減衰](#) を参照してください。

Energy (エネルギー)

光源の照度の強さを (0.0 から 10.0 の値で) 示します。

Color (色)

光源の色です。クリックすると色選択ダイアログが開きます。

Negative (負)

選択すると負の強さを持った光源になります。

This Layer Only (このレイヤーのみ)

選択すると照明と同じレイヤーに存在するオブジェクトだけを照らします。

Specular (スペキュラー)

反射光のハイライトが生成されます。

Diffuse (ディフューズ)

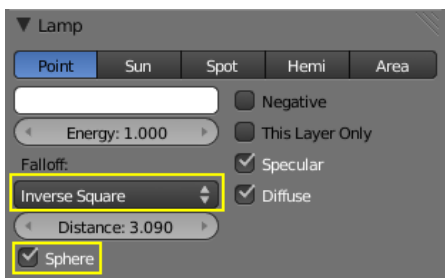
選択すると拡散光が有効になります。

Light Attenuation

(光の減衰)

Description

(解説)



Lamp パネル。減衰のオプションがハイライトされています

Point(ポイント)とSpot(スポット)光源のための、光の減衰用の主要コントロールは二つあります。

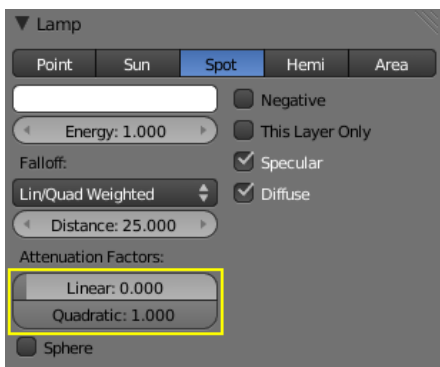
- 光源の Falloff(減衰) 種類のドロップダウンリスト
- Sphere(球) ボタン

Falloff types

(減衰の種類)

Lin/Quad Weighted

(ウェイトつき線形/二次式)



Lin/Quad Weighted Falloff(線形/二次式ウェイト) オプションがハイライトされた Lamp パネル

Lin/Quad Weighted Falloff(線形/二次式ウェイト) 設定が選ばれると表示される二つのスライダー Linear(リニア)と Quadratic(二次式)は、それぞれ減衰曲線の「線形らしさ」と「二次式らしさ」を調整します。

照明減衰のウェイトは二つを混合させることができます(linearとquadratic減衰タイプ)。

Linear

(リニア/線形)

スライダー入力欄には **0.0** から **1.0** の値が入ります。Linear(リニア)欄の値 **1.0** と Quadratic(二次式)欄の値 **0.0** の組み合わせは、この光源から届く光の減衰が完全な線形で表せることを意味します。これは、Distance(距離)欄で指定された距離(単位は Blender Units)において、この光源の強度が元の強度の半分になるということです。

Quadratic(二次式)スライダーが **0.0** に設定されると、完全な線形減衰の特定範囲の減衰計算式は次のようになります:

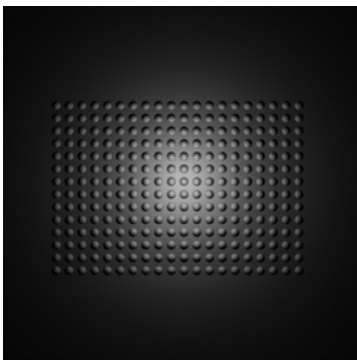
$$I = E \times (D / (D + L \times r))$$

ここで

- I は計算された光の強度(Intensity)
- E は現在の Energy(エネルギー) スライダーの設定
- D は現在の Distance(距離) 欄の設定
- L は現在の Linear(リニア) スライダーの設定
- r は光の強度を計測した場所とランプの距離

Quadratic

(二次式)



デフォルト設定の Lin/Quad Weighted 減衰を持ったランプ

このスライダー入力欄には **0.0** から **1.0** までの値が入ります。Quadratic(二次式) 欄の値 **1.0** と Linear(リニア) 欄の値 **0.0** の組み合わせは、この光源からの光の減衰が完全な二次式で表せることを意味します。

二次式の減衰タイプの照明は、(現実世界の)光の減衰のしかたをより正確に表現します。実際に Lin/Quad Weighted(線形/二次式ウェイト)用のデフォルト設定は、完全な二次式の減衰です(図「デフォルト設定の Lin/Quad Weighted 減衰を持ったランプ」をご覧ください)。

ここでも光の強度は、ランプから Distance 値だけ離れた場所で半分になります。二次式の減衰と線形の減衰を比べると、強度はセットした Distance より手前までははるかに遅く衰えますが、Distance に到達後ははるかに早く減衰します。

Linear スライダーが **0.0** にセットされると、完全な二次式の減衰の特定範囲での減衰の計算式は次のようになります:

$$I = E \times (D^2 / (D^2 + Q \times r^2))$$

ここで

- I は計算された光の強度(Intensity)
- E は現在の Energy(エネルギー) スライダーの設定
- D は現在の Distance(距離) 欄の設定
- Q は現在の Quad(二次式) スライダーの設定
- r は光の強度を計測した場所とランプの距離

Mixing “Linear” and “Quad”

(線形と二次式の混合)

Linear と Quad の両方のスライダーに **0.0** を上回る値が入ると、光の減衰の計算式は次のようになります:

$$I = E \times (D / (D + L \times r)) \times (D^2 / (D^2 + Q \times r^2))$$

ここで

- I は計算された光の強度(Intensity)
- E は現在の Energy(エネルギー) スライダーの設定
- D は現在の Distance(距離) 欄の設定
- L は現在の Linear(リニア) スライダーの設定
- Q は現在の Quad(二次式) スライダーの設定
- r は光の強度を計測した場所とランプの距離

Zeroing both “Linear” and “Quad”

(線形と二次式のゼロ化)

Linear と Quadratic の両方のスライダーに値 **0.0** が入ると、光の強度は距離によって減衰しなくなります。これは光が暗くならないことを意味しません。光は暗くなります。ただ、暗くなるのは光の持つエネルギーが広範囲にどんどん遠くまで広がった場合だけです。とは言え、広がった光のエネルギーの合計は同じままです。光の角度も、目に見える光の量に影響します。実際のところ、これは深宇宙の真空空間での光の振る舞いです。

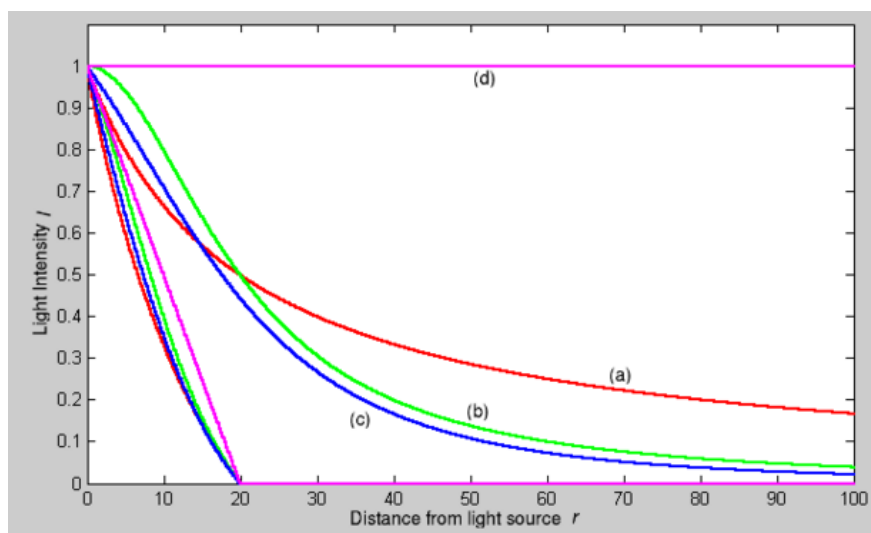
減衰せず衝突の起きたどの場所においても同量の光の強度を持つ光源を作りたいなら、Constant(一定)タイプの Falloff(減衰)のような属性を持つ照明が必要です。

また Linear と Quad のスライダーがどちらも **0.0** なら、上の式で示されるように、Distance 欄は光の減衰に一切影響しなくなります。

Graphical Summary

(グラフによる要約)

下のグラフは線形/二次式減衰の種類をまとめたもので、Sphere(球) オプション(後述)を使う場合と使わない場合の減衰を示しています。



光の減衰:

縦軸: 光の強度 I , 横軸: 光源からの距離 r

a) 線形 (Linear=1.0, Quad=0.0); b) 二次式 (Linear=0.0, Quad=1.0);

c) 線形と二次式 (Linear=Quad=0.5); d) なし (Linear=Quad=0.0).

なお「同じ」曲線の Sphere (球) ボタンをオンにしたグラフが同じ色で描きこまれています。

Custom Curve

(カスタムカーブ)

Custom Curve (カスタムカーブ) タイプの照明減衰は非常に柔軟です。

他のほとんどのランプの減衰は、光の強度を最大値で開始し(光源が一番近いとき)、光源からの距離が増すにつれ、光の強度を規定のパターンで下げることで動作します。

カスタムカーブ 照明減衰タイプを使うと新たに Falloff Curve (減衰曲線) と呼ばれるパネルが作られます。この減衰曲線を使うと光の減衰グラフに沿って、特定箇所での(光源から一定距離での)光の強度を変えることができます。



減衰曲線のグラフは二つの軸“Distance(距離)”と“Intensity(強度)”を持ちます。

距離の軸

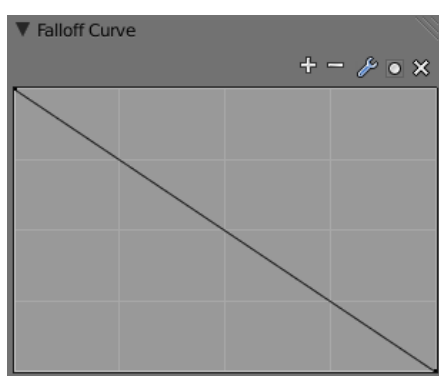
光源の減衰経路上の、特定箇所での位置を表します。左端は光源の位置、右端は光源の影響が通常は完全に弱まる位置です。「通常は」と言ったのは減衰曲線は必要なら正反対を行うように変えられるからです。

強度軸

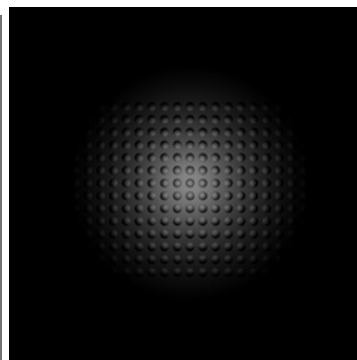
光源の減衰経路上の、特定箇所での強度を表します。高い強度は強度軸が高くなり、一方低い強度の光は強度軸上低くなることで表されます。

減衰曲線の形状グラフを変えるのは簡単です。グラフの変更したい部分を LMB  クリックして好きな場所にドラッグするだけです。小さな黒点のハンドルの上か近くをクリックすると色が白に変わり、このハンドルがいま選択されていることを表します。このハンドルをドラッグして新たな位置に動かすことができます。ハンドルの近くでないところをクリックすれば、クリックした箇所にハンドルが作られ、好きな場所にドラッグできます。また Ctrl キーを押しながら LMB  クリックすると、グラフの特定箇所にハンドルを作ることができます(付近に既存ハンドルがあっても新規作成します)。

下の例(デフォルトの減衰曲線のグラフ)では、グラフは、光の強度が最大値(光に近い側)で始まり、右に動くにつれて(光源から遠い側)線形に減衰していることを示しています。

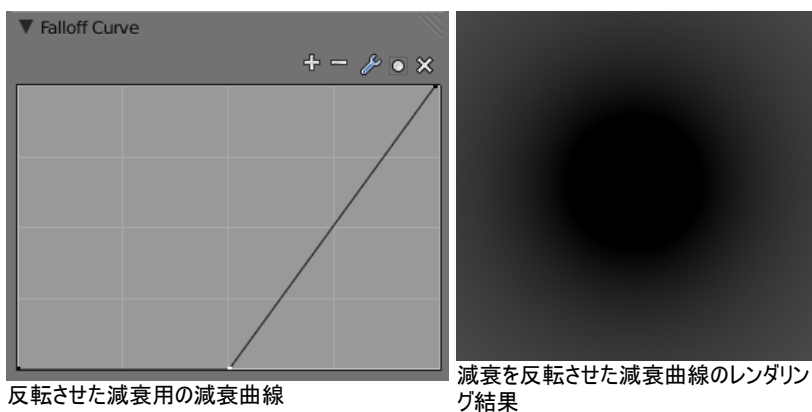


デフォルトの Falloff Curve パネルのグラフ



デフォルト設定の Custom Curve 照明減衰タイプを示すレンダリング画像

光の減衰グラフを光源から離れるほど強くしたい場合は、次のようにグラフを変えることができます:

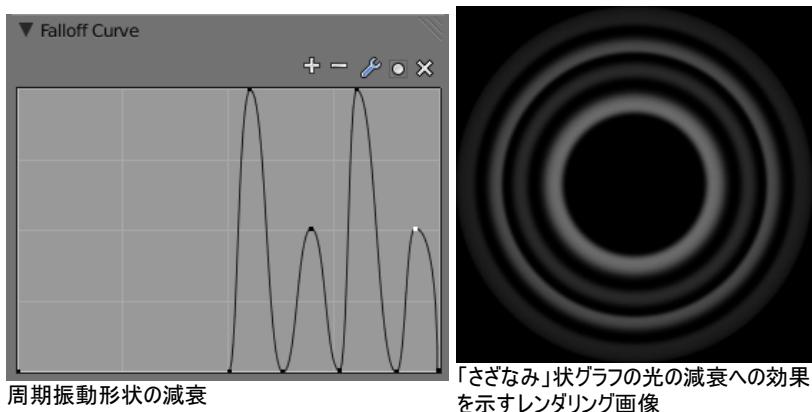


反転させた減衰用の減衰曲線

減衰を反転させた減衰曲線のレンダリング結果

もちろん減衰グラフを反転させるような簡単な変更に限られることなく、ほぼどんなグラフも作れます。

次の図は別の形をした減衰曲線のグラフと、そのレンダリング結果です:

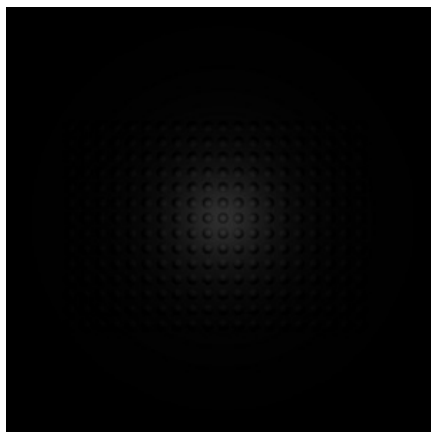


周期振動形状の減衰

「さざなみ」状グラフの光の減衰への効果を示すレンダリング画像

Inverse Square

(逆二乗式)

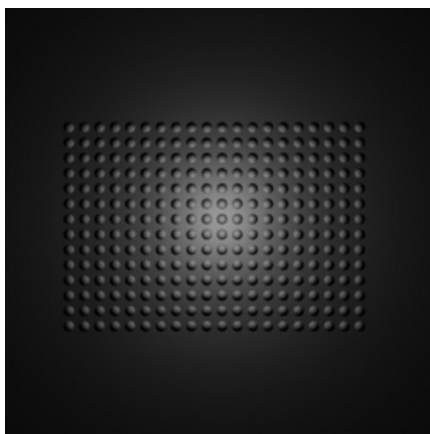


デフォルト設定の Inverse Square 照明減衰タイプを示すレンダリング画像

この照明減衰タイプは、Distance 値で拡大縮小された逆二乗式に従って強度を減衰させます。逆二乗式はより鮮明で現実味のある減衰で、卓上ランプや街灯用に役立ちます。これは昔の Quad オプションに似ていますが、少し変更があります(新しい Lin/Quad Weighted オプションの Linear が **0.0**、Quad が **1.0** になる点です)。

Inverse Linear

(逆線形)

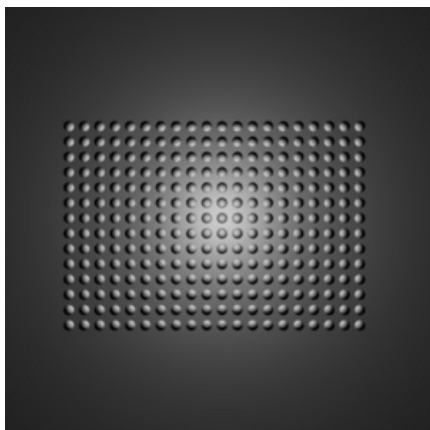


デフォルト設定の Inverse Linear 照明減衰タイプを示すレンダリング画像

この照明減衰タイプは Dist 値で拡大縮小した線形に従って強度を減衰させます。これはデフォルト設定で以前のバージョンの Blender とほぼ同じ振る舞いをします (違うのは Quad スイッチがオンで、その結果新しい Lin/Quad Weighted オプションの Linear が **1.0** に、Quad が **0.0** になることです)。物理的に正しくはありませんが、照らしやすくなります。

Constant

(一定)

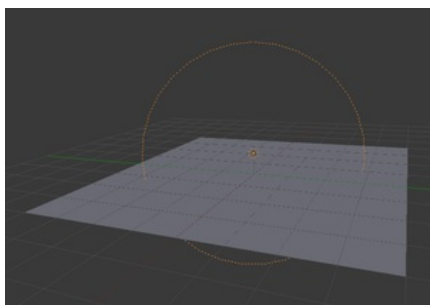


デフォルト設定の Constant 照明減衰タイプを示すレンダリング画像

この照明減衰タイプは距離によって強度を減衰させません。これは太陽や空のような、非常に距離が離れていて減衰が目立たない、遠い光源用に便利です。Sun(サン) と Hemi(ハミ) ランプは常に一定の減衰を持ちます。

Sphere

(球)



3D ビューウィンドウのスクリーンショット、Sphere(球) 形の、光のクリッピングサークルを示しています

Sphere(球) オプションは Point(ポイント) または Spot(スポット) ランプの照射範囲に制限をかけます。つまりランプから Distance 欄で指定された距離(単位は Blender Units)に一度達すると、照射が完全に止まります。

Sphere オプションを有効にすると破線の球が光源の周りに現れ、光の強度が無効になる境界点を示します。

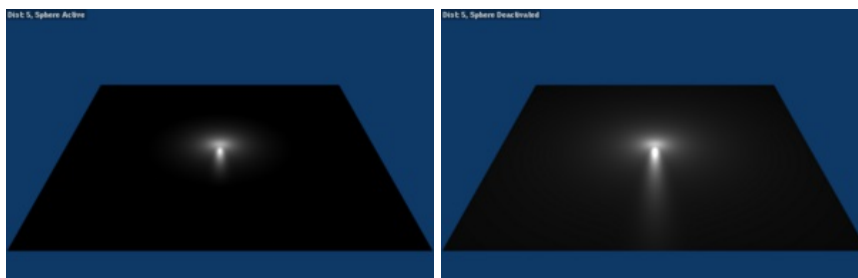
Sphere オプションは選ばれた減衰の原理(それが何であれ)に項をひとつ付け加えます:

$r < D$ ならば $I' = I \times (D - r) / D$; それ以外は 0

ここで:

- I' 必須の光の強度です (Sphere オプションが有効なとき)
- I は選ばれた減衰原理で計算された光の強度です (Sphere オプションを使わないとき)。
- D は現在の Distance 欄の設定です。
- r は光の強度を測定した場所とランプの距離です。

Lin/Quad Weighted 減衰オプションの説明の終わりにある、グラフをご覧ください。



Sphere オプションを有効にした Constant 照明減衰タイプを示すレンダリング画像

Sphere を無効にした Constant 照明減衰タイプを示すレンダリング画像

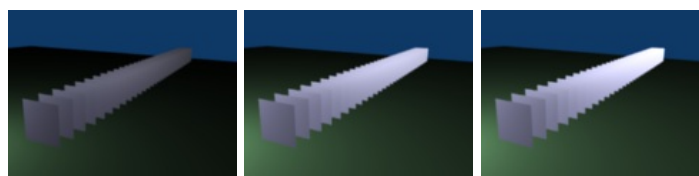
Examples

(例)

Distance

(距離)

この例では ポイントランプ は平面のグループの非常に近くに置かれています。このため前、中央、後ろの平面への光がより著しく影響されます。図「様々な Distance 設定」をご覧ください。Distance が増えるほどだんだんと多くのオブジェクトが徐々に明るくなります。



Distance: 10.

Distance: 100.

Distance: 1000.

様々な Distance 設定 (影は無効)。

Distance (距離) パラメータは光の弱まる場所を操作します。デフォルトでは線形の比率で、光の元の強さからその半分まで弱まります。この値を増減するたびに、この「半減衰」の発生する場所を変えることになります。Distance を球の表面と考えると、この表面は光の強さが全方向に、弱まって半分になった場所です。なお光の強さは Distance を越えても減衰し続けます。Distance はただ、光のエネルギーの半分が弱まる場所を指しているだけです。

図「Distance: 1000」では一番遠くにあるオブジェクトが非常に明るいことに注目してください。これは減衰がはるか遠くに延ばされたからで、最後の数オブジェクトに当たるときには光が非常に強いということです。光の強さが元の強さの半分まで弱まることは 1000 BU に遠ざかるまでありません。

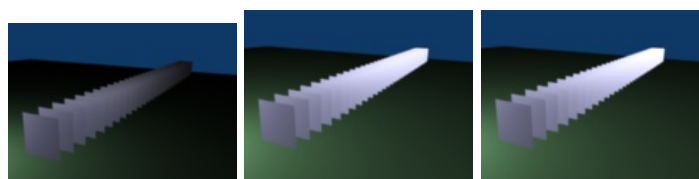
これを、減衰がすぐに起き、遠いオブジェクトがほとんど照らされていない図「Distance: 10」と比べてみてください。光の強さは10番目のオブジェクトに達する時には半分まで衰えています。

最初の数枚の平面がなぜ薄暗いのか不思議に思いませんか？ 光と、オブジェクト表面の法線が成す角が斜めに近づいているからです。これは Point (ポイント) ランプオブジェクトの特性です。光源を無限遠に動かすことで、Sun (サン) ランプの特性に近づけるかもしれません。

Inverse Square

(逆二乗式)

Inverse Square (逆二乗式) は光の強度を非線形の比率で、厳密には二次式の比率で減衰させます。Inverse Square の特性は、光の強さが非常に遅く減衰し始めますが、続いて非常に敏速に減衰し始めることです。図「Inverse Square を選択」でこの様子を見ることができます。



Inverse Square with 10.

Inverse Square with 100.

Inverse Square with 1000.

Inverse Square を選択 (Distance に指定値をセット)

大まかにいえば Inverse Square が選択されると、Distance 値は光が速く減衰し始める場所を指し示します。詳しくは [光の減衰の解説](#) をご覧ください。

図「Inverse Square with 10」では、光の強度はとても速く減衰し、最後の数オブジェクトは照らされてもいません。

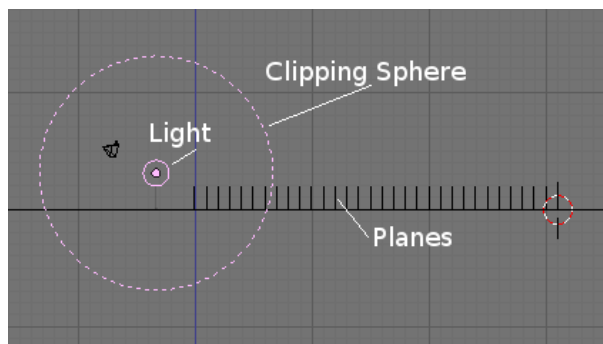
図「*Inverse Square with 100*」と「*Inverse Square with 1000*」はどちらもほとんど同じに見えますが、これは Distance が一番遠くのオブジェクトの距離である約 **40 BU** を超えているからです。このためすべてのオブジェクトがほぼ最高の強度の光を受けています。

上のように、最初の数オブジェクトはそれより奥のものより暗く見えますが、これは光源に非常に近いためです。オブジェクト表面の明るさは、オブジェクトの表面法線とランプから届く光線の成す角度にもとづいていることを忘れないでください。

つまり表面の明るさを操作するものが少なくとも二つあるということです。光の強度と、光源と表面法線の成す角度です。

Sphere

(球)



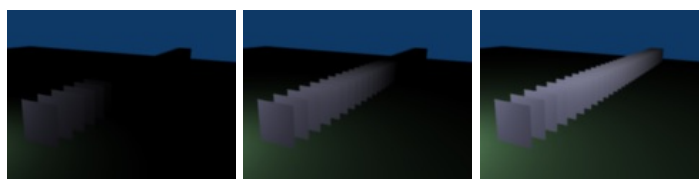
クリッピング球

Light:光源 Clipping Sphere: クリッピング球 Planes: 平面

Sphere (球) は光の強度が、選択した照明の減衰に関わらず Distance (距離) を超えると無効になることを示します。図「*クリッピング球*」は Sphere を有効にし Distance を **10** にした設定例 (の側面図) です。

球の向こう側にあるオブジェクトは、ランプから光を一切受け取りません。

Distance (距離) 欄は照明の光線が無効になる位置と強度の減衰設定の両方を表すようになります。なお球の位置で急に切り替わるわけではありません。光の減衰は進行します (詳細は上記 [Sphere オプション](#) と [光の減衰](#) をご覧ください)。



Sphere with 10.

Sphere with 20.

Sphere with 40.

指定距離で Sphere を有効化、Inverse Linear の光減衰

図「*Sphere with 10*」では、クリッピング球の半径は **10 BU** で、これは光の強度が距離 **10 BU** ごとに操作されるということです。線形の減衰を使うと、最初のオブジェクトにたどり着く前であっても光の強度は非常に低い値に減衰します。

図「*Sphere with 20*」では、クリッピング球の半径が今度は **20 BU** で、光の一部は中央のオブジェクトに達しています。

図「*Sphere with 40*」では、クリッピング球の半径は **40 BU** で、これは最後のオブジェクトを超えています。しかし光が最後の数オブジェクトに達していないのは、強度がほぼ **0** に減衰したからです。

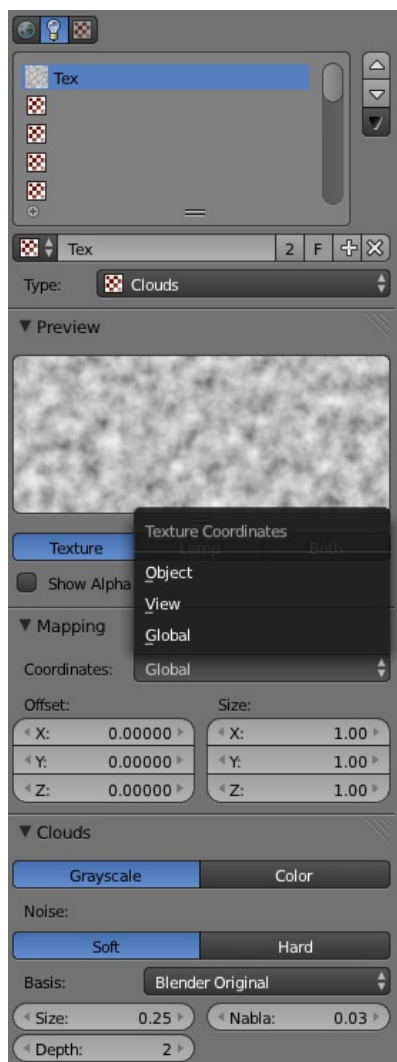
Hints

(ヒント)

ランプの光が影を作らないように設定すると、光は壁や同種のを貫通して照らします。窓の外から見える炎やろうそくで照らされた屋内装飾のような素敵な効果を作りあげるなら、Sphere (球) オプションは必須です。Distance (距離) 値を慎重に調整することで、暖かな火の光が部屋の中だけで発せられ、外では冷たい月の光が照らすようなことができます。後者は Sun (サン) か Hemi (ヘミ) 照明、またはその両方を使って作ることができます。

Lamps Textures

(ランプテクスチャ)



ランプの Texture パネル

新たに追加された光源は、均一で平坦な色の光を作ります。簡単なレンダリングではこれで十分ですが、[テクスチャ](#)を使ってもっと洗練された効果を得ることができます。繊細なテクスチャは照明に微妙な差異を持たせることができ、一方コントラストの強いテクスチャはもっとはっきりとした効果、たとえばミラーボールや木漏れ日のまだらな光、投影機のシミュレートに使えます。このテクスチャは 10あるチャンネルのうちのひとつに割り当てられ、マテリアルの表面性質ではなく光源の色と強さに影響することを除けば、マテリアルテクスチャとまったく同じように動作します。

Options

(オプション)

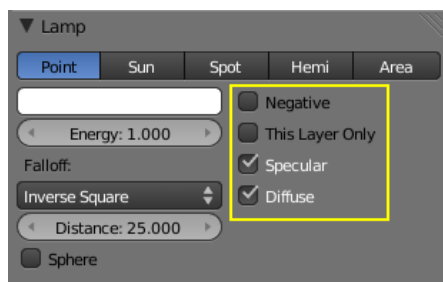
ランプテクスチャの設定は2つのパネルにグループ化されています。ここではオブジェクトのマテリアルテクスチャと異なる設定だけを説明します。標準的なオプションの詳細については [マテリアル](#)と [テクスチャ](#) の各章をご覧ください。

テクスチャの明細と Mapping (マッピング) のパネルは同じままです。ただ、Mapping オプションが減っていることに気づくでしょう。テクスチャ coordinates (座標) では Global(グローバル)、View(ビュー)、別の Object(オブジェクト) だけしか選べません(光源そのものがテクスチャ座標を持たないためです)。テクスチャは拡大縮小したりオフセットをとったりできます。

また Mapping (マッピング) パネルは、マテリアル用の同パネルの部分集合です。ランプテクスチャはおなじみの「基本 Color(色)」、「Shadow(影)の色」にしかマップできません。色にだけ作用させることができ、また光源そのものがテクスチャ座標を持たないことから、Diffuse、Specular、Shading と Geometry オプションはなくなっています。

What the Light Affects

(照明が影響するもの)



光の影響オプションがハイライトされた ランプ パネル

ランプはどれも、その光を受ける物体や、マテリアルとの相互作用の仕方を調節するスイッチを持っています。

Negative (負)

このランプが作る光は、衝突した表面の「有効な」ものから *差し引かれ*、表面を明るくする代わりに暗くします。

This Layer Only (このレイヤーのみ)

ランプが同じレイヤーにあるオブジェクトだけを照らすようにします。

Diffuse (ディフューズ)

ランプが [拡散光](#) を作らないようにします (ものを実際に照らしません)

Specular (スペキュラー)

ランプが [反射光のハイライト](#) を作らないようにします。

Lamps Related Settings

(ランプに関する設定)

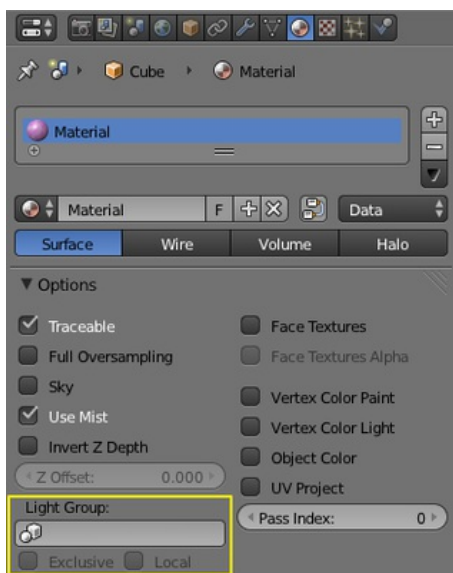
以下は、ランプの設定ではないのに、光源と密接に関わる設定です。

Lighting Groups

(照明グループ)

Materials

(マテリアル)



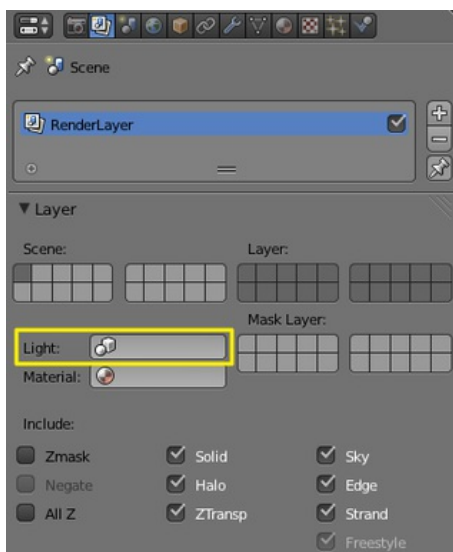
マテリアル 用の Light Group (照明グループ) オプション

デフォルトでは、マテリアルはすべての可視のレイヤーにあるすべてのランプで照らされますが、1つのマテリアル(およびこのマテリアルを使う全オブジェクト)をランプの1グループだけが照らすように制限できます。この種の制御は信じられないほど便利で、入り組んだライティングが組まれたシーンではなおさらです。有効にするには、Material(マテリアル) コンテキストの Options(オプション) パネルを開いて、ランプのグループを Light Group(照明グループ) 欄で選びます。なお [照明グループ](#) を先に作っておく必要があります。

Exclusive(排他) ボタンを有効にすると、指定グループにある照明はこのマテリアルを持ったオブジェクト *だけ* に影響します。

Render Layers

(レンダーレイヤー)



レンダーレイヤー 用の Light Group (照明グループ) オプション

似たコントロールが [Render Layers\(レンダーレイヤー\)](#) (eng) コンテキストの Layer(レイヤー) パネルにあります。照明グループ名が Light(ライト) 欄で選択されると、指定グループのランプだけでシーンが照らされます。

See Also

(参考)

- [照明](#)
- [影](#)
- [マテリアル](#)

Shadows

(影)

光はその片割れなしには存在すれません。影です。影とは、物体への光の照射が部分的または完全に遮られ、暗くなった部分です。影はシーンに、濃淡とボリュームを足します。現実世界に影のない場所はほとんどないため、現実味のある描画のためには影は必要です。Blender は次の種類の影をサポートします:

1. [レイトレースによる照明の影](#)
2. [バッファによる照明の影](#)
3. [アンビエントオクルージョン \(eng\)](#)
4. [間接照明](#)

アンビエントオクルージョンは光そのものではなく、ジオメトリを元にした影です。ただ、物体を完全に一様に照らすことのない光の効果を真似るので、ここで説明します。また Ambient を増やすほど影の効果が弱まるため、Ambient Lighting の話題も重要です。

レイトレースとバッファシャドウを組み合わせて使い、違った結果を得ることができます。レイトレースシャドウの内側においても、照明によって違った様式や強度の影が作られます。照明の並べ方によって、ある照明が他の照明の作る影を打ち消すことも、重なることもあります。

影は、結果を出すためにさまざまな場所でたくさんの準備をする必要があります:

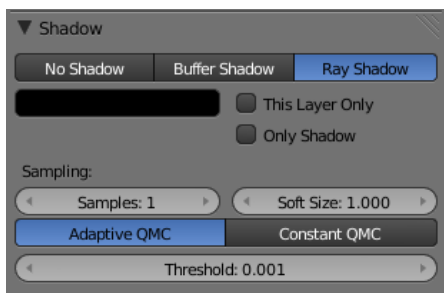
1. 照明が影を作らなければならない(能力と向き)
2. 不透明な物体は通過する光を遮らなければならない(位置とレイヤー)
3. 相手の物体の材質が影を受けなければならない(Shadow(影)と Receive Transparent(半透明影の受信) が有効であること)
4. レンダーエンジンが影を計算しなければならない(バッファシャドウ用には Shadow、レイトレースシャドウ用には Shadow と Ray)

たとえば単純な Point(ポイント)、Area(エリア) と Sun(サン) 光源はバッファシャドウではなくレイシャドウを作ることができます。Spot(スポット) 光源はどちらも作ることができ、一方 Hemi(ヘミ) 光源は一切影を作りません。Sun 光源が横向きに射すと、平面上の球から平面に影は作られません。これは光がその方向には進まないためです。影を作れるすべての光源には共通のオプションがあり、これは [こちら](#) で説明されています。

さらなる影のオプション(と混乱と問題)のご参考までに、光源と材質は、物体の diffuse/specular 特性を照らす影を作ったり受けたりすることだけができるように、個別に設定できます。またレンダーレイヤーは影のパスのオンオフ切り替えができ、出力に影情報を含むかどうかを決められます。

Lamps: Ray-traced Shadows

(光源:レイトレースによる影)



Ray Shadow enabled for a lamp

レイトレースシャドウは処理時間を費やして、非常に少ないメモリ消費で非常に正確な影を作ります。このタイプの影は Hemi を除くすべての光源で利用できます。

[バッファによる影](#)とは対照的に、レイトレースシャドウは均一光源から光線を、一様にあらゆる方向に注ぐことで得られます。レイトレイサーは次に、光線に最終画像のどのピクセルが当たるか、どのピクセルが当たらないかを記録します。当たらないピクセルは当然、影によって不明瞭になります。

光源の種類によって光線は違った方法で放たれます。たとえば Spot 光源は円錐の範囲内のあらゆる方向に均一に光線を放ちます。Sun 光源は無限遠から光線を放ち、光線はすべて Sun 光源の向きと平行になります。

シーンにレイトレーシングを有効にした光源を追加するたびに、レンダリング時間が増加します。レイトレースによる影はバッファによる影よりも多くの計算を必要としますが、非常に少ないメモリ消費で、縁の鋭い影を生みます。

レイトレースシャドウを有効にするには、3つの操作が必要です:

- Render(レンダー) コンテキストの Shading(シェーディング) パネルで Shadows(影) をグローバルに有効にする
- 同じパネルで Ray tracing(レイトレーシング) をグローバルに有効にする
- 光源の Object Data(オブジェクトデータ) コンテキストの Shadow(影) パネルにある Ray Shadow(レイシャドウ) ボタンを使って、この光源用のレイトレースによる影を有効にします。このパネルは光源のタイプによって変化します。
 - レイトレースシャドウを作れるすべての光源には共通オプションがあり、これは [レイトレースによる影のプロパティ](#) で説明されています。

レイトレースシャドウは次のタイプの光源によって作れます:

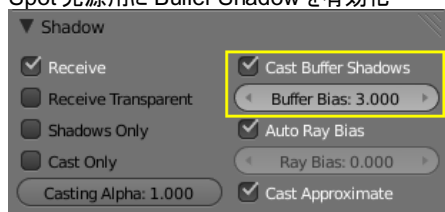
- [Point\(ポイント\)](#)
- [Spot\(スポット\)](#)
- [Area\(エリア\)](#)
- [Sun\(サン\)](#)

Lamps: Buffered Shadows

(光源: バッファによる影)



Spot 光源用に Buffer Shadow を有効化



マテリアル用に Cast Buffer Shadows を有効化

バッファによる影、バッファシャドウは精度や品質を犠牲にして、影を高速描画します。バッファシャドウはまた、メモリをレイトレースより多く必要とします。バッファシャドウを利用すべきかどうかは要件次第です。アニメーションをレンダリングする場合や、ソフトシャドウを使った複雑なシーンのレンダリングに何時間も待てない場合は、バッファシャドウは良い選択です。

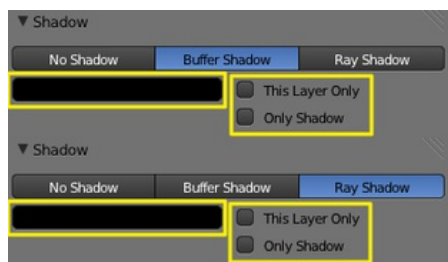
スキャンラインレンダラー用に – Blender 組み込みのエンジンはスキャンラインレンダラーです – 影は シャドウバッファを使って計算できます。スポット光源の視点で見た「画像」が「レンダリング」され、この画像内の各点とスポット光源の距離が保存されます。「レンダリング」された画像内にある、スポット光源の画像にあるどの点よりも遠い点はすべて、影になっているとみなされます。シャドウバッファはこの画像データを保管します。

バッファシャドウを有効にするには、以下の操作が必要です:

- World (ワールド) コンテキストの Gather (ギャザー) パネルで Approximate (近似) を選び、グローバルに影を有効にします
- 光源の Lamp (オブジェクトデータ) コンテキストの Shadow (影) パネルで Buffer Shadow (バッファシャドウ) ボタンを使ってこの光源に関する影を有効にします。
- 各 Material (マテリアル) コンテキストの Shadow (影) パネルで Cast Buffer Shadows (バッファシャドウを投影) オプションが有効であることを確認してください。
- [Spot lamp \(スポットランプ\)](#) はバッファシャドウを作れる唯一の光源です。

Common Shadowing Lamps Options

(ランプの影付け用共通オプション)



ランプの影付け用共通オプション

影を作るランプ ([Point\(ポイント\)](#)、[Spot\(スポット\)](#)、[Area\(エリア\)](#) および [Sun\(サン\)](#)) はすべて、以下の同じオプションを持ちます:

This Layer Only(このレイヤーのみ)

このオプションを有効にすると、光源と同じレイヤーにあるオブジェクトだけが、影を作ります。

Only Shadow(影のみ)

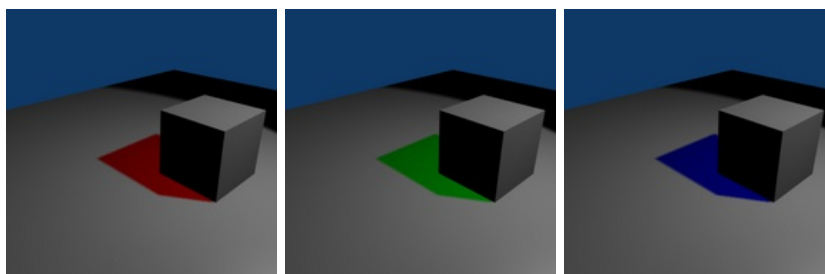
光源はオブジェクトを照らしませんが、本来現れていた影だけを生成します。

この機能は、影を落とす方法と場所の調整によく使われます。ものを照らしても影は持たない光源と、照らさないが Only Shadow(影のみ) を有効にした2つ目の光源を組み合わせ、影のみ光源をあちこちに動かして影の設置場所を操作できます。

影の色

カラーピッカーで影の色を選べます。デフォルトは黒です。

下の画像はすべて、白色の照明でレンダリングし、影の色はそれとは別に選びました。



赤色の影の例

緑色の影の例

青色の影の例

影の色は真っ白にもできますが、こうすると影が見えなくなります。

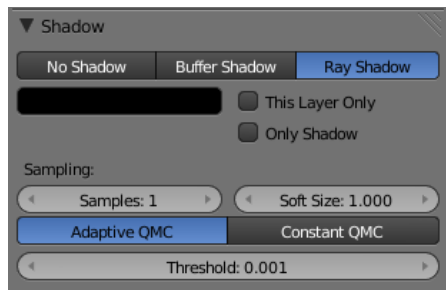
See Also

(参考)

- [影](#)
- [レイトレースによる影の共通オプション](#)
- [ポイントランプ](#)
- [スポットランプ](#)
- [エリアランプ](#)
- [サンランプ](#)
- [スポットランプのバッファによる影](#)

Lamps Raytraced Shadows

(ランプのレイトレースによる影)



ランプ用のレイシャドウのオプション

ほとんどの種類のランプ ([ポイント](#)、[スポット](#) および [サン](#)) はレイトレースによる影の生成用に同じオプションを持っており、以下に説明します。なお [エリア](#) ランプはこれらのオプションのほとんどを使いますが、[エリアランプのレイトレースによる影のページ](#) で説明されるような専用のものをいくつか持っています。

Ray Shadow (レイシャドウ)

光源がレイトレースによる影を生成できるようにします。

Ray Shadow ボタンが選ばれると、次のようなオプションが利用できるようになります：

影のサンプル生成方式

Adaptive QMC が一番高速で、Constant QMC はノイズが一番減りますが遅くなります。

レイトレースによる影の計算に貢献する、サンプル生成に使うアルゴリズムを選べます (現在のところ、主に二種類の準モンテカルロがあります。[後述](#) します)：

Constant QMC (コンスタント QMC)

非常に均一な、一様に分散する方法で影の値を計算するのに使われます。影の値を非常にうまく計算しますが、Adaptive QMC 方式ほどは速くありません。ただし Constant QMC のほうが正確です。

Adaptive QMC (アダプティブ QMC)

若干均一性の少ない、分散する方法で影の値を計算するのに使われます。影の値をうまく計算しますが、Constant QMC ほどではありません。Adaptive QMC を使う利点は一般的に高速であることですが、全体的な結果に関しては Constant QMC と比べてそれほど悪くありません。

Samples (サンプル数)

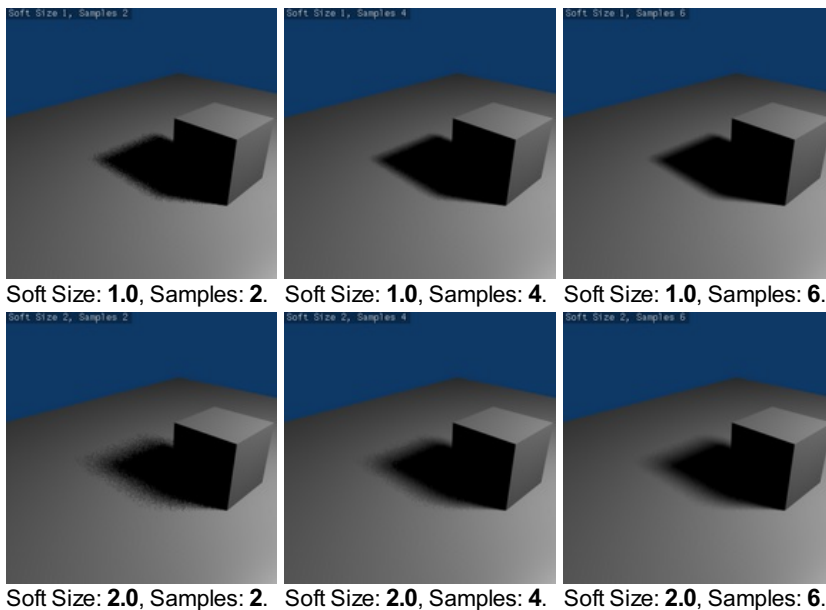
追加のサンプル数 (サンプル数 x サンプル数)。

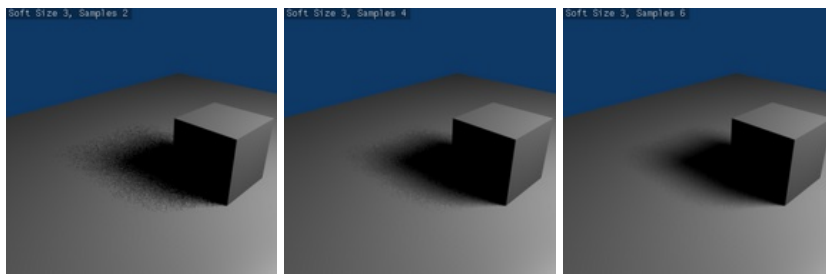
このスライダーは {Literal|Constant QMC} と Adaptive QMC の両方で影の計算に使われるサンプル数の、最大値をセットします。最大値は **16** です。サンプル数のほんとうの数は実際にはその二乗で、サンプル値が **3** なら実際には $3^2 = 9$ サンプルが取られることになります。

Soft Size (ソフトサイズ)

レイシャドウのサンプリング用の光の大きさです。このスライダーは影の縁の周囲にある、不鮮明で拡散した半影部分の大きさを決めます。Soft Size はソフトシャドウの勾配や滑らかさではなく、幅だけを決めます。やわらかくきれいな勾配の幅を持った影を作る場合は、Samples 欄にも **1** より高いサンプル数を設定する必要があります。そうしないとこの欄の効果は見られなくなり、生成された影はやわらかな縁を持たなくなります。Soft Size の最大値は **100.0** です。

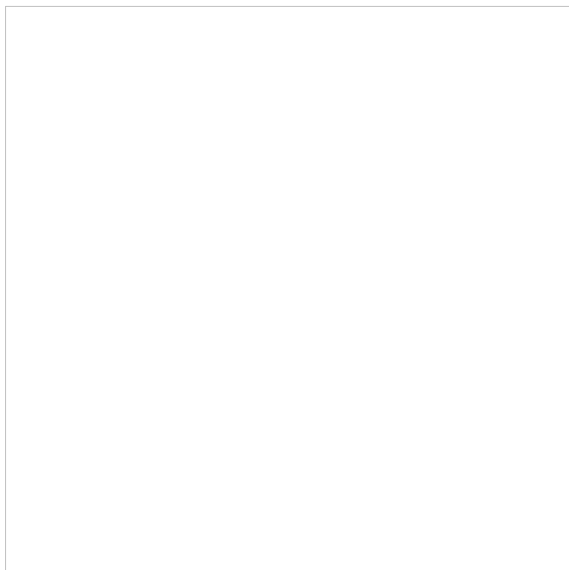
以下は さまざまな Soft Size と Samples 設定によるレンダリング画像の表で、さまざまな値の影の縁のやわらかさへの影響を示しています：





Soft Size: **3.0**, Samples: **2**. Soft Size: **3.0**, Samples: **4**. Soft Size: **3.0**, Samples: **6**.

以下は上の表の画像のアニメーション版です:



さまざまな Soft Size と Samples 設定によるレンダリング画像のアニメーション版で、さまざまな値の影の縁のやわらかさへの影響を示しています

Threshold (しきい値)

Adaptive QMC 方式で使われます。ある領域がすでに影付けされていると判断するしきい値で、Adaptive QMC 影のサンプル計算が省略できるかどうかを決めるのに使われます。Threshold の最大値は **1.0** です。

What is Quasi-Monte Carlo?

(準モンテカルロ法とは)

モンテカルロ法とは、計算全体の正確性の測定に通常複数の標本値が必要な計算において、行うべき正しい操作を決めるため、ランダムな領域またはその周辺から一連の値(光の値、色の値、反射状態のようなあらゆる種類の値)の標本/読み取り値を取り出す方式です。モンテカルロ法は可能な限りランダムであろうとします。この方式はよく、抽出領域に大きな規格外の空白(標本抽出/読み取りがされていない場所)を発生させます。このことが回りまわってとある計算(影の計算など)で問題を起しかねません。

この解決策が準モンテカルロ法です。

準モンテカルロ法もランダムですが、標本/読み取りが分散し(抽出領域内に残される規格外の空白を少なくします)、領域に渡ってもっと一様に広がるよう努めます。この方法には時折、標本/読み取り値にもとづいたもっと正確な計算に至るという利点があります。

Volumetric Lighting

Wikipedia の [volumetric lighting\(英文\)](#) によれば、

« ...は 3D コンピューター・グラフィックスで使われる技術で、レンダリングされたシーンに照明効果を付加します。環境に差し込む光の筋が目に見えるようになります。窓から差し込む日差しは volumetric lighting の一例で、ゴッドレイとしても知られています。この表現は映画技術より導入されたようで、現在は 3D モデリングやレンダリングで広く使われ、特に 3D ゲームの分野で使われます。volumetric lighting では、光源から放射される光の円錐が透明な物体としてモデリングされ「体積 (volume)」のある入れ物とみなされます。この結果光は現実世界と同じように、その体積内にある本当に 3 次元的な媒体 (霧、ほこり、煙、蒸気など) を、通過するエフェクトを生むことができます。»

古典的な例として、サーチライトで辺りを見回したとき、ライトから放たれる目に見える光輪や光芒があります。

デフォルトでは Blender はこの性質の光を作りません。たとえば Blender が スポットランプ で何かに光をあてると、その物体と床は照らされて見えますが、対象に向けて進み途中で拡散するはずのスポットライト由来の光芒/光輪は見えません。

現実世界では光が空気中の粒子によって散乱し、一部がそれて人の目に入り、光輪/光芒として知覚されます。光源からの光の散乱は様々なオプションを使って Blender でシミュレートできますが、デフォルトでは有効化されていません。

volumetric 効果を作れる唯一の光源が [Spot ランプ](#) です (Sun ランプの "[Sky & Atmosphere\(空と大気\)](#)" 効果の一部も volumetric とみなせるかもしれませんが)。

Example

(例)

[スポットライトアニメーションの Blend ファイル](#)
[\[video link\]](#)

See also

(参考)

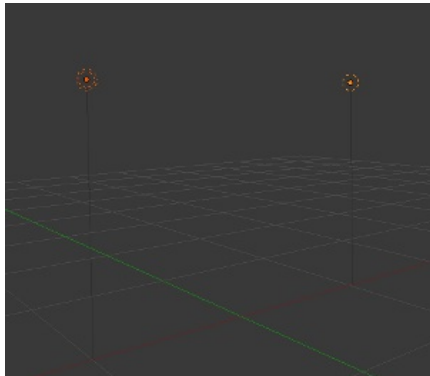
- [Mist\(霧\)](#)
- [Smoke\(煙\)](#)
- [Volume Rendering\(ボリウムレンダリング\)](#)

Lamps

(ランプ)

Blender は5種類の光源を備えており、それぞれに独自の長所と短所があります。以下は利用できる光源です:

- [Point\(ポイント\)](#) は全方向性の点光源で、白熱電球に似ています(訳注:2.4 系ではこのランプは単に「Lamp」という名前でした。このため2.4系の記事を流用している記事では Lamp がこのランプを指していることがあります)
- [Spot\(スポット\)](#) は指向性の点光源で、スポットライトに...似ています。
- [Area\(エリア\)](#) は窓やネオン、テレビ画面のような、光を生む領域をシミュレートする光源です。
- [Hemi\(ハミ\)](#) は非常に広く、遠い場所にある空のような光源をシミュレートします。
- [Sun\(サン\)](#) はかなたにある、太陽に似た点光源です。



ふたつの点光源の鉛直高さや影のマーカ。左の光源はレイシャドウが有効にされています

シーンへの新しい光源の追加は、ヘッダーの Add(追加)メニューか、⇧ ShiftA » Add(追加) » Lamp(ランプ) を使って行えます。

追加すると3Dビューに、光源の位置が丸囲みの点で示されます。ほとんどのタイプは原点と属性を知るのに役立つ破線のワイヤーフレームを持っています。タイプ別に違う表現がされますが、すべてに共通する表示があります:

Shadows(影)

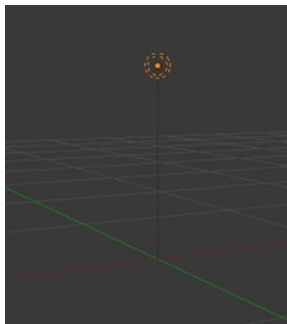
影が有効なら、塗りつぶされた円の周りに追加の破線の円が描かれます。光源が影を有効にしているかどうかをわかりやすくします。

Vertical Height Marker(鉛直高さマーカ)

暗い灰色の線で、グローバルな X-Y 平面からの光源の相対位置を決めるのに役立ちます。

Point Lamp

(ポイントランプ/点光源)



Point ランプ

ポイントランプは全方向性点光源で、つまりすべての向きに同じ量の光を放射する点です。ポイントランプは簡単な丸囲みの点で表示されます。点光源であるために、オブジェクトの表面に衝突している光の向きは、ランプとオブジェクト表面の点そのものを結ぶ直線で決まります。

光の強さ(intensity)とエネルギー(energy)は(他の変数も影響しますが、中でも特に)ポイントランプから物体への距離によって衰えます。言い換えれば、遠い面ほど暗くレンダリングされます。

Lamp Options

(ランプのオプション)

Distance(距離)、Energy(エネルギー)、Color(色)

この設定はほとんどのタイプのランプで共通で、[照明のプロパティ](#) で説明されています。

Negative(負)、This Layer Only(このレイヤーのみ)、Specular(スペキュラー)、Diffuse(ディフューズ)

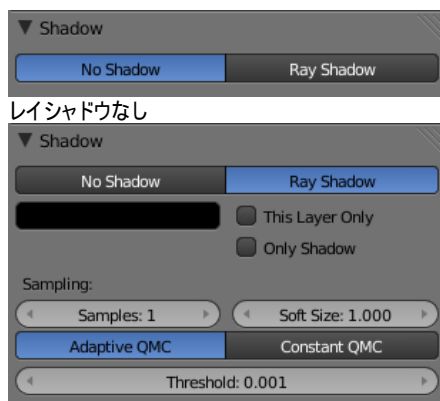
この設定はランプが影響するものを調整します。[照明の影響するもの](#) で説明されています。

Falloff(減衰)と Sphere(球)

この設定は ランプ の光の、距離による衰え方を調整します。詳細は [照明の減衰](#) をご覧ください。

Shadows

(影)



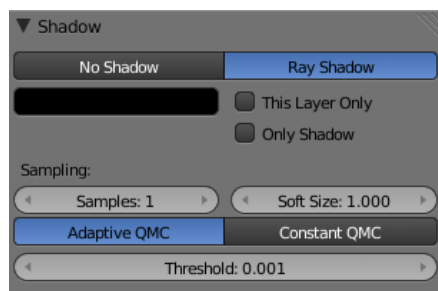
レイシャドウと Adaptive QMC サンプル生成を有効にした ポイントランプ

ポイント光源はレイトレースによる影だけを作ります。影付けのオプションは他のタイプのランプと同じものを使います。[影のプロパティ](#) で説明されています。

レイトレースされた影の設定は他のタイプと共通で、[レイトレースによる影のプロパティ](#) で説明されています。

Raytraced Shadows

(レイトレースによる影)



Shadow パネル

Point (ポイント) 光源はレイトレースによる影だけを作ります。影付けの共通オプションは、[影のプロパティ](#) で説明されているように、他のタイプのランプと同じものを使います。

レイトレースによる影の設定は他のタイプと共通で、[レイトレースによる影のプロパティ](#) で説明されています。

Spot Lamp

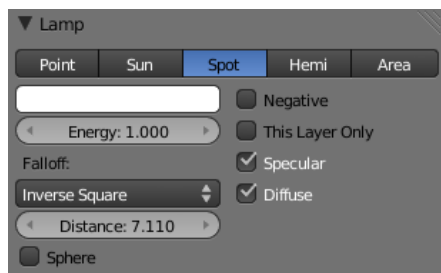
(スポットランプ/点光源)

Spot(スポット) ランプは円錐状の光の束を、円錐の先端から指定された向きに放射します。

スポットランプは照明オブジェクトの中で一番複雑で、実際長い間影を作れる唯一の光源であったためもっともよく使われていました。現在は、レイトレイサーが Blender の内蔵レンダーエンジンに組み込まれており、すべてのランプが影を落とすことができます (Hemi(ヘミ)を除く)。とはいえ スポットランプのシャドウバッファはレイトレースによる影よりもレンダリングが速く、プレーヤソフトがかけられたときは特にそうです。スポットランプは「ボリューメトリック」な halo のような、他の機能も備えています。

Lamp options

(ランプのオプション)



スポットの、ランプ共通のオプション

Distance(距離)、Energy(エネルギー)、Color(色)

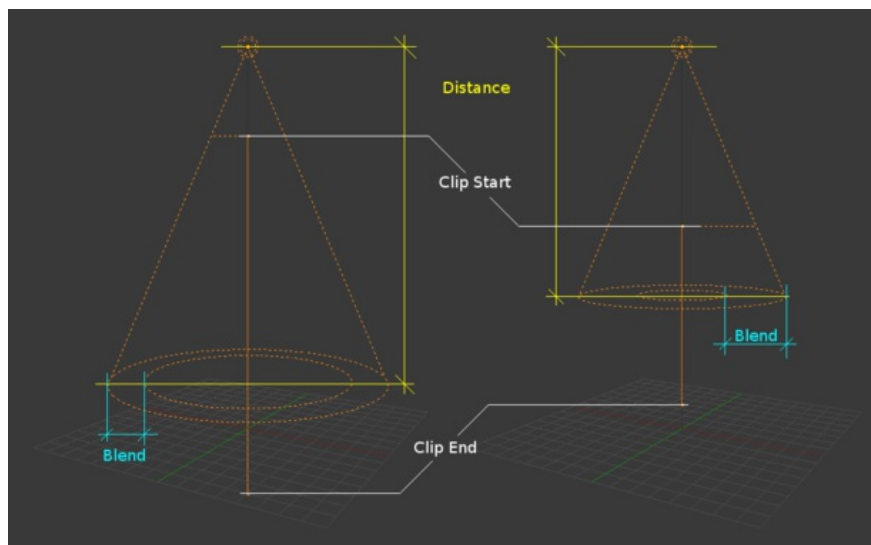
この設定はほとんどのタイプのランプに共通で、[照明のプロパティ](#) で説明されています。

This Layer Only(このレイヤーのみ)、Negative(負)、Diffuse(ディフューズ)、Specular(スペキュラー)

この設定は照明が影響するものを調整します。[照明が影響するもの](#) で説明されています。

Light Falloff(光の減衰)と Sphere(球)

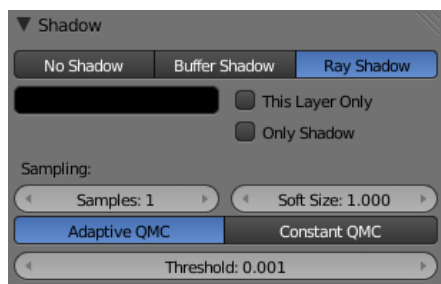
この設定は スポット の光が距離とともにどのように減衰するかを調整します。詳細は [照明の減衰 \(eng\)](#) をご覧ください。



スポットのオプション設定の変更により 3D ビューに表示されているスポットライトの見た目も変化します

Shadows

(影)



Ray Shadow にセットされた Shadow パネル

スポットライトはレイトレースによる影とバッファによる影のどちらも使えます。この二つはどちらもさまざまな追加のオプションを備えています。レイトレースさ

れた影のほうが通常は正確で、透明な影などの追加機能を持ちますが、レンダリングは非常に遅くなります。

No Shadow(影なし)

このスポットランプからの影を切る場合に選んでください。シーンに控えめな指向性の光をいっくら足すのに便利です。

Buffer Shadow(バッファシャドウ)

バッファシャドウは深度マップシャドウとしても知られています。影は光からシーンオブジェクトまでの距離の差を計算することで作られます。この機能を使うための詳細は [バッファシャドウ](#) をご覧ください。

バッファシャドウは準備が複雑になり、含まれる嘘は増えますが、レンダリングの速さは間違いなく長所です。他のタイプのランプと同じ [影の共通オプション](#) を持ちます。

Ray Shadow(レイシャドウ)

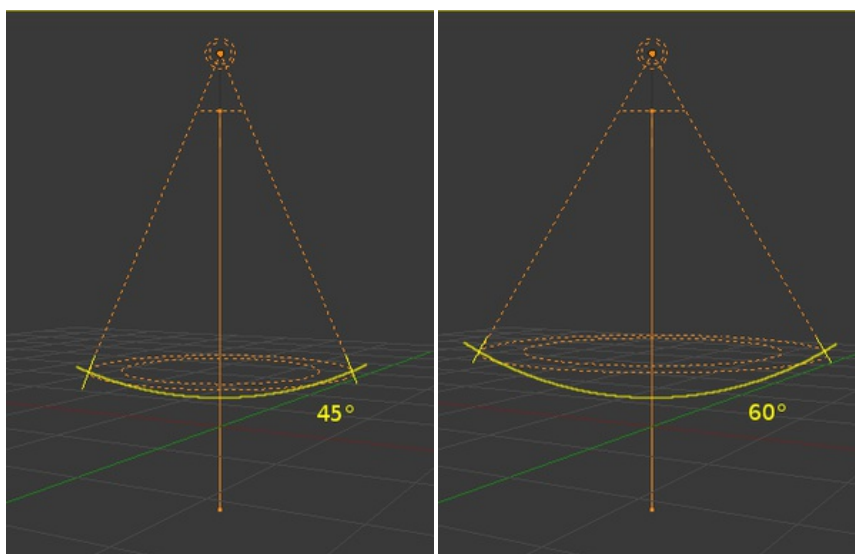
レイトレースによる影の設定は他のタイプのランプと同じもので、[レイトレースによる影のプロパティ](#) で説明されています。

Spot Shape

(スポット形状)

Size(サイズ)

スポットの外側の円錐の大きさを、主にスポットライトが覆う丸い領域を調整します。このスライダーは実際のところ、照明円錐の頂点の角度を調整し、**1.0°** から **180.0°** までの値にできます。



スポットランプの Size オプションの変更

Blend(ブレンド)

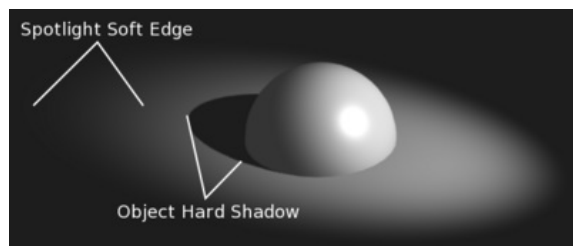
スポットの内側の円錐を調整します。Blend 値は **0.0** から **1.0** までの値にできます。この値は外側の円錐 (Size(サイズ)) 内に内側の円錐が占める割合と大きさを表しています。

内側の円錐の境界線は、スポットからの光がぼやけてやわらかくなり始める点を示しています。この点より前では光はほぼ最高の強度です。値を増やすほどスポットライトの縁はぼやけてやわらかくなり、小さくするほど内側の円錐の円形領域が小さくなります (これに伴い、早くぼやけてやわらかくなります)。

スポットに鋭い減衰率を持たせて縁のぼかし/やわらかさを抑えるには、Blend (ブレンド) 値を下げます。Blend を **0.0** にすると光と影間の遷移区間がまったくなくなり、スポットライトの縁が非常に鋭くなります。

スポットランプの光の減衰率は Blend と Size の比率です。二者間の環状の隙間が大きくなるほど、Blend と Size 間で光はゆるやかに弱まります。

Blend と Size は スポット ライトの円錐の開口部とやわらかさ(「放射状の」減衰)だけを調整します。上述のように影のやわらかさは調整しません。



レンダリング画像は、縁のやわらかなスポットライトが当てられた領域とオブジェクトの鋭く硬い影を示しています

上の画像ではオブジェクトの影はレイトレースの結果のように鋭く、一方スポットライトの縁はやわらかであることに注目してください。スポットの領域内で他のものがやわらかな影を作りたい場合は、影の他の設定を変える必要があるでしょう。

Square(正方形)

スポットライトがデフォルトの丸い領域ではなく、正方形の光の束を放射するようにします。

Show Cone (コーンを表示)

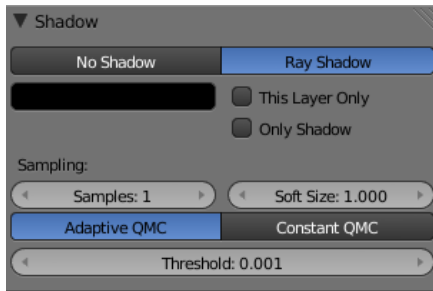
3D ビューに透明な円錐を描き、内部に含まれるオブジェクトをわかりやすくします。

Halo (ハロー)

ポリューメトリック効果をスポットランプに付加します。[スポットハロー](#)をご覧ください。

Raytraced Shadows

(レイトレースによる影/レイシャドウ)



Shadow パネル

Spot(スポット) 光源はレイトレースによる影だけを作ります。影付けのオプションは他のタイプのランプと同じものを使います。[影のプロパティ](#) で説明されています。

レイトレースされた影の設定は他のタイプと共通で、[レイトレースによる影のプロパティ](#) で説明されています。

Spot Buffered Shadows

(スポットランプのバッファシャドウ)



スポット ランプ用に有効にされた バッファシャドウ

スポットライトは [レイトレースによる影](#) とバッファによる影のどちらも使えます。この二つはどちらもさまざまな追加のオプションを備えています。

レイトレースされた影は通常より正確で、透明な影などの追加機能を持ちますが、レンダリングは非常に遅くなります。

バッファによる影は、準備が複雑になり、含まれる嘘は増えますが、レンダリングの速さは間違いなく長所です。他のタイプのランプと同じ [影の共通オプション](#) を持ちます。

Shadow Buffer Types

(シャドウバッファの種類)

Buffer Shadow (バッファシャドウ) ボタンを有効にすると、現在選択している スポット 照明がレイトレースではなく「シャドウバッファ」を使って影を生成します。Shadow (影) パネルにはさまざまな追加のオプションやボタンが現れます。

Buffer Type (バッファの種類)

バッファシャドウを生成する方法は一つではありません。このコントロールは使用するジェネレータを選びます。

影の生成方式には次の4つがあります:

- Classical (クラシカル)
- Classic-Halfway (クラシカルハーフウェイ)
- Irregular (イレギュラー)
- Deep (ディープ)

影の生成方式の詳細は次のリンクをご覧ください:

- [Development Release Logs 2.43: Irregular Shadow Buffer \(英文\)](#)
- [Blender Nation: Blender Gets Irregular Shadow Buffers \(英文\)](#)
- [Development Release Logs 2.43: Shadow Buffer Halfway Average \(英文\)](#)

“Classical” and “Classic-Halfway”

(「クラシカル」と「クラシカルハーフウェイ」)



Classic-Halfway にセットされた バッファシャドウ

Classical (クラシカル)

かつて Blender のデフォルトだった、バッファシャドウ生成の独自方式です。バッファシャドウ生成の古い方法を使っていましたが、生成された影の正確性に問題が起きることがあり、シャドウバッファの解像度 (Shadow Buffer→Size)、Bias 値の変化、それにこの方法がもたらすすべてのセルフシャドウの問題に非常に過敏です。

影生成の Classical 方式はもう使われることはなく、Blender の過去のバージョンとの後方互換性のためだけに存在します。他のほとんどの場合、Classic-Halfway を代わりに使うほうがよいでしょう。

Classic-Halfway (クラシカルハーフウェイ)

このタイプのシャドウマップは改良されたシャドウマップ方式で、Blender のデフォルトオプションです。最も近いZ深度値と二番目のZ深度値の平均値をとることで動作し、Bias 値を下げられるようにしながらもセルフシャドウの問題で苦しむことがそれほどありません。

Bias 値が大きいと、小さな面が影を失う可能性があると言えるので、Bias 値を上げる必要がないことは、大きな値で影が大きすぎることを避けると同時に、影の正確性に貢献します。

Classic-Halfway は面が重なっているとあまりうまく動かず、バイアスの問題が起きることがあります。

以下はこれらの生成方式特有のオプションです:

Size (サイズ)

Size 数値項目には **512** から **10240** までの値が入ります。Size はシャドウマップを作るのに使う解像度を表します。シャドウマップはシーン内のどこに影を置か決めるのに使われます。

例として Size 欄に **1024** を入力していれば、影のデータは選択したスポットライトからのサンプルごとに **1024×1024** ピクセルの正方形の解像度のマップに書き込まれます。

Size 以外の光源やシーンのプロパティがすべて同じままなら、Size 値を高くするほど解像度やできあがる影の正確性は高まりますが、メモリ使用量や処理時間は増えます。逆もまた真です。Size 値が下げられるとできあがる影の質は低くなりますが、計算に必要なメモリや処理時間は減ります。

Size 値が生成された影の質に影響すると同様に、Spot ランプのマップシャドウの質に影響するもう一つのプロパティが、スポットライトの照らす領域の角度です (Spot Shape (スポット形状) パネルの Size (サイズ) 欄で指定します)。

スポット形状のサイズ値が増えると、作られる影の質は下がります。これは (スポット形状のサイズ値が増え) スポットの照らす範囲が広くなると、シャドウマップ領域を引き延ばし、新たな照射範囲の大きさに合わせて拡大しなければならないからです。

Size の解像度がスポットライトのサイズ変更を補うために変更されることはなく、したがって影の質は落ちます。生成された影を同じ質に保ちたい場合は、スポット形状の Size 値を上げるのに合わせて、マップの Size 値も上げる必要があります。

要するに

大きなスポットライトには、影の質を保つため大きなマップ Size が必要になります。逆もまた真です。生成された影の質はふつう、スポット ランプの覆う領域が小さいほど向上 (上限を上回らない範囲で) します。

Filter Type (フィルターの種類)

Box (ボックス)、Tent (テント)、Gauss (ガウス) の各フィルターはマップシャドウのアンチエイリアスを行うのに、どのフィルタリングアルゴリズムを使うのか調整します。

Samples 数値欄と密接な関係があり、この設定が **1** にされるとシャドウフィルタリングは無効になり、これらのボタンは何の効果も持たなくなりません。

Box (ボックス)

マップシャドウは「ボックス」フィルタリング方式を使ってアンチエイリアス処理されます。

これは Blender で使われる独自のフィルターです。比較的低品質で、非常に鋭いアンチエイリアスを作るため、低解像度のレンダリングに使われます。このフィルターを使うと、単一ピクセルに属するオーバーサンプリングデータだけを考慮し、周囲のピクセルサンプルは考慮しません。鋭角要素や水平/垂直な線を含む画像に役立つことがよくあります。

Tent (テント)

マップシャドウは「テント」フィルタリング方式を使ってアンチエイリアス処理されます。

鋭い結果を生む簡単なフィルターで、優秀な汎用的フィルタリング方式です。このフィルターは最終的なフィルタリング値の計算時、隣接ピクセルのサンプル値を考慮に入れます。

Gauss (ガウス)

マップシャドウは「ガウス」フィルタリング方式を使ってアンチエイリアス処理されます。

非常にやわらかでぼやけたアンチエイリアスを作ります。この結果、高解像度のレンダリングとよく合います。

レンダリングの章の [アンチエイリアスのページ](#) には、さまざまなフィルタリングや配置方式とその用途について、さらなる情報があります。

Samples (サンプル)

Samples 数値欄には **1** から **16** までの値が入ります。シャドウマップの計算時に、ピクセルごとに取得するサンプル数を調整します。

値が高いほどフィルターされ、ランプの作る影は滑らかでアンチエイリアスのかかったものになりますが、計算時間は長くなりメモリ使用は増えます。使われるアンチエイリアス方式は、Box、Tent、Gauss のどれが有効であるかによって決まります (上記参照)。

Samples 値を **1** にすることはマップシャドウ用のアンチエイリアスを切るのと同じことです。

Soft (ソフト)

Soft 数値欄には **1.0** から **100.0** までの値が入ります。マップシャドウにアンチエイリアス処理を行うとき、抽出される範囲の広さを示しています。生成された影の縁のアンチエイリアス/ソフトがかけられた範囲は、Soft 値が大きいくほど滑らかで、やわらかくなります。

Sample Buffers (サンプルマップ)

Sample Buffers 設定には値 **1**、**4**、**9** のいずれかを選び、マップシャドウのアンチエイリアス処理時に使われる、シャドウマップの数を表します。

このオプションは特別な場合、たとえば (ストランドのような) 非常に小さなオブジェクトが動いていて、ほんとうに小さな影を生成する必要のあるときに使われます。通常はピクセル幅の影には正しくアンチエイリアスがかからず、Buffer Size を増やしてもそれほど効果がないように見えます。

つまりこのオプションは上記の標準の抽出 (Box/Tent/Gauss、Samples および Soft 設定で調整されるもの)を行った上で、一種の追加の抽出パスをもたらします。

デフォルトの値 **1** はこのオプションを無効にします。

値を高くするほど滑らかなアンチエイリアスを作ります。しかし注意してください。Sample Buffers を **4** にすると、Blender が4倍のサンプルバッファを計算する必要があるため、4倍のメモリや処理時間が必要になります。

“Irregular”

(「イレギュラー」)



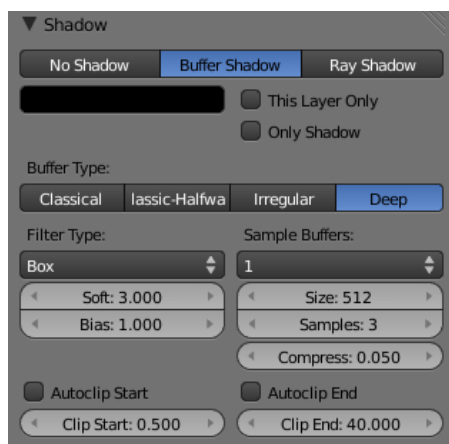
Irregular にセットされた バッファシャドウ

Irregular シャドウ方式は鋭く硬い影を生成するのに使われ、この影はレイトレースによるものと同じくらい正確に設置されます。この方式はマルチスレッド処理が行えるため処理性能は非常によくなります。

この方式は透明な影をサポートします。このためにはまず、透明な影を受けるオブジェクトに対して、影の設定を行う必要があります (Material (マテリアル) → Shadow (影) → Cast Buffer Shadows (バッファシャドウを投影) および Buffer Bias (バッファバイアス))。

Deep generation method

(「ディーブ」)



Deep にセットされた バッファシャドウ

Deep シャドウバッファはメモリ使用や処理時間の増加と引き換えに、透明度とさらに優れたフィルタリングをサポートします。

Compress (圧縮): ディープシャドウマップの圧縮のしきい値

Common options

(共通のオプション)

以下はすべてのバッファシャドウ生成方式に共通する設定です。

Bias (バイアス)

Bias 数値欄には **0.001** から **5.0** までの値が入ります。Bias はオブジェクトとそれが投じる影の間にわずかにオフセット距離を足すのに使われます。この措置はオブジェクトの一部が影の中にあるかどうかを調べる計算の不正確さのために、必要になることがあります。

Bias (バイアス) 値を小さくするほどオブジェクトと影間の距離は狭くなります。Bias 値が小さすぎると、オブジェクト上に線や邪魔な模様のような画像の乱れ (artifacts) が現れることがあります。この問題は通常「セルフシャドウイング」と呼ばれ、Bias 値を増やすことで修正できます (この項目はこのために存在しています！)。

セルフシャドウイングの修正方法としては他に、Shadow Buffer Size を大きくするか Classic-Halfway や Irregular のような別の影計算方式を使います。

セルフシャドウイング干渉は平らな表面よりも曲がった表面に作用する傾向があります。シーンに曲がった表面がたくさんあれば Bias 値か Shadow Buffer Size 値を上げる必要があるということです。

必要以上に大きな Bias 値は、影を作るオブジェクトから遠く離れて影が設置されるだけでなく、非常に小さなオブジェクトが影をまったく作らせないようにします。そこで問題修正のために、特に Bias、Shadow Buffer Size、Spot Size 値を変更する必要があります。

より細かなバイアス調整

Bias スライダーを使って、Bias 値を [マテリアル](#) ごとに調整可能です(マテリアル コンテキストの Shadow(影) パネル)。この値はランプの光がこのマテリアルを使ったオブジェクトに衝突するたびに、各 スポット ランプのバツファシャドウの Bias 値に掛け合わされる係数です。値 **0.0** と **1.0** は等値です。どちらもランプの元の Bias 値を変化させません。

Clip Start(クリップ始端) & Clip End(クリップ終端)

バツファシャドウを持つ スポット 光源がシーンに追加されると、3D ビューの スポット の表示に、追加の線が現れます。

この線の始点は Clip Start 値を表し、終点は Clip End 値を表します。Clip Start には **0.1** から **1000.0** までの値を、Clip End には **1.0** から **5000.0** までの値を入力できます。どちらも単位は Blender Units です。

Clip Start は、Spot の照射域内でバツファシャドウが存在できる最初の位置を示します。この位置より前に存在し得る影はどれも無視され、影は生成されません。

Clip End は、Spot の照射域内でバツファシャドウが存在できる最後の位置を示します。この位置より後ろに存在し得る影はどれも無視され、影は生成されません。

Clip Start と Clip End 間の範囲は、生成されるバツファシャドウを持つことができます。

Clip Start と Clip End 値を変えて、影が生成できる場所を制御できます。Clip Start と Clip End 間の範囲の調整は、レンダリングの高速化やメモリの節約、できあがる影の正確性の向上に役立ちます。

バツファシャドウをともなう スポット ランプを使うとき、生成される影の管理や質の向上のために、Clip Start と Clip End を調整して影を生成したい領域付近まで近づけるとよいでしょう。Clip Start と Clip End の間の領域を最小化すると影の計算される領域が最小化され、領域が狭いほど影の質が上がるためです。

Autoclip Start(オートクリップ始端) & Autoclip End(オートクリップ終端)

バツファシャドウの開始および終了位置を決めるのに Clip Start と Clip End 欄を手動で設定する以外に、Blender に Clip Start と Clip End 欄の最適値をそれぞれ選ばせることができます。

Blender は Spot ランプの位置から見たときに、目に見える頂点のある場所から判断して最適値を決めます。

Hints

(ヒント)

Blender のオブジェクトはどれも、3D ビュー内でカメラとして振る舞うことができます。Spot 光源を選択して Ctrl0 NumPad を押し、この光源の視点からのビューに切り替えてください。

Spot Volumetric Effects



Spot ランプの Halo オプション

Spot(スポット) 照明は "volumetric" な効果を作ることもできます。この意味については [ボリュームメトリックライト](#) をご覧ください。

Halo(ハロー)

Halo ボタンは Spot ランプに volumetric 効果を使えるようにします。ボリュームメトリック効果が目に見えるためには、このボタンが有効でなければいけません。なおバツファードシャドウを使っている場合は、[スポットランプのバツファシャドウ](#) ページで説明されている別のオプションも使えます。

Intensity(強度)

Intensity スライダーは光源から生成されるボリュームメトリック効果の強さ/濃さを調整します。Intensity スライダーの値が低いほど目に見えるボリュームメトリック効果は減り、一方 Intensity 値が高いほどはっきりした濃いボリュームメトリック効果を生みます。

Step(ステップ)

この欄には **0** から **12** までの値を入力できます。この Spot がボリュームメトリックシャドウを作るかどうかと、影の質を決めるのに使われます。

Step が値 **0** に設定されている場合は、ボリュームメトリックシャドウは生成されません。

他のほとんどのコントロールと違って、Step 値が増えるほどボリュームメトリックシャドウの品質は下がります(レンダリング時間は減ります)。逆も同様です。

**Step 値**

普通は Halo Step に **8** を使うとうまく速度と精度の折りがつきます。

Blender では 内蔵レンダラーを使っているときの Spot ランプだけがボリュームメトリック照明をシミュレートします。光源の Energy や halo の Intensity の設定の組み合わせによっては変わった結果を引き起こします。

たとえば、無効な Spot 光源あるいは非常に低い Energy 設定の光源と Intensity 設定が非常に高い halo で、現実世界ではありそうにない暗くて黒い halo ができます。内蔵レンダラーと halo を使うときは、こうした可能性に注意してください。

備考

halo のエフェクトはバツファードシャドウを使うと大きく強化されます: halo の Step が空でなければ"volumetric shadows"を作れます。さらなる情報は Spot [バツファシャドウ](#) をご覧ください。

See Also

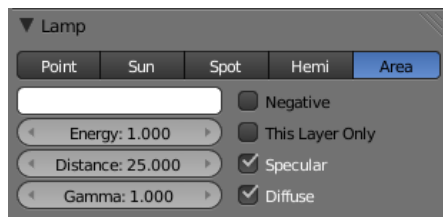
(参考)

- [影](#)
- [スポットランプ](#)
- [スポットランプのバツファシャドウ](#)

Area Lamp

(エリアランプ/面光源)

Area (エリア) ランプは、ものの表面(あるいは面状のもの)から発せられる光をシミュレートします。たとえばテレビ画面、スーパーマーケットのネオン照明、窓、曇り空はエリアランプのほんの一例です。エリアランプは、指定した大きさのグリッドに沿って光源をサンプリングすることで、エリアランプは境界線の柔らかな影を作ります。これは、ポイントランプのような人工的な鋭い境界線を作るものとは対照的です。



共通オプション

Lamp options

(照明のオプション機能)

Distance(距離)、Energy(エネルギー)、Color(色)

ほとんどのタイプの照明に共通の設定で、[照明のプロパティ](#) で説明されています。

なお エリア 光源の Distance 設定は、他の照明に比べてはるかに感度が高く重要です。通常は Distance の範囲内にあるオブジェクトは露出しすぎた状態になります。よい結果を生むために Distance を照らしたいオブジェクトとの距離よりわずかに少ない値にしてください。

Gamma(ガンマ)

照明の明度をガンマ補正する量です。値を高くするほどコントラストが増え、減衰が短くなります。

エリア 光源には光の減衰の設定がなく、「二次関数の逆関数」方式で減衰します。減衰を調整する唯一の方法は、Distance と Gamma のいずれか、または両方の設定を使うことです。

This Layer Only(このレイヤーのみ)、Negative(負)、Specular(スペキュラー) および Diffuse(ディフューズ)

光源の作用対象を決める設定です。[照明の作用対象](#) で説明されています。

Shadows

(影)

エリア光源のレイトレースによる影については [レイトレースによる影](#) で説明されています。

エリア 光源を選ぶと、Shadow パネルは次のデフォルト配置になります:



Adaptive QMC の設定
Area 光源選択時の Shadow パネル

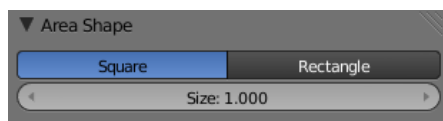


Constant Jittered の設定

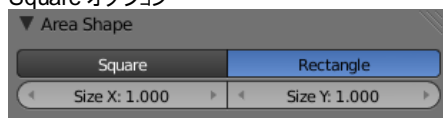
Area Shape

(エリアランプの形状)

エリア光源の形状は Square(正方形) か Rectangle(長方形) にできます。



Square オプション



Rectangle オプション

Square(正方形)/Rectangular(長方形)

正方形または長方形の領域から光を放射します

Size/Size X/Size Y

Square や Rectangle の寸法です

形状のヒント

エリア光源に適切な形を選ぶことで、シーンの本物らしさを増すことができます。たとえば屋内のシーンに窓から差し込む光をシミュレートしたいとします。正しい比率の X と Y を持つ長方形のエリア光源を、窓の中に(垂直方向に)(またはネオンからの光は水平方向に)置くとよいでしょう。テレビ画面から放たれる光をシミュレートするには、たいていの場合、垂直方向に正方形のエリア光源を置くとうまくいくでしょう。

Area Raytraced Shadows

(エリアランプのレイトレースされた影)



Adaptive QMC 設定

Area 光源はレイトレースによる影のみを作ります。レイトレースによる影の設定のほとんどは、[レイトレースによる影のプロパティ](#) で説明されているように、他の影の設定と共有されます。Area 光源専用の設定は以下に詳しく述べます:

Shadow Samples

(影のサンプリング)

Samples (サンプル数)

他の光源と同じ役割を持ちますが、長方形の Area 光源を使うと、面の縦横それぞれの方向に対するサンプリング値 Samples X と Samples Y を設定できます。

Constant Jittered (コンスタントジッター) のサンプル生成方式を使うと、これは領域内にある仮想的な光源の数とだいたい同じになることに注意してください。QMC サンプル生成方式を使うと、Lamp や Spot 光源に似た挙動になります。

Sample Generator Types

(サンプル生成の方式)

Adaptive QMC (アダプティブ QMC); Constant QMC (コンスタント QMC)

これらの共通設定は [影のプロパティ](#) で説明されています。



Constant Jittered 設定

Constant Jittered (コンスタントジッター)

Area 光源には三番目のサンプル生成方式である Constant Jittered があります。これは光の並びをシミュレートするようなものです。以前の方式と同じオプションを持ちます: Umbra、Dither、Jitter。

次の三つのパラメータは Constant Jittered サンプル生成方式を使うときにのみ利用可能で、品質の低下と引き換えに、人為的に「柔らかな」影の効果を増大させるものです:

Umbra (暗部強調)

影光線の完全に内側にある影の強さを強調します。完全に影のできた領域と完全に照らされた領域間の光の遷移がより速くなります (影のグラデーションが鋭くなります)。効果を確認するためにはこのボタンの Samples 値を 2 以上に設定する必要があります。

Dither (ディザ)

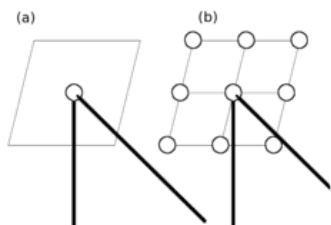
OSA ボタンでオブジェクトの境界線にアンチエイリアスが適用されるのと同じやり方で、影の境界線上のサンプリングに適用されます。人為的に影の境界線を柔らかくします。Samples が非常に低く設定されているとよい結果を望めません。Ditherは中程度の Samples 値で使ってください。高い Samples 値を使っていれば境界線はすでに柔らかいので、無意味です。

Jitter (ジッター)

硬い影のサンプルの、境界線を散らすためにノイズを加え、擬似乱数を使って位置をずらします。このオプションも高い Samples 値を使うと無意味で、ノイズが粒状性を非常に目立たせる欠点があります。

Technical Details

(技術的な詳細)



Area 光源の仕組み

柔らかな影のシミュレート方法をわかりやすく説明するため、図「Area 光源の仕組み」を使います。

(a) は Blender における Area 光源です。形が正方形なら、影の柔らかさは縦横方向それぞれにある光の サンプル 数によって決まります。たとえば 図 (b) は、Shadow パネルで Samples を 3 に設定した Area 光源に相当します。

こうして Area 光源は、合計9つの光源に対して、各方向に解像度「3」を持ち、各節に「頂点を複製した」格子状の光とみなされます。

(a) の例ではエネルギー (E) は $E/1$ になり、(b) の例では擬似光源それぞれのエネルギーは $E/(\text{光源数})$ に等しくなります。各擬似光源はエネルギーに比例した弱い影を作り、影の重なる部分が柔らかな影を作ります (影の重なった部分は濃くなり、その他の部分は薄くなります)。

Hints

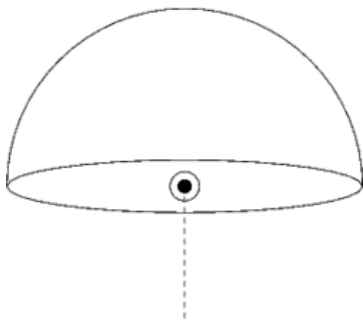
(ヒント)

Area 光源の Size パラメータを変えても、シーンへの光の強さに影響しないことに気がつかれたかもしれません。一方、3Dビューで S を使って光源の大きさを変えると、シーンへの光の強さが激しく増減します。このようにプログラムが記述されている理由は、光源の強さの激しい変化に悩まされずに全光源の設定を微調整し、シーン全体を拡大縮小できるようにするためです。Area 光源の強さをいじらずに大きさだけを変えたいときは、代わりに Size ボタンを使うべきです。

それほど高速ではないコンピュータをお使いなら、Constant Jittered サンプル生成方式を使う場合、2 などの低い Samples 値を使い、Umbra、Dither、Jitter のいずれか、または全部を有効にしてわずかに柔らかな影をシミュレートのようにすることが役立つかもしれません。しかしながら、同じ光源の Samples 値を高くしたものを超える結果にはならないでしょう。

Hemi Lamp

(ヘミ/半球ランプ)



Hemi 光源の概念図

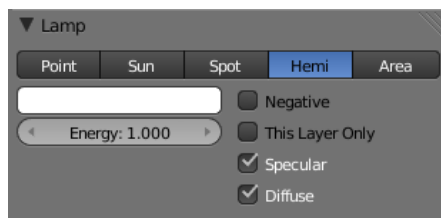
Hemi (ヘミ) ランプは180度の半球の向きに光を作ります。厚く雲で覆われた空、あるいは一様な空から届く光をシミュレートするように設計されています。言い換えれば、シーンを覆う光った円天井から一様に発せられる光です。

Sun (サン) ランプと同じく、ヘミの位置は重要ではありませんが、方向は重要です。

ヘミランプは4つの円弧と、半球状のドームの方向を視覚化したもの、それに半球内部で最も大きなエネルギーが放射される向きを示す破線で表されます。

Options

(オプション)



Hemi ランプのパネル

Energy (エネルギー) と Color (色)

この設定はほとんどのタイプのランプで共通で、[照明のプロパティ](#) で説明されています。

Layer (レイヤー)、Negative (負)、Specular (スペキュラー)、Diffuse (ディフューズ)

この設定はランプの影響するものを調整します。[照明の影響するもの](#) で説明されています。

ヘミランプに光の減衰の設定はありません。常に一定の減衰を使います (すなわち減衰しません)。

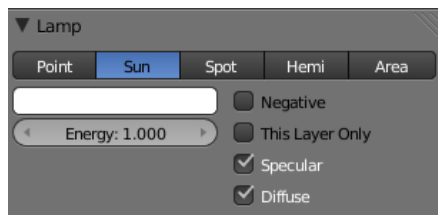
このランプは影を投じない唯一のランプで、Shadow (影) パネルはありません。

Sun Lamp

(サンランプ)

Sun(サン) ランプは単一方向に一定の強さで放射される光をもたらします。サンランプは一様に明るい日光による、開けた空間の照明に非常に便利です。サンランプは 3D ビューでは八方に光を放つ丸囲みの黒点と、光の向きを示す破線で表されます。

他のオブジェクトと同様に サンランプ を回転させると向きを変えられますが、光が一定の向きに放射されるため、サンランプ の位置はレンダリング結果に影響しません ("[Sky & Atmosphere \(空と大気\)](#)" オプション を使う場合を除く)。



Sun ランプのパネル

Lamp options

(ランプのオプション)

Energy(エネルギー) と Color(色)

この設定はほとんどのタイプのランプに共通で、[照明のプロパティ](#) で説明されています。

Negative(負)、This Layer Only(このレイヤーのみ)、Specular(スペキュラー)、Diffuse(ディフューズ)

この設定は [照明の影響するもの](#) で説明されているように、ランプの影響するものを調節します。

サンランプ には光の減衰の設定はありません。常に不変の減衰を使います (減衰しません！)。

Sky & Atmosphere

(空と大気)

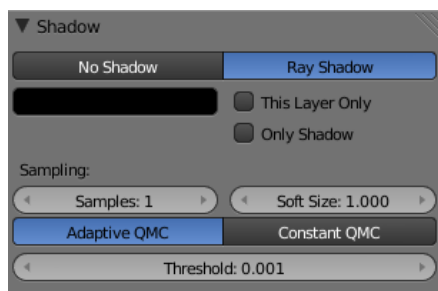


Sky & Atmosphere(空と大気) パネル

空にある太陽の見た目や、日光が通過する大気に関する様々な設定が利用できます。詳しくは [Sky and Atmosphere \(空と大気\)](#) をご覧ください。

Shadow

(影)



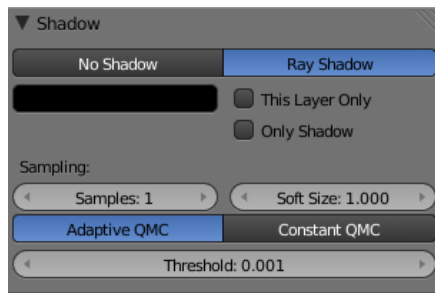
Shadow パネル

サン 光源はレイトレースによる影だけを作ることができます。影付けのオプションは [影のプロパティ](#) で説明されているように、他のタイプのランプと共通のものを使います。

このランプのレイトレースによる影の設定は他のランプと共通で、[レイトレースによる影のプロパティ](#) で説明されています。

Raytraced Shadows

(レイトレースによる影)



Shadow パネル

Sun(サン) 光源はレイトレースによる影だけを作ります。影付けの共通オプションは、[影のプロパティ](#) で説明されているように、他のタイプのランプと同じものを使います。

レイトレースされた影の設定は他のタイプと共通で、[レイトレースによる影のプロパティ](#) で説明されています。

Sun: Sky & Atmosphere

(太陽: 空と大気)

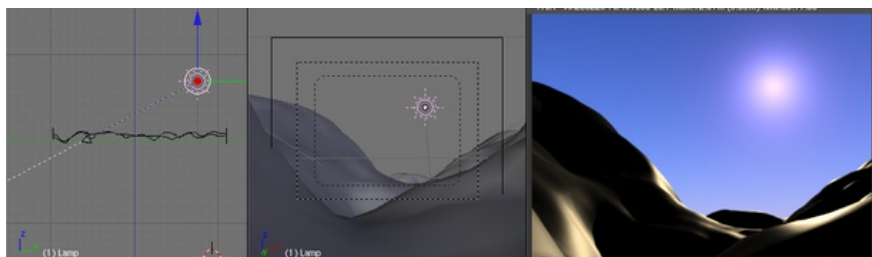


Sky & Atmosphere パネル

このパネルでは現実の空と大気さまざまな属性を(日光の拡散光が数キロメートル上空を横切るように)シミュレートする効果を有効にできます。たとえば太陽が高いと空は青くなります(そして地平線は白みがかかります)。太陽が地平線に近いと空は濃い青や紫になり、地平線はオレンジに変わります。大気の分散は少し霧が出ているとよく目立ちます。ものは遠くなるほど薄い灰色に「色褪せます」...これは天気の良い暑い日に田舎に出かけるとわかります。

この効果を有効にするには Sun(サン) 光源を使う必要があります。いつものようにランプの **位置** は重要ではなく、**回転** が極めて重要です。回転は現在の時刻を決めます。初めに Sun の回転をリセットしましょう(AltR または Transform Properties (トランスフォームのプロパティ) パネル(N) の Rotation (回転) の3つの欄 X/Y/Z にそれぞれ **0** を入力します。この方法で正午の(熱帯の)太陽ができます。

ところで Sky/Atmosphere エフェクトには二つ、重要な角度があります。「入射」角(光の向きと X-Y 平面)は「時刻」を決めます(デフォルトの回転である真下は「正午」、光が真上を向くと「真夜中」といった具合です)。Z 軸 周りの回転はカメラを中心にした太陽の位置を決めます。



カメラの焦点と交差する Sun ランプの破線の「光線」

実際のところ、世界のどこに太陽を置けばいいか決めるには、カメラ視点の 3D ビューで、カメラ中央(焦点)にランプの破線の「光線」を常に交差させるようにするのがよいでしょう(図「カメラの焦点と交差する Sun ランプの破線の「光線」」)。この方法を使うとカメラビューに(0 NumPad、図の中央のウィンドウ)この効果が作る「仮想的な」太陽の場所がわかります。

太陽の **位置** がこのエフェクトに関係しないことを理解するのは大切です。太陽の **回転** だけが関係します。位置はシーン設計で役立つかもしれないという程度です。

Options

(オプション)

Sun & Sky Presets (太陽と空のプリセット)

- Classic(クラシック):
- Desert(砂漠):
- Mountain(山):

Sky

(空)

Sky(背景)

空の設定を有効にします。「太陽」が見えていれば太陽とともに「空」を作り、World(ワールド) 設定で定義された背景と混ぜ合わせます。

Turbidity(濁度)

太陽の見た目、空、大気に作用する汎用パラメータです。低い値は晴れた空を、高い値は曇り空を表す大気のパラメータです。通常は低い値は小さな太陽と晴れた真っ青な空を作ります。高い値は大きなかさを持った太陽と、もっと赤味を帯びた空を作ります。なおこのパラメータは Sun(サン 照明の「強度」)を実際に編集できます。下の例をご覧ください。

以下はこのオプション特有のコントロールです:

Blending(ブレンド)

- 最初のドロップダウンリストはさまざまなブレンド方式を示します。選択した方式は、空と太陽を World (ワールド) 設定で定義された背景と混ぜ合わせるために使われます。ブレンド方式はたとえば [Mix Compositing Node \(eng\)](#) ページで説明されているものと同じです。
- Factor(強さ)

空と太陽の効果が World 背景に適用される量を調整します。

Color space(色空間)

効果が使用する色空間を次から選びます:

- CIE
- REC709
- SMPTE
- Exposure(露出)

この数値欄はレンダリングされた空と太陽の露出を編集します(0.0なら補正なし)。

Horizon(水平線)

- Brightness(明るさ)

地平線の色の明るさを調整します。値は 0.0 から 10.0 の範囲内にすべきです。ゼロに近い値は地平線に明るさがないことを示し、大きな値にするほど地平線は明るくなります。下の例をご覧ください。

- Spread(広がり)

地平線の光の広がりを調整します。値は 0.0 から 10.0 の範囲内にすべきです。この範囲内で小さい値ほど地平線の光の広がりは狭く、大きな値は空全体に広がります。

Sun(太陽)

- Brightness(明るさ)

太陽の明るさを調整します。値は 0.0 から 10.0 の範囲内にすべきです。低い値は空に太陽がないことを、高い値は太陽だけがあることを表します。

- Size(サイズ)

太陽の大きさを調整します。値は 0.0 から 10.0 の範囲内にすべきですが、低い値は大きな太陽、高い値が小さな太陽を表すことに注意してください。なお太陽の全体の明るさは一定のまま(Brightness で設定した値)です。したがって太陽が大きいくほど(Size が小さいほど)太陽は空に「溶け込みます」。逆もまた同様です。

- Back Light(バックライト)

「後方散乱光」用の太陽の色に生じる効果で、値が高いほど太陽周囲の光が増えます。値の範囲は -1.0 から 1.0 です。負の値は太陽周囲の光を減らします。

Atmosphere

(大気)

Atmosphere(大気)

大気の設定を有効にします。背景を編集せずに、大気中の太陽光の散乱やその減衰など大気の効果を実ミュレートしようとします。

Intensity(強度)

- Sun(サン)

太陽の強度を設定します。値は 0.0 から 10.0 です。値が高いほど遠くのオブジェクトに青色の光が当たります。

- Distance(距離)

この係数は Blender units を大気エフェクトの理解できる単位に変換するのに使われます。0 から開始し、高い値ほどシーンの黄色の光が増えます。

Scattering(散乱)

- Inscattering(内散乱)

カメラとシーン内オブジェクト間の大気に光が内散乱する効果を減らすのに使える係数です。この値は 1.0 であるべきですが、変更して素敵だが現実味はない画像を作ることができます。

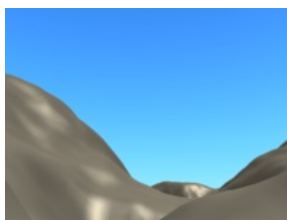
- Extinction(消滅)

シーン内オブジェクトからの減衰光の効果を下げるのに使える係数です。Inscattering 係数のようにこのパラメータは 1.0 であるべきですが変更することができます。低い値は光の減衰を少なくします。値の範囲は 0.0 から 1.0 です。

Examples

(例)

まず太陽の方向を変えると何が起きるか見てみましょう:



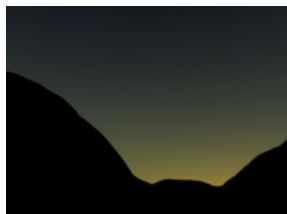
太陽が真上にある状態(正午)



太陽が「地球の下」深くにある状態(真夜中)



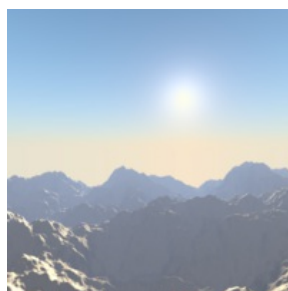
地平線のわずかに上にある太陽
(黄昏のはじまり)



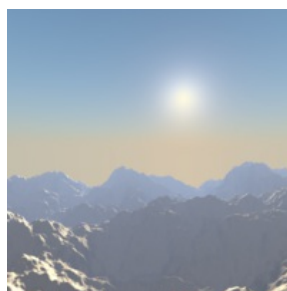
地平線のわずかに下にある太陽
(黄昏の終わり)

Sunの向きによる変化。Sun Size は 5.0、他の設定はすべてデフォルト
[この例の 2.4 .blend ファイル](#)。

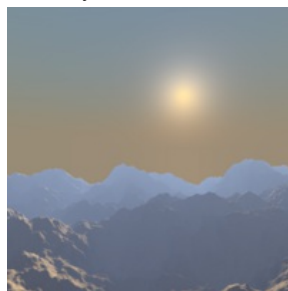
それでは次に、さまざまな設定による効果です(例は [この 2.4 .blend ファイル](#) で作りました):



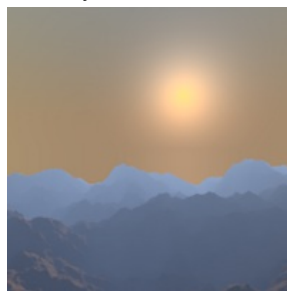
Turbidity: 2.0.



Turbidity: 2.3.



Turbidity: 5.0.

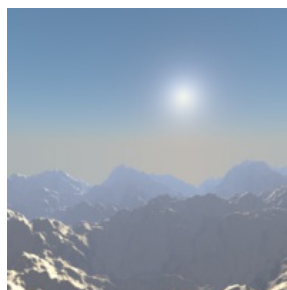
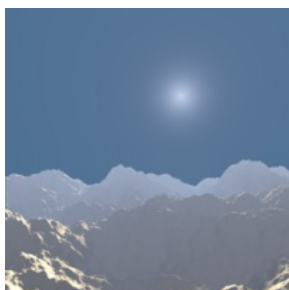


Turbidity: 10.0.

Turbidity パラメータによる変化。他の設定はすべてデフォルト

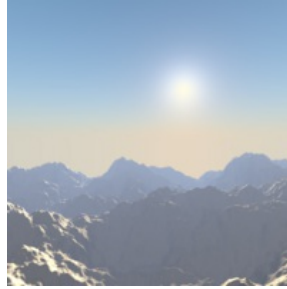
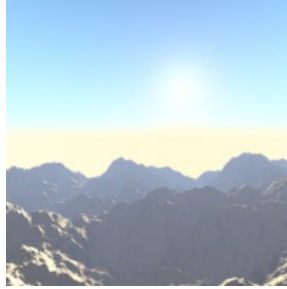
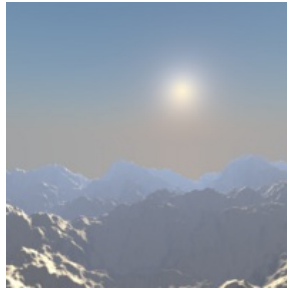
Sky

(空)

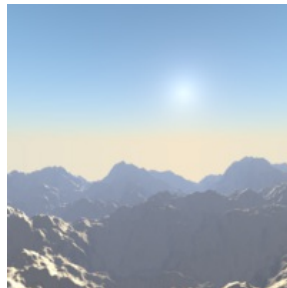
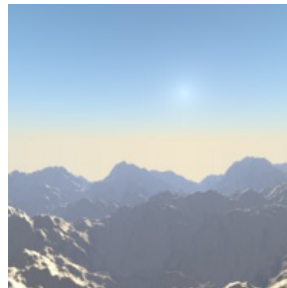
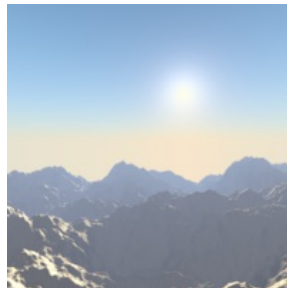
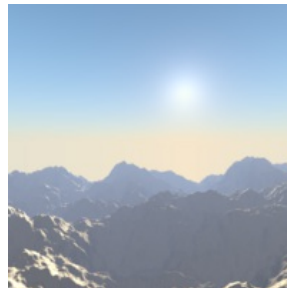


Horizon Brightness: **0.0**.Horizon Brightness: **0.85**.Horizon Brightness: **1.04**.Horizon Brightness: **1.13**.

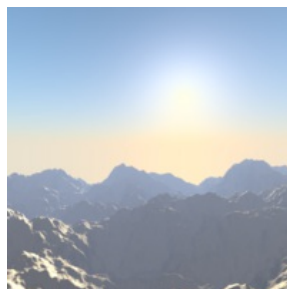
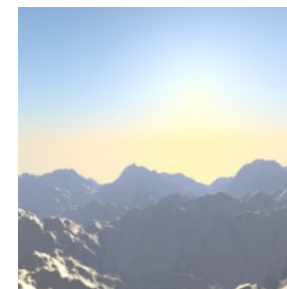
Horizon Brightness パラメータによる変化。他の設定はすべてデフォルト

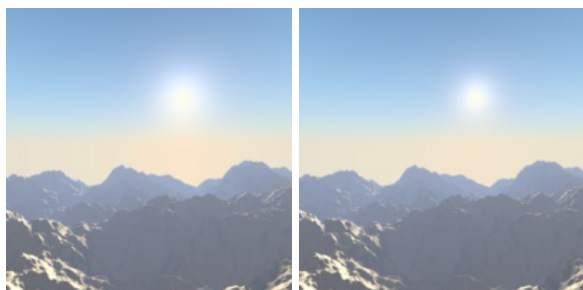
Horizon Spread: **0.7**.Horizon Spread: **1.2**.Horizon Spread: **2.2**.Horizon Spread: **5.0**.

Horizon Spread パラメータによる変化。他の設定はすべてデフォルト

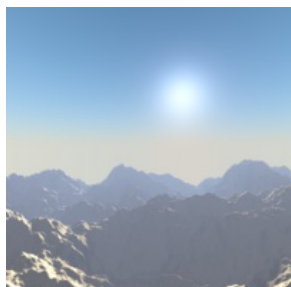
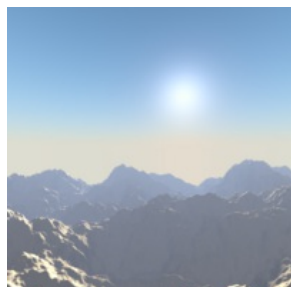
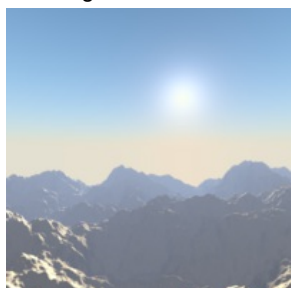
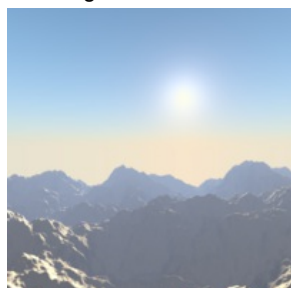
Sun Brightness: **0.2**.Sun Brightness: **0.5**.Sun Brightness: **0.75**.Sun Brightness: **1.0**.

Sun Brightness パラメータによる変化。他の設定はすべてデフォルト

Sun Size: **2.0**.Sun Size: **4.0**.

Sun Size: **7.0**.Sun Size: **10.0**.

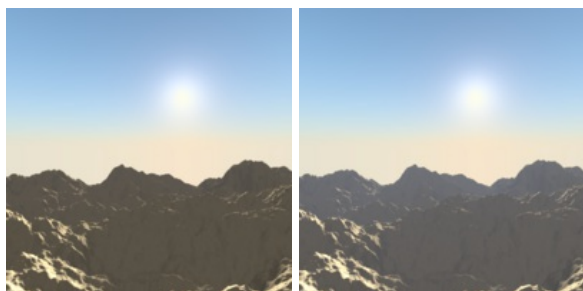
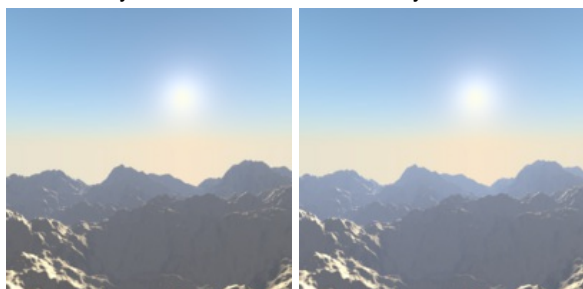
Sun Size パラメータによる変化。他の設定はすべてデフォルト

Back Light: **-1.0**.Back Light: **-0.33**.Back Light: **0.33**.Back Light: **1.0**.Back Light パラメータによる変化。Sun Bright は **2.5**、他の設定はすべてデフォルト

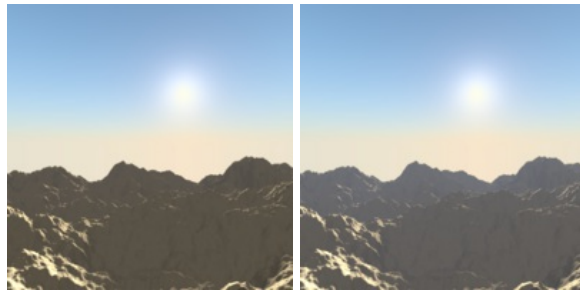
Atmosphere

(大気)

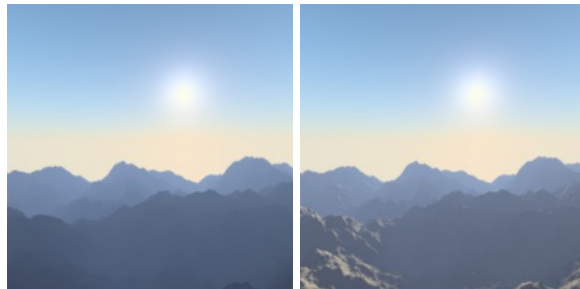
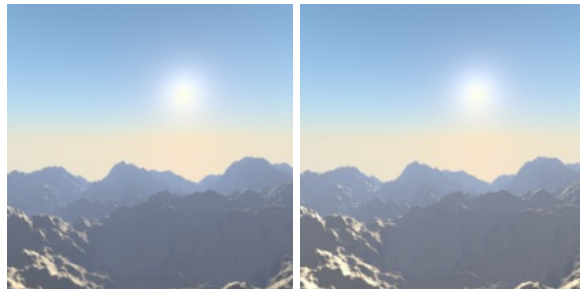
この下のレンダリング画像はどれも Hor.Bright が **0.2** に、Sun Bright が **2.0** に設定されています。

Sun Intensity: **1.0**.Sun Intensity: **3.33**.Sun Intensity: **6.66**.Sun Intensity: **10.0**.

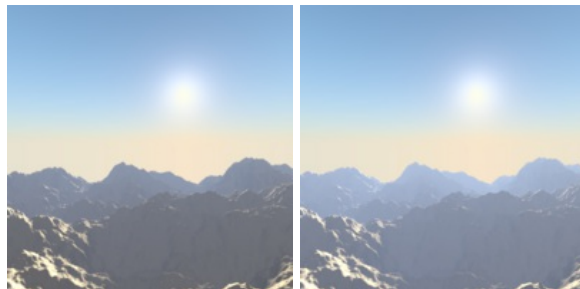
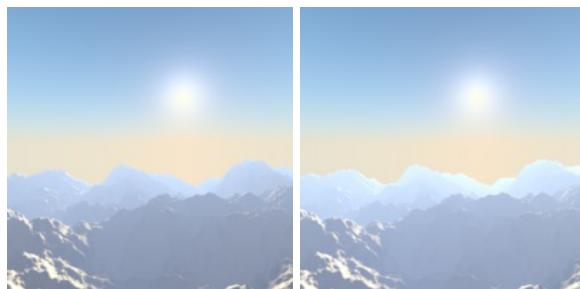
Sun Intensity パラメータによる変化。他の設定はすべてデフォルト

Inscattering: **0.1.**Inscattering: **0.33.**Inscattering: **0.66.**Inscattering: **1.0.**

Inscattering パラメータによる変化。他の設定はすべてデフォルト

Extinction: **0.0.**Extinction: **0.33.**Extinction: **0.66.**Extinction: **1.0.**

Extinction パラメータによる変化。他の設定はすべてデフォルト

Distance: **1.0.**Distance: **2.0.**Distance: **3.0.**Distance: **4.0.**

Distance パラメータによる変化。他の設定はすべてデフォルト

Hints and limitations

(ヒントと制限事項)

Sun(サン) ランプが常にカメラの中央を指すように、サンオブジェクトに [TrackTo\(トラック\) コンストレイント](#) を使えます。ターゲットとしてカメラを指定し“To”軸を -Z にします (“Up” 軸には X か Y のどちらかを使ってください)。こうするとただ動かすだけで、レンダリング画像内での太陽の高さや位置の編集ができます。方向はコンストレイントにより自動的に処理されます。カメラそのものが動いていれば [Copy Location\(位置コピー\) コンストレイント](#) も Sun(サン) ランプに追加するとよいでしょう。カメラをターゲットにして Offset(オフセット) オプションを有効にします。こうすると太陽光源はカメラが動きまわっても変化しません。

デフォルトの Blending(ブレンド) である Add(追加) を使う場合は、正しい「夜」を作るためワールド色にととも濃い青を使うのがよいでしょう。

この効果は Hemi(ハミ) ランプや Sun の影を埋めるアンビエントオクルージョンとともうまく連携します。

大気シェイディングは現在、反射や屈折を不正確に加えます。またソリッドな陰影づけがされた面だけをサポートします。これは将来のリリースで解決される予定です。

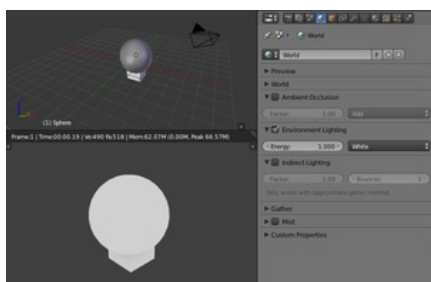
Lighting Rigs

(ライティングリグ)

リグとは広く使われるオブジェクトの配置や組み合わせです。ライティング用のリグ、アーマチュアのリグなどがあります。リグによる基本配置を使えば、よく知っている工程から作業を始めることができます。リグによって用途や再現する状況は違います。シーンで伝えたいものによって使うべきリグは変わります。ライティングは非常に入り組んだものになり、デフォルトのままではよい結果になりません。その上非常に小さな変化が雰囲気や色に大きな影響を与えることがあります。大手のスタジオではライティングを専門技能として、一つの完全な工程として扱います。さて混沌の闇を消し去りましょう。解説させていただきます。

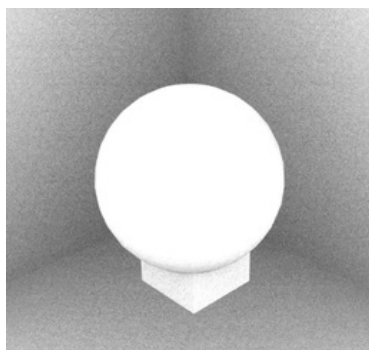
どのライティングリグもデフォルトカメラはロングレンズ(80mm)を使って常にどんびしゃから 15 度ずらし、被写体の約 25 BU (Blender Units) 後ろ、9 BU 横の目の高さで設置します。35mm レンズでは近づくと絵が歪みます。長尺のレンズはシーンの多くをとらえます。どんびしゃのカメラ角度は上げすぎますし、シーンが広くなりすぎてフレームに取り込めません。さてこれでおわかりのように、次回から観劇の際は中央より外れた席に座ったほうが、脇で行われた演技を見逃すこともなく、舞台装置の奥行きも堪能できます。ともあれカメラアングルの話はもう十分です。これはライティングの話ですから。

Environment or Ambient Only



Environment (Ambient) lighting only.

World(ワールド) コンテキストにある Environment Lighting (環境照明) パネルでは、シーンの Environment 照明、Ambient 照明を有効にできます。Ambient 照明とは日光が面に衝突するたびに反射し散乱した光で、オブジェクトに衝突してカメラにも届きます。

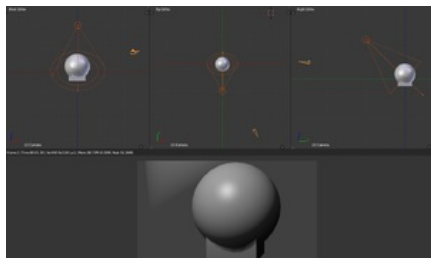


アンビエントオクルージョン

Ambient light は影を作らず、陰のない完全に均衡のとれた方法で明るくします。シーン全体の ambient light の強さは [アンビエントオクルージョン \(eng\)](#) を使って変更されます。Ambient の色は明るい白です。

Single Rig

(単一リグ)



標準的な スポット 光源のリグ

スポットライトのリグは唯一のライト、あるいはキーライトとして使われ、単独か近くに固まった数個の物体を照らして、印象的で人目を引く効果を生みます。このリグは単独の Spot (スポット) 光源で、通常は硬い輪郭を持ちます。この画像で Halos (ハロー) を有効にすれば煙たちこめるナイトクラブの場面を想起させます。この光源は被写体の正面斜め上に設置されます。図ではちょうど舞台のように 10 BU 前の 10 BU の高さであり、下方に約 40 度の範囲を照らします。Quadratic(二次式)の減衰を使いました。

Spot Shape(スポット形状) パネルでは Size(サイズ) を上げるとスポットランプを広くすることができ、Blend(ブレンド) を上げるとやわらかな輪郭にできます。スポットを主演の子にすれば、主演の動きを追うことができます。主演の近くにある物は自然に明るさが増し、観客の注意を惹きます。

スポットを頭の真上に動かして下を向ければ尋問効果を作ります。やわらかなろうそくの光(短い減衰と黄色の光を持つ Point(ポイント) ランプ)を被写体に非常に近づけると自己顯示と正反対の感情を表し、ぞっとする「暗闇に迷った」効果を演出します。

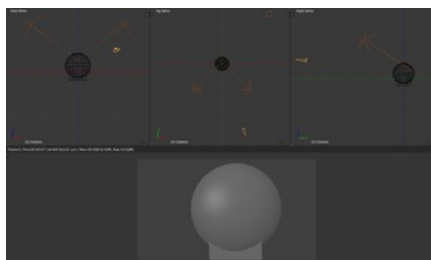
床の上の硬いものが上向きに照らされると、恐怖の感情が湧くことがあります。試しに懐中電灯を持って浴室に入り、ドアを閉めてください。明かりを消してあごの下に懐中電灯を上向きに持ちます。鏡を見ながら電源を入れましょう。お化けだ！ 怖かったですか？ でも怒らないでください、私の言いた

いことは通じたと思います。照明は光源がひとつしかなくても、強さや位置、向きを変えれば、シーンにあるあらゆるものを変化させることができます。

このリグを Environment Lighting (環境照明) (およびマテリアル設定で環境照明を受け取る設定にしてある小道具)といっしょに、単独の人物か物にスポットライトを当てたシーンで使ってください。このリグは開けた広い場所で使うことやモデルのあらゆる面を見せるのには向いていません。

Two-Point Rig

(2点リグ)



Standard two-point light rig.

2点照明のリグはオブジェクトに安定した照明を当てます。右の図は標準的な2点照明のリグの例です。2点とは2箇所に光源があるという意味です。標準的な2点照明のリグは、3D空間にあるテクスチャのないオブジェクトに安定した照明を当てます。実際のスタジオでは商品用の、特に光沢のある商品用の照明にこのリグを使います。

2つの光源はほとんど同じものですが、役割は違います。どちらも Hemi (ヘミ) にすることで非常に広がりのある柔らかな光をエミュレートします。現実世界では銀色の傘の中で光を反射させてこの光源を作ります。

低い Energy (エネルギー) を使って球の立体感を出していることに注目してください。十分には強調できません。強く明るい光は球をかえて平坦にし、見る者を細目にします。やわらかな光は焦点を合わせやすくします。右側の Hemi (ヘミ) はスペキュラを無効にしてあり、額や鼻をそれほど光らせません。

左のランプはスペキュラを有効にしています。スペキュラの光は球の中心より少し上を小さく明るくします。

このリグは主焦点のない場面にむらのない照明を与えるのに使ってください。2つの Hemi (ヘミ) が背景のオブジェクトや小道具を照らし出すため、Environment Lighting (環境照明) はそれほど重要ではありません。照明領域の反対側では硬い輪郭を持つ2つの細い Spot (スポット) ライトが、高出力で被写体に集中砲火を浴びせているかのような (「警察です。両手を上げて出てきなさい」) 見た目を作ります。

Three-Point Rigs

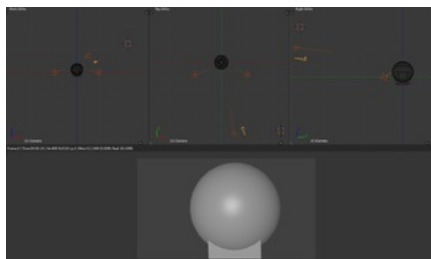
(3点リグ)

物や場면을照らすのに、標準的な3点照明のリグがもっともよく使われることは間違いありません。モデルを目立たせたいときはこのリグを使ってください。ご覧のようにテクスチャもマテリアルもない球体があります。このリグに関する論文は複数あり、次の2つのうち1つがよく使われます:

- スタジオ – 実際のスタジオでグリーンスクリーンや背景幕前での撮影に使われます。俳優と CG オブジェクトに同じ照明が当たるようにアルファ合成してレンダリングするときは、このリグを使ってください。
- 標準 – 現実世界でセット上の俳優を照らすのに使われ、逆光を少し当てて俳優の輪郭を強調して目立たせ、深みを与えます。

Studio rig

(スタジオリグ)



スタジオ3点照明のリグ

右の図は「スタジオ」の、標準的な3点照明のリグの上面図・正面図・側面図です。明るい「キー」ライトで物体にハイライトを与え、側部にある2つの「フィルライト(補助光)」でキーライトの作る影をやわらかくして、場面の持つ迫力を変化させます。

スタジオでは、俳優がグリーンスクリーンの前にいるか他の人といっしょにいるとき、主役にキーライトを当てたまま語り手の顔(俳優)を撮影するときにこのリグを使います。このリグは製品をあらゆる角度から照らすのにも使われ、横にあるフィルライトが小物を照らし出します。

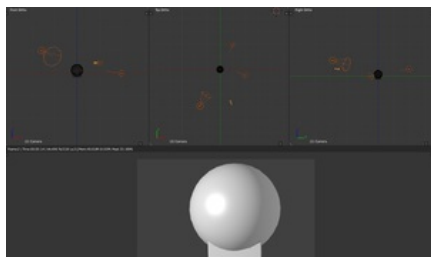
キーライトはカメラの左側、少し上に設置された Area (エリア) 光源です。この光源はスペキュラが見えるようにします。被写体から約 **30 BU** 後ろにあり、カメラにあわせて移動します。小さなスペキュラの輝きがあることで、そこに光源があること、幽霊を見ているのではないことがわかります。現実世界では光の範囲を狭めるバツルやブライндаをともなったスポットライトに相当します。

2つのサイドライトはフィル(補填)専用です。どちらも被写体の前に **5 BU**、横に **20 BU** の地面に設置された Hemi (ヘミ) 光源です。スペキュラは無効にしてあり小さな光点はできません。キーライトの生む影を無効化して、あごより下、あるいはあらゆる横向きの表面を地面から照らします。

このリグを使って安定したやわらかな照明を与えることで、メインとなる人物や物を強調します。単独のリグと2点リグ両方の良い点を組み合わせて、均衡のとれた照明と前面のハイライトを作ります。広い場面に対しては、サイドライトを2点リグのようにやや後退させる必要があるかもしれません。

Standard Rig

(標準リグ)



標準的な3点照明のリグ

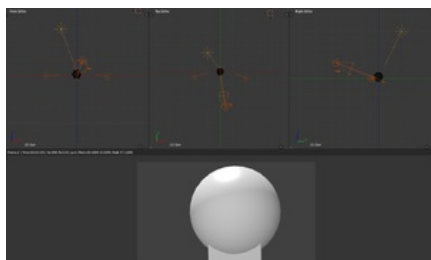
メインの被写体の後ろにカーテンを置かない限り、扱うべき奥行きが存在します。左のフィルライトはカメラに写らない被写体の背後に移動させましたが（したがって呼称はバックライトに変わります）、右のフィルライトは同じままです。キーライトはスペキュラ反射を作るのでオブジェクトのマテリアル設定でスペキュラや硬さを調節できます。キーライトは被写体を強調して「スポットライトを浴びている」感を与え、バックライトは被写体の輪郭を背景に対しくっきりと浮かび上がらせます。これは被写体を目立たせるのに役立ちます。

このリグではキーライトはかなり明るいスポットライトです。この光はとても明るいので薄い黄色にします。こちら側の光源はこれだけです。片方のサイドライトは後ろに移動して目(カメラ)の高さに上げました。このバックライトの Energy(エネルギー) は半分に減らす必要があります。そうしないと残ったサイドライトと足されて、側面を必要以上に照らして目立たせてしまいます。バックライトの角度や高さを変えて、太陽に照らされる様子を模すことができます。

通常の 3D アニメーションで主役を照らすのにこのリグを使ってください。特にガラスのような透明な物体を通してカメラにたくさんの光が射し込む場合は、このリグを使ってください。ここで大変なのは3つの光源がチームとして連携していることを確かめながら、ひとつの光源が他と競合したり強くなりすぎたりしないように光の強さのバランスをとることです。

Four-point Rig

(4点リグ)



4点照明のリグ

4点照明のリグは Sun(サン) ランプを被写体の **30 BU** 上、横に **10 BU**、**15 BU** 後ろに追加することで、屋外の照明をうまくシミュレートします。この日光は逆光を作り、また被写体上部に補助光として働きます。頭の上に作られるまぶしい輝きも太陽が上にあることを知らせます。色を黄色にして青いサイドライトとバランスをとっていることに注目してください。

キーライトを Spot(スポット) に変えるため、Inverse Square(逆二乗式) を選び Specular(スペキュラ) を無効にしてください。そうすれば面を照らすとき、上側の太陽の光と純白の光が混ざってやわらかくなり、明るい日光の効果を得られます。上にふたつ光源があるということは影の鋭さが増すということでもあるので、側部のフィルライトを調整するとよいでしょう。この図ではフィルライトは Specular(スペキュラ) を無効化した Hemi(ヘミ) のままです。

人物の背後から肩越しに撮影するときは、太陽が逆光になるのでこのリグを使ってください。透明なオブジェクトを透過する光がカメラに届く場合にもこのリグを使ってください。

フィルライト用のもう一つのスポットライトは主役の顔を照らし、あごや首より下を明るくします。あご下にできる見苦しい影—修正しないと俳優が太って見えるか二重あごに見える—を取り除ける場合があります。この光源は顔の照明を均一にします。

Troubleshooting

(トラブルシューティング)

妙に明るい部分がある、妙に暗い部分がある、変な影がある、オブジェクトに線が入るなど、レンダリングで問題にぶつかったときは以下の対策を試してみてください:

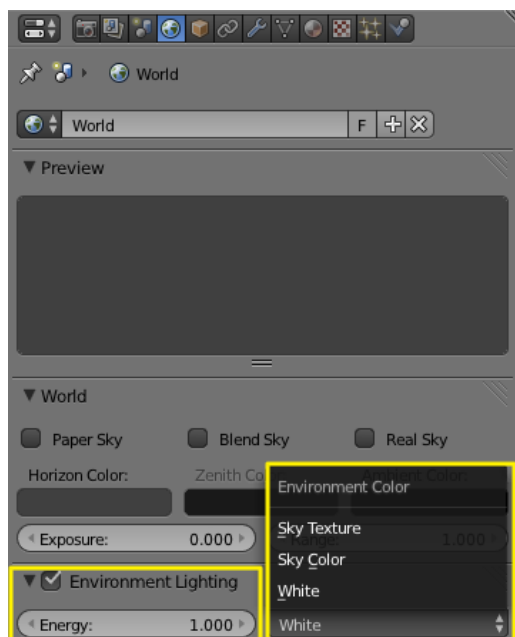
1. まず全てのマテリアルの無効化を試してみましょう(デフォルトの灰色のものを作って Render Layers(レンダーレイヤー) コンテキストの Layer(レイヤー) パネルにある Material(マテリアル) 欄にその名前を入力します。通常のマテリアルに戻すには入力欄を空にするだけです!)。灰色のオブジェクトだけでも同じ問題が起きるか確認します。問題がもう起きなければ、マテリアルに光の作用に関する問題があるわかります。マテリアル設定、特に ambient や reflection、それに Material(マテリアル) コンテキストにある小さなボタンやスライダーを確認してください。光源は特定マテリアルにだけ作用するように設定できるので、数個のオブジェクトだけが明るくなる問題があれば、この設定になっていないか確認してください。
2. 次に光源を「消して」いきます(未使用レイヤーに移動します)。光源が1つになるまでさかのぼって問題がないことを確認したら、1つずつ足していきます。光源が足されるたびに、試し終えた光源の出力を減らしてきれいに溶け込ませるか、まったく足さないようにします。もしくは入れたばかり

- りの照明の Energy(エネルギー) を特に減らします。
3. 照明はあるレイヤーに属するオブジェクトだけを照らすように設定できるので、灰色の球に変なものがあればその設定も同じように確認してください。この場合もうっかり設定してしまった可能性があるため、光源を削除してデフォルトのものを追加しなおし、正しいことがわかっている状態に戻すことが役立つ場合があります。
 4. ネガティブな光源は扱いにくく、モデルがまだらになることがあるため、この特別な光源を使うときは特に注意してください。影のみの光源もシーンの見た目を狂わせます。過剰にテクスチャが使われた光源はシーンをランダムな変わった色にします。青味や黄味を帯びた色や白の陰影からかけ離れたものにはしないでください。そうしないと Material(マテリアル) コンテキストで青く表示されたマテリアルがレンダリングすると緑になるかもしれず、とても混乱するでしょう。
 5. World(ワールド) コンテキストの環境の設定を確認してください: Horizon(水平色) と Zenith(天頂色)、それに Environment Lighting(環境照明) です。

Environment Lighting

(環境照明)

環境照明は全方向から来る光を提供します。光の計算は、アンビエントオクルージョンで使用されているものと同じレイトレーシング手法で行われます。違いは、環境照明では、マテリアルのシェーディングの設定の "ambient(周囲)" パラメータが考慮されることです。これは、マテリアルが受ける周囲光/色の量の指標となるものです。



Environment Lighting (環境照明) パネル

また、環境のカラーソース (白、空の色、空のテクスチャ) や、光のエネルギーを選択することができます。

Energy(エネルギー)

環境照明の強さを決めます。

Environment Color(環境色)

環境照明色の元になる色を決めます。

より良いグローバル照明を達成するために、両方の設定を同時に使用してみましょう。

屋外の光で空を再現するときに効果的です。環境照明は、時にはかなり派手派手しくなることがあります。

Ambient Occlusion

(アンビエントオクルージョン)

アンビエントオクルージョン(Ambient Occlusion、環境閉塞)は高度なレイトレーシング演算で、Ambient Light (周辺光)が塞がれ届かない隅やメッシュの交差、しわ、割れ目に暗がりを作り、やわらかな大域照明の影をシミュレートします。

現実世界には AO のようなものはありません。AO は物理的な正確さのない(しかし通常は見栄えのいい)特殊な描画トリックです。基本的には表面上の各点を中心とした半球をサンプリングしてその半球の何割が他のジオメトリに塞がれているか調べ、それに応じてピクセルに陰影をつけます。

AO は照明とは全く関係ありません。描画トリックでしかない AO が自然に見えるのは、現実世界で間近にある面同士(小さなすきまなど)は影やほこりなどのせいで、前に何も置かれていない面より暗くなるためです。

とは言え AO が行うのはこの結果の近似です。光の跳ね返りや通過のようなものはシミュレートしません。これがシーンにひとつも光源がなくても AO が動作する理由です。また AO を単体で有効にすることがシーンの「照明」方法として非常にまずいのは、このためです。

AO を動作させるためには Render(レンダー) コンテキストの Shading(シェーディング) パネルでレイトレーシングを有効にしておく必要があります。

World(ワールド)コンテキストで Ambient Light Color(周辺光の色)をセットしておかねばなりません。デフォルトでは Ambient Light Color は停電時の真夜中の地下室と同じ黒です。Ambient としてこの色を用いると事実上全ての色を暗くします。屋外の真昼に適した色は RGB (0.9, 0.9, 0.8) で、これは晴れてはいるが日差しはさほどきつくない日中の、白味がかった黄色の太陽の色です。

Options

(オプション)



World(ワールド) パネルの Ambient Color(周辺光の色)

Factor(強さ)

AO 効果の強さで、加算用の係数です。

アンビエントオクルージョンはレンダリング中に合成されます。2つのブレンドモードが利用できます:

Add(加算)

ピクセルは塞がれていない 光線/rayの本数に比例する光を受けます。シーンの明るさは増します。これは大域照明をシミュレートします。

Multiply(乗算)

アンビエントオクルージョンはシェーディング処理に上乘せられ、ものの暗さを増します。

備考

Multiply(乗算)を選ぶ場合は他の光源が必要です。そうしないとシーンは真っ暗になります。Add(加算)ならば明確な光源が存在しなくても AO エフェクトだけでシーンが照らされます。シーンを照らす手軽なショートカットとして AO の単独使用を好む人は多いですが、結果は曇りの日のような色味の抑えられた平坦なものになります。たいていは Blender の標準ランプできちりとシーンを照らすのが一番で、その後で AO をその上に使って、詳細の追加や Contact Shadow 用に「Multiply(乗算)」に設定します。

Gather パネルにはアンビエントオクルージョンの品質に関する設定があります。この設定は Environment Lighting (環境照明)や Indirect Lighting(間接照明)にも使われることに注意してください。

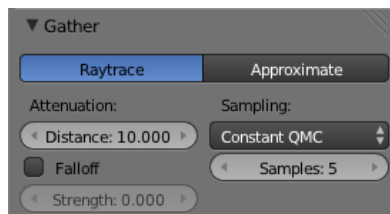
アンビエントオクルージョンの主な計算方式は2つあります。Raytrace(レイトレース)と Approximate(近似)です。

Gather

(ギャザー/集光)

Raytrace

(レイトレース)



アンビエントオクルージョン パネル、Raytrace(レイトレーシング) 方式

Raytrace 方式のほうが正確ですが結果にノイズが増えます。ノイズがほとんどない画像も作れますが、相応のレンダリング時間が必要です。空/Sky のテクスチャの色を使いたい場合は、これが唯一の選択肢になります。

Attenuation(減衰)

光線/ray の長さはどの程度離れた他の面と閉塞効果を作れるか定義します。この距離が長いほど遠いジオメトリまで閉塞効果が及びます。レンダーは Distance (距離) 値が高いほど閉塞するジオメトリを遠くまで探索する必要があるため、満足する効果が得られる範囲でこの距離をなるべく短くするとレンダリング時間を最適化できます。

Sampling(サンプリング)

Samples(サンプル)

オブジェクトが閉塞しているかどうかの検知に使う光線の本数です。サンプル数が多いほど滑らかで正確な結果が得られますが、レンダリング時間は遅くなります。デフォルト値の **5** は通常はプレビューに適しています。実際に射出される光線の本数はこの値の2乗になります (Samples が **5** なら **25** 本の光線)。光線はランダムなパターン(上述のサンプル方式によって決まります)によって半球に対して射出されます。射出された光線が望ましい統計的データを作れるほど多くないと、隣接ピクセルの閉塞パターンに違いがでます。

サンプル数 **3** を使ったアンビエントオクルージョン サンプル数 **6** を使ったアンビエントオクルージョン サンプル数 **12** を使ったアンビエントオクルージョン

サンプリングのオプションが3つあります:

Constant QMC

基本の擬似モンテカルロ法。光線が均一にランダムに分配されます。

Adaptive QMC

QMC の改良版で、サンプルレートを下げたりサンプルをスキップしたりする条件を次の2つの設定を基にして決めます:

Threshold(しきい値)

サンプルが完全に閉塞している(「黒」)か閉塞していない(「白」)とみなしスキップする下限値。

Adapt to Speed(速度に順応)

高速移動するピクセルの AO サンプリングを減らすための係数。Vector(ベクトル) レンダーパスを使うため、このパスが有効である必要があります([レンダーパスのページ](#) をご覧ください)。

QMC について

擬似モンテカルロサンプル方式について、詳しくは [レイトレーシングによる影のページ](#) をご覧ください。

Constant Jittered

昔使われていたサンプル方式です。「bias」アーティファクトを作りやすいです... Bias(バイアス)

半球を狭くする角度です(ラジアン単位)(この半球はもう本当の半球ではありません。半円ではなく、「 π - バイアス値」ラジアン)の円の円弧です)。

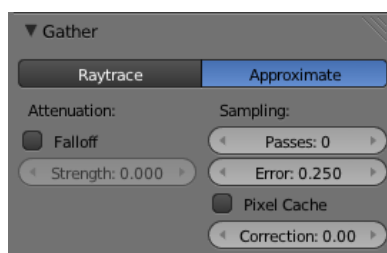
バイアス設定は AO のレンダリングにどれだけ滑らかな「Smooth」な面が表示されるか調整します。AO は Smooth 前の角ばったメッシュに生じるため、「Smooth」がオンになったオブジェクトであっても AO の光で面が見えるようになる可能性があります。これは AO の光線の射出方法によるもので、Bias(バイアス) スライダーで調整できます。なおこの現象は QMC サンプル方式を使っても起きる可能性があります。Constant Jittered を使うほうがはるかに目立ちます。いずれにせよ QMC には Bias(バイアス) オプションはありません。

Bias (バイアス) を **0.05** (デフォルト) にした 24×24 の UV 球。Smooth にセットされているにも関わらず球の表面が角張っていることに注目してください。

Bias を **0.15** に上げると角張りのアーティファクトが消えます。

Approximate

(近似)



アンビエントオクルージョン パネル、Approximate (近似) 方式

Approximate (近似) 方式ではより滑らかな結果を同じレンダリング時間で得られますが、その名の通り Raytrace (レイトレース) 方式の近似でしかなく、アーティファクト (画像の乱れ等) を作る可能性があります。またベースカラーに 空/Sky のテクスチャを使えません。

この方式は結果を「閉塞しすぎる」傾向があるようです。これを抑える補完オプションが2つあります:

Passes (パス)

前処理のパス数を **0** (前処理なし) から **10** の間で設定します。数を増やすほどレンダリング時間が増えますが、アーティファクトや閉塞のしすぎの一部を解消します。

Error (エラー)

近似エラーについての許容係数です (近似結果とフル計算した結果との許容差異)。値が低いほど結果の正確性は増しますがレンダリングは遅くなります。**0.0** から **10.0** の範囲で、デフォルトは **0.250** です。

Pixel Cache (ピクセルキャッシュ)

有効にすると計算したピクセルの値を保持して隣接ピクセルを使って補間します。通常は目立った劣化なしにレンダリングを高速化します

Correction (補正)

閉塞のしすぎを抑えるための補正係数です。**0.0** (無補正) から **1.0** の範囲です。

Common Settings

(共通設定)

Falloff (減衰)

有効にすると閉塞を作るオブジェクト間の距離が影の「深度」に影響します。閉塞を作るジオメトリが離れるほど、「影」は薄くなるということです。この効果は Strength 係数が **0.0** を上回る場合にだけ起きます。これは大気中の光の分散を模したものです。

Strength (強さ)

Falloff (減衰) で有効化された影の減衰を調整します。値が高いほど影は短くなり、減衰は速くなります (霧/ほこりの濃さに相当します)。**0.0** (デフォルト、減衰なし) から **10.0** の値を取ります。

Technical Details

(技術的詳細)

アンビエントオクルージョンの計算は目に見えている点それぞれから光線を打ち、そのうち実質的に空まで届く数と、逆にオブジェクトに遮られる数を数えて行われます。

ある点における光の量は、「抜け出して」空に到達した光線の数に比例します。この判定は影の光線を半球領域のあちこちに発射して行われます。光線が他の面に当たれば (塞がれているので) この光線を「影」とみなし、当たらなければ「光」とみなし。「影」と「光」の比率が指定ピクセルの明るさを決めます。

Hints

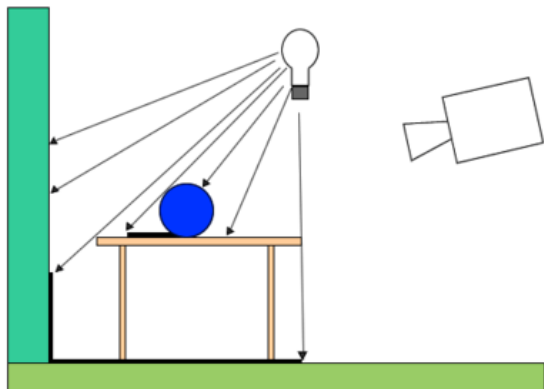
(ヒント)

アンビエントオクルージョンは(少なくとも Raytrace(レイトレース)方式に関しては)光線をたどる技術であるため、処理は遅くなる傾向があります。さらに処理効率は 8分木/octree の大きさに左右されます。詳しくは [レンダリングの章](#) をご覧ください。

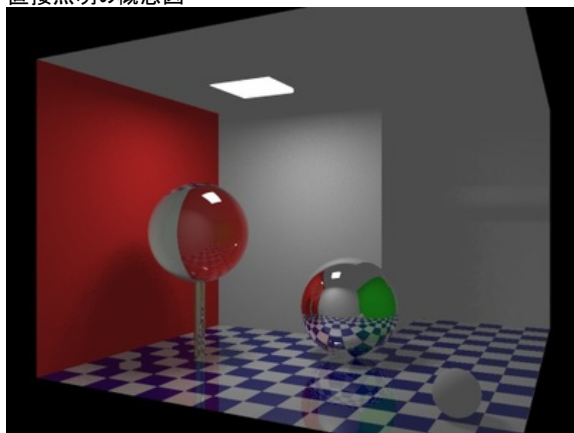
Inderect Lighting

(間接照明)

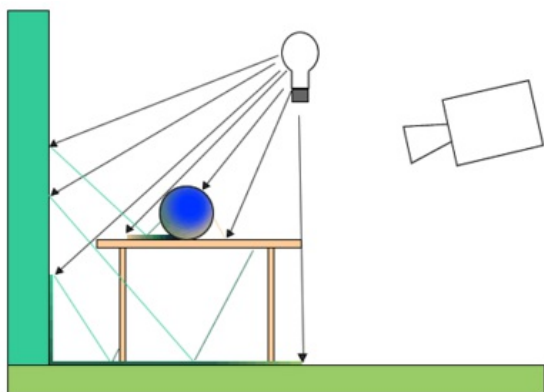
Inderect Lighting (間接照明) は周囲の物体によって跳ね返る間接的な光を加えます。この方法は面に他の面から反射される光をシミュレートします。より大局的な、物理的に正しい方法で、現実味のある画像を生みます。一方計算負荷は増えます。以下は直接光や、直接光と間接光の両方で照らされたシーンの例です:



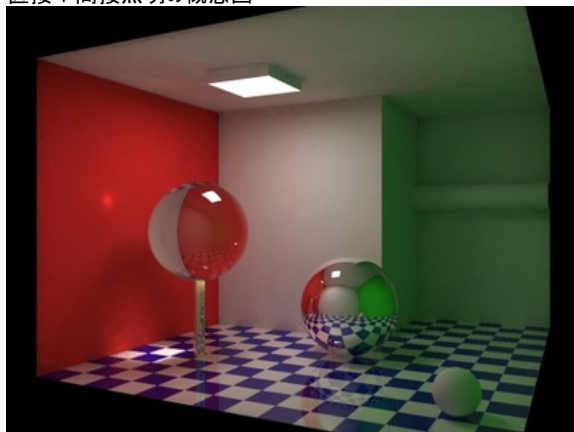
直接照明の概念図



直接照明のレンダリング画像



直接+間接照明の概念図



直接+間接照明のレンダリング画像

[[Images courtesy\(英文\)](#)]

間接照明は Approximate Gather(近似ギャザー)方式でのみ動作します。



Indirect Lighting のパラメータ

Options

(オプション)

Indirect Lighting(間接照明) パネルには2つのオプションがあります:

Factor(強さ)

周囲のオブジェクトが光に影響する度合いを決めます。

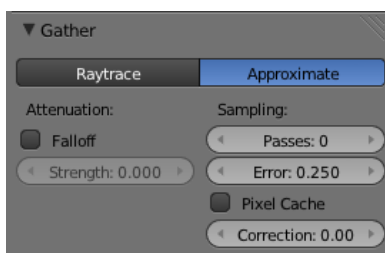
Bounces(バウンス)

間接拡散光が跳ね返る回数。

Gather(ギャザー) パネルは間接照明の品質の設定を含みます。この設定は Environment Lighting(環境照明)や Ambient Occlusion(アンビエントオクルージョン)にも使われる点に注意してください。

Approximate

(近似)



Indirect Lighting(間接照明) パネル、Approximate(近似) 方式

Approximate(近似) 方式では同じレンダリング時間でより滑らかな結果を得られますが、その名の通り Raytrace(レイトレース) 方式の近似にとどまり、画像の乱れ(artifacts)を生む可能性を暗に示しています。また、空のテクスチャを基本色に使うことができません。

この方式は結果を「閉塞しすぎる」傾向があります。これを抑えるため次の2つのオプションを使えます:

Passes(パス)

前処理のパス数を **0**(前処理なし) から **10** で設定します。値が高いほどレンダリング時間は増えますが、画像の乱れ(artifacts)の一部や過剰な閉塞をなくします。

Error(エラー)

近似エラーの許容係数です(近似結果と完全に計算した結果との最大許容値)。値が低いほどレンダリングが遅くなりますが、結果の正確性は増します。**0.0** から **10.0** の範囲で、デフォルトは **0.250** です。

Pixel Cache(ピクセルキャッシュ)

有効にすると、計算されたピクセルの値を保持し、隣接ピクセルを使って補間します。通常は目立った劣化なしにレンダリングを高速化します

Correction(補正)

閉塞過多を抑えるための補正係数です。**0.0**(無補正)から**1.0**の範囲です。

Exposure and Range

(露出と範囲)

モード: 全モード

パネル: World(ワールド) コンテキストの World(ワールド) パネル

Description

(解説)

Exposure(露出)とRange(範囲)はGimpやPhotoshopのツール「カラーカーブ」と似ています。

このコントロールはレンダリングされた画像に作用し、結果はその画像にベイクされます。レンダリングコントロールを使った似た効果の作成方法については [色調補正と露出](#) をご覧ください。

以前のBlenderは、利用できるRGB空間を超えた色を、直接“1.0”(または255)に切り捨てました。これは見苦しい濃淡や度が過ぎたハイライトを引き起こしました(図「露出過度のティーポット」)。

指数関数補正式を使って、現在はうまく補正できます。

Options

(オプション)



ExposureとRangeのスライダー

Exposure(露出)

指数関数の曲率で、**0.0**は直線に、**1.0**は曲線になります。

Range(範囲)

可視光に対応づけられる入力色の範囲(**0.0** – **1.0**)。

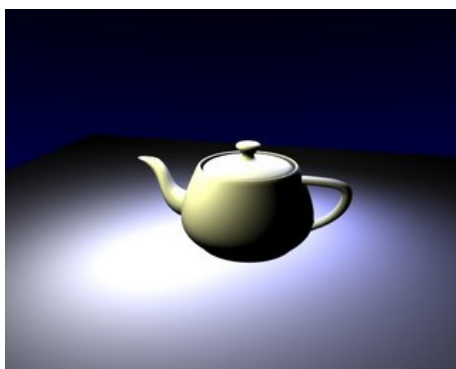
したがってExposure(露出)を使わないとすべての色値は直線的に補正されます:

1. Range(範囲) > **1.0**: 画像は補正前より暗くなります。Range = **2.0**にすると、色値 **1.0**(デフォルトの一番明るい色)は切り捨てられて **0.5**(半分の明るさ)になります(図「Range: **2.0**」)。
2. Range < **1.0**: 画像は補正前より明るくなります。Range = **0.5**にすると、色値 **0.5**(デフォルトの半分の明るさ)は切り上げられて **1.0**(一番の明るさ)になります(図「Range: **0.5**」)。

Examples

(例)

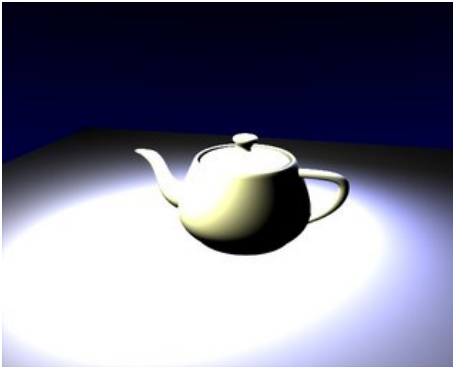
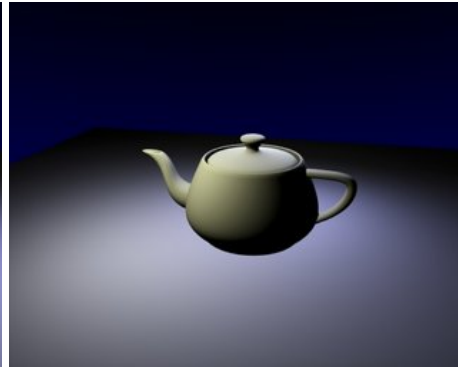
直線的補正を使うとすべての色値が変更されますが、おそらく望んだ結果になりません。Exposureは暗いピクセルを明るくしますが、画像のもっと暗い部分はまったく変更しません(図「Range: **2.0**, Exposure: **0.3**」)。



露出過度のティーポット



Range: **2.0**.

Range: **0.5**.Range: **2.0**, Exposure: **0.3**.

Hints

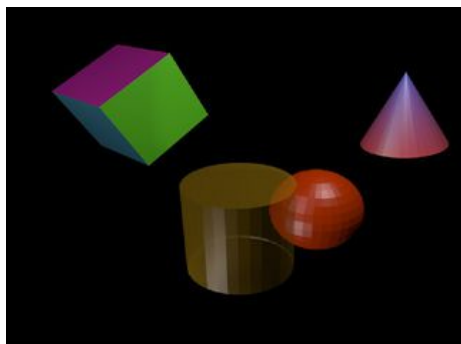
(ヒント)

露出過度の部分がわずかに明るすぎないぐらいになる、最適な Range 値を探してみてください。そうしたら、画像全体の明るさに満足できるまで Exposure 値を上げてください。これは Area Lamps (エリアランプ) に特に役立ちます。

Introduction to Materials

(マテリアル入門)

マテリアルはオブジェクトの材質の芸術性を決めます。マテリアルの一番簡単な使い方として、オブジェクトの材質を表すことやオブジェクトを様々な色で塗ることができます。通常は物質を表面の特性(色、輝き、反射など)で表現しますが、もっと複雑な効果、例えば透明度、回折/diffraction、表面下散乱も表現できます。真鍮、肌、ガラス、リンネルはよく使われるマテリアルです。



様々な基本マテリアル(単独、複数マテリアル、透明度、頂点ペイント)。

Blender の基本マテリアル(テクスチャのないもの)はオブジェクトのどの面にも一様です(ただし照明効果によってオブジェクト各面のピクセルは違って見えるかもしれませんが)。しかしながらオブジェクトの面ごとに違ったマテリアルを使えます([複数のマテリアル](#)をご覧ください)。

Blender ではマテリアルは関連づけられたテクスチャを(オプションとして)持つことができます。テクスチャとは物質を表現するものです。たとえば 磨かれた真鍮、汚れた ガラス、刺繍入りのリンネルなど。[テクスチャ](#) の章ではテクスチャをマテリアルに追加する方法について説明します。

How Materials Work

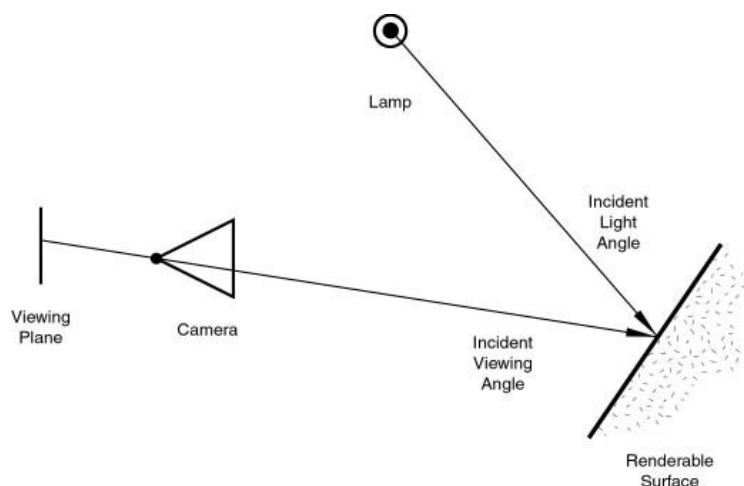
(マテリアルの動作の仕組み)

マテリアルの効率の良いデザインの仕方を理解するには、その前に Blender のレンダリングエンジンでシミュレートされた光と表面の相互作用の仕方と、マテリアルの設定でこの相互作用を制御する方法を理解しなければなりません。エンジンの深い理解は、それを最大限に活用する助けになります。

Blender で作成されるレンダリング画像は *viewing plane* と呼ばれる仮想的な面にシーンを投影したものです。*viewing plane* は従来のカメラにおけるフィルム、あるいは人間の目でいう棹状体と錐状体に似たものですが、現実の光ではなくシミュレートした光を受けます。

シーンをレンダリングするために最初に調べる必要があるのは、シーンからのどの光が *viewing plane* の各点に到達しているのかということです。この問いに答える最善の方法は、*viewing plane* の点と焦点(カメラの位置)を通る直線(シミュレートされた光線)を逆方向に、シーン内のレンダリング可能な面に当たるまでたどることです。そうすればどの光がその点に当たるのかを調べられます。

表面特性と入射光の角度から、入射視線の方向への光の反射量がわかります(図「レンダリングエンジンの基本原理」)。



レンダリングエンジンの基本原理。

面上のどの点においても、光線が当たると二つの基本となる現象が起こります: 拡散反射と鏡面反射です。拡散反射と鏡面反射は、主に入射光の角度と反射光の角度の関係によって区別できます。

レンダリング時のオブジェクトのシェーディング(言い換えれば彩色)は、そのベースの色(拡散反射と鏡面反射現象によって変化したもの)と光強度を考慮に入れて行われます。

内部レイトレーサを使うと別の(より高度な)現象が発生する可能性があります。レイトレースでの 反射/reflection では面に光線が当たると、光線が当たった点はマテリアルの反射率(ベースの色と周囲の環境の混合)と視線角度に応じて、その周囲の環境の色を返します。

一方レイトレースでの 屈折/refraction では、視線角度を歪めるマテリアルの透明度(ベースの色と背景の環境色をオプションのフィルタリング値に従い混ぜたもの)やマテリアルの屈折設定に応じて、光線が当たった点はその背景の環境の色を返します。

光線に当たった物体のシェーディングはもちろん、レンダリング中にほぼこれらすべての現象をいっぺんに混ぜ合わせたものになります。レンダリングしたオブジェクトの見た目は、相互に関連する多くの設定に左右されます:

- ワールド (周辺光、ラジオシティ、アンビエントオクルージョン)
- 照明
- マテリアル設定 (周辺光/ambient、放射/emission、マテリアルコンテキスト内のすべてのパネルの設定)
- テクスチャとその混ぜ方
- マテリアルノード
- カメラ
- 視点角度
- 障害物と透明度の閉塞
- 他の不透明/透明なオブジェクトの影
- レンダリング設定
- オブジェクトの体積 (SS 設定は体積に関係します)
- オブジェクトの形状 (屈折、フレネル効果)

Using Materials

(マテリアルの利用)

レンダリングして確認

マテリアル(およびテクスチャや照明)のデザインでは、ご自身の選んだレンダリングエンジン/シェーダーの設定を使って、レンダリングしたシーンの見た目を頻繁に確認してください。見た目が 3D パネルで表示されたテクスチャとまったく違うかもしれません。

上述のようにマテリアル設定はふつう、オブジェクト表面の特性を決めます。Blender でマテリアルを設定する方法はいくつかあります。これらの方法は一般的には互換性を持ちません。シーンにあるオブジェクトごとに、どの方法を使うのか選ばねばなりません。

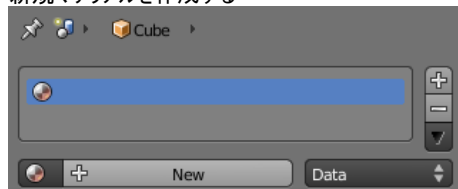
第一に、様々なマテリアルパネルにある [プロパティ](#) を設定できます。

次に [ノード](#) を使うことができます。グラフィカルなノードエディタを利用できます。

最後に、様々な特殊エフェクトを使ってオブジェクト表面の色を直接設定できます。厳密に言うとこれはマテリアルではありませんが、オブジェクトの見た目に影響するためここに含めます。[頂点ペイント](#)、[ワイヤーフレームレンダリング](#)、[ボリュームレンダリング](#)、[ハローレンダリング](#) があります。

マテリアル設定の効力には多くのシステム設定が影響します。その筆頭は使っているレンダリングエンジンです。Cycles と Blender Render Engine (別名 Blender Internal、BI) は似た結果を得るためにまったく違う照明調整が必要で、それでもオブジェクトの見た目がまったく変わることがあります。マテリアルプロパティの設定はテクスチャ方式(単独テクスチャ、マルチテクスチャ、GLSL)も影響します。したがっていつも、マテリアルデザインを始める前に適切なシステム設定を選ぶようにするとがよいでしょう。

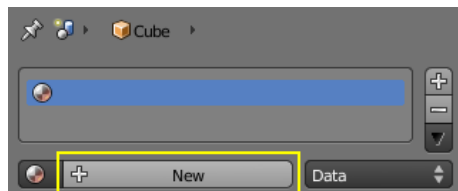
新規マテリアルを作成する



Material menu.

いかなる時でも、新しくオブジェクト作成されたときには、そこにマテリアルは全くリンクされていません。🔴をクリックすることで、シェーディングコンテキストに切り替わり、そしてマテリアルボタンウィンドウが表示されます。このウィンドウは、この時点ではほとんど空っぽになっているはずです。

Add *newmaterial*.に示すように、メニューボタンを使用して、新素材の追加を行いましょ。以下のオプションが利用できます:

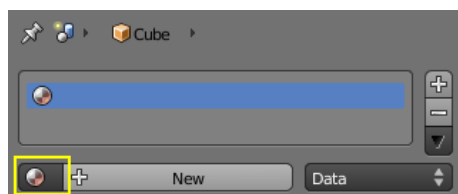


Add new material.

新素材を追加し、アクティブなオブジェクトまたはオブジェクトデータにリンクさせます。他のデータ・ブロックのように、Blenderが自動的にMaterial.001のような名前を設定してくれます。

マテリアルに明確な名前を付けると、把握しやすくなるので、とても良い考えです。特にそれらが複数のオブジェクトにリンクされている場合はね。

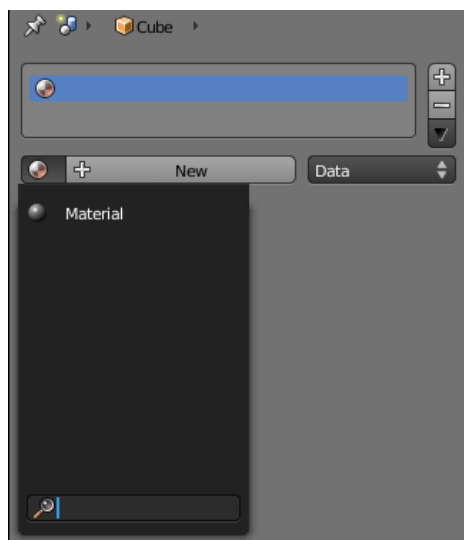
別のオブジェクトからマテリアルを共有する



Select an existent material.

Select an existing material

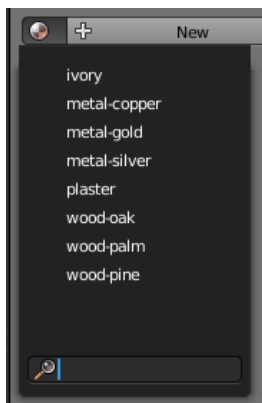
リストから既存のマテリアルを選び出します。



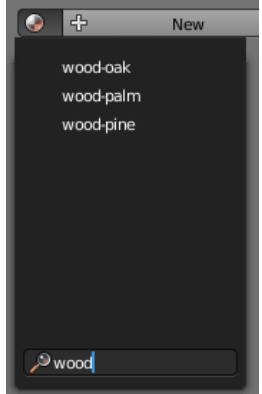
Existent material drop down menu

ブレンダーは、多くのオブジェクトの間で、マテリアルの設定も含む *あらゆるものを再利用できるように構築されています*。重複したマテリアルを作成する代わりに、簡単に既存のマテリアルを再使用することができます。2通りの方法があります:

- メッシュを選択した状態で、マテリアル名の左側にある球をクリックします。ポップアップリストには、現在のすべてのマテリアルが表示されます。いずれかを使用するには、単にそれをクリックします。



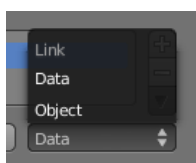
List of available materials



Filtered list

- 2.5の新要素は、マテリアルリストの下部にある検索フィールドです。例えば、"wood" と入力することにより、今在るすべてのマテリアル名がフィルタにかけられ、"wood" を含むマテリアルだけがリストに表示されます。
- 3Dビューで、CtrlL を使うと、選択しているすべてのオブジェクトを、[active object](#)のマテリアルに(および他の要素に) すばやくリンクさせることができます。多数のオブジェクトに同じマテリアルを設定する必要がある場合に非常に有用です。そのままそれらすべてを選択し、その後所望の材料を持っているオブジェクトも選択、そして CtrlL で、その "親"にリンクさせます。

オブジェクトまたはオブジェクトデータにマテリアルをリンクする



Link material to object or to object's data

マテリアルは、オブジェクトやオブジェクトデータにリンクさせることができます。

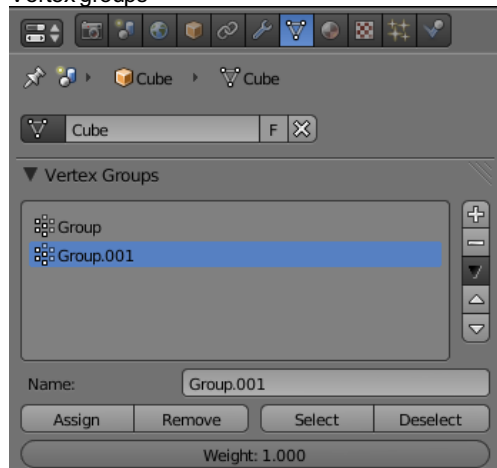
Please expand this section.

マテリアルのオプション

オブジェクトが少なくとも一つの材料を持ってリンクされると、たくさんの新しいパネルがすぐに表示され、マテリアルのシェーディングを精密に制御できるようになります。これらの各パネルの使用については、[次のセクション](#)で説明します。

[プレビューパネル](#)には、シェーダが生成する様々な種類の幾何学的な基本形状が表示されます。ボタンのウィンドウのレイアウトに応じて、パネルを折りたたむこともできます。

Vertex groups



Vertex groups menu in the object data context



Sort/copy vertex groups drop down menu

マテリアルは、頂点グループに割り当てるわけではなく、頂点に直接割り当てるので、複数のマテリアルを追加するときに頂点グループをメッシュに割り当てる必要はありません。ただし、頂点グループをつかると、シンプルな方法でマテリアルの割り当てを管理できます。

頂点グループの管理

新しい頂点グループは、+記号をクリックすることで簡単に作成でき、そして既存の頂点グループは頂点グループのリストの隣の-記号をクリックすることで簡単に削除できます。新しいグループはリストに追加されます。頂点グループの管理は簡単です:

1. 上矢印と下矢印を使用すると、選択している頂点グループをリストの上下に移動することができます。
2. ドロップダウンメニューをよびだすと、さらにいろいろなやり方でリスト内の頂点グループをソートしたりコピーすることができます。

デフォルトでBlenderが頂点グループにつける名前は、Groupや、Group.001のようなものです。各頂点グループは、名前フィールドで名前を変更できます。

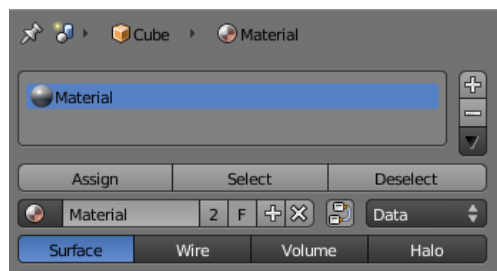
頂点グループに頂点を割り当てる

頂点は、自動的に頂点グループに追加されません。Assign ボタンと Remove ボタンを使用して、選択状態の頂点グループに選択状態の頂点を追加または削除しましょう。

頂点が頂点グループに割り当てられているとき、Blenderはこれらの頂点に Weight 値を割り当てます — 異なる頂点に異なる重みを割り当てることができ、多くの状況で役に立ちますが、マテリアルに関しては考慮されていません。

どんな時でも、頂点グループの選択または選択解除を、Select と Deselect のボタンを使用して行うことができます。

頂点にマテリアルを割り当てる



Material menu in edit mode

エディットモードの、マテリアルのコンテキストで、Assign ボタンを使用して、選択状態のマテリアルを選択状態の頂点に割り当てましょう。これは、前

もってマテリアルスロットを作成した後のマテリアルスロットでしか動作しないことに注意してください。

Select と Deselect のボタンを使用して、どの頂点に選択状態のマテリアルが割り当てられているか確認しましょう。

マテリアルのプレビュー

Mode: All Modes

Panel: Shading/Material Context → Preview

概要

プレビューパネルには、Shaders, Ramps, Mirror Transp properties および Texturesを含む、アクティブなマテリアルとそのプロパティが簡易的に可視化されます。新しくシェーダをデザインするときには非常に有用な多くの形状が用意されています: いくつかのシェーダ (Ramp の色をベースとするようなものやMinnaert)のような Diffuse shader) においては、シェーダ・イン・デザインの実現できるかを判断するには、かなり複雑な、または特定のプレビュー形状を必要とします。

オプション

Flat XY plane

壁、紙などのような平らなオブジェクトのテクスチャやマテリアルをプレビューするのに便利です。

Sphere

球のようなオブジェクトのテクスチャやマテリアルをプレビューするために有用だけでなく、金属や他の透明/反射材をデザインするとき、背景のチェッカー模様が役立ってくれます

Cube

キューブのようなオブジェクトのテクスチャやマテリアルをプレビューするときだけでなく、プロシージャルテクスチャをデザインするときにも有用です。チェッカー模様の背景を備えています。

Monkey

生き物や、あるいは複雑な非プリミティブ形状のテクスチャやマテリアルをプレビューするのに便利です。チェッカー模様の背景を備えています。

Hair strands

草、毛皮、羽毛、毛髪のような鎖のようなオブジェクトのテクスチャやマテリアルをプレビューするのに便利です。チェッカー模様の背景を備えています。

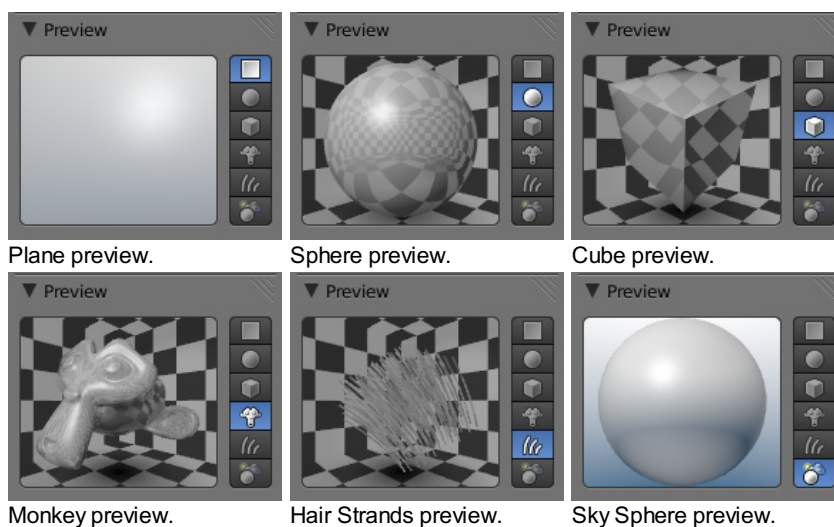
Large Sphere with Sky

球のようなオブジェクトのテクスチャやマテリアルをプレビューするときに有用だけでなく、金属や他の反射材をデザインするときにも、Sky の背景のグラデーションが役に立ってくれます。

Preview uses OSA (oversampling)

どのようなプレビューオプションであっても、品質をより良くするためにOSA (oversampling) を使用します。お使いのコンピュータがすでに遅い、または古い場合は、このオプションを無効にしましょう。

Examples



拡散シェーダ

Mode: All Modes

Panel: Shading/Material Context → Diffuse

拡散シェーダが決定するのは、シンプルに言えば、光がそこを照らすときのマテリアルの一般的な色です。ほとんどのシェーダは、現実を模倣するように設計されていて、強い照明の点から影の領域までスムーズに明暗を減衰させますが、Blenderは様々な特殊効果のためのさまざまなシェーダも持っています。

オプション

すべての拡散シェーダには、次のオプションがあります:

Color

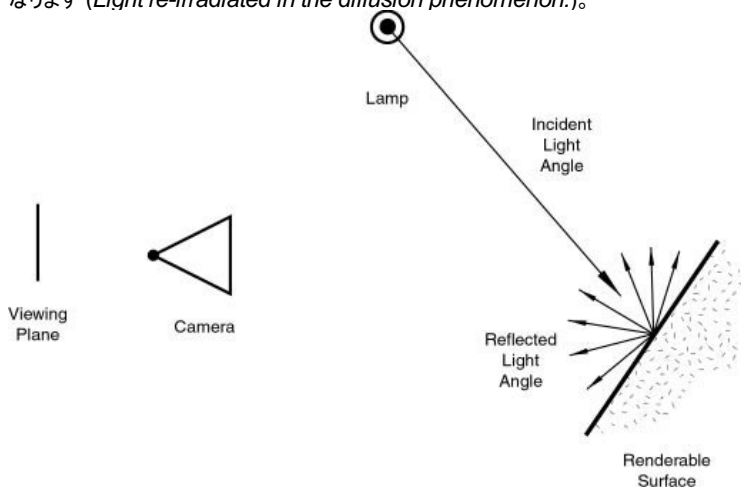
マテリアルのベースとなる **散乱光の色**

Intensity

シェーダの明るさ、またはより正確に言えば、実際にカメラに向かって拡散的に反射される入射光のエネルギーの量。

技術詳細

表面に当たって拡散現象で再照射される光は、散り散りになります。すなわち、すべての方向に等方的に再照射されます。これは、**視点からの入射角**にかかわらず、カメラはその表面上の点から同量の光を見ることを意味します。この特性によって、拡散光は **視点から独立したもの**になっています。もちろん、表面に当たる光の量は、入射光の角度に依存します。表面に当たる光の大部分が乱反射される場合は、表面がつや消しの外観になります (*Light re-irradiated in the diffusion phenomenon.*)。



Light re-irradiated in the diffusion phenomenon.

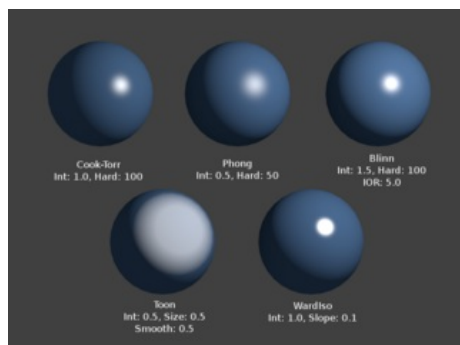
ヒント

いくつかのシェーダの名前は奇妙に聞こえるかもしれませんが - それらは伝統的に、それらが基づくモデルを最初に導入した人々にちなんで命名されています。

ランベルト

Mode: All Modes

Panel: Shading/Material Context → Shaders



Lambert Shader

これはBlenderのデフォルトの拡散シェーダであり、有能で普遍的でオールラウンドな働き者です。

[ヨハン・ハインリッヒ・ランベルト](#) (1728-1777)

はスイスの数学者、物理学者、かつ天文学者で、光の反射に関する著作を出版し、[ランベルト・ベールの法則](#)という光の吸収の法則を定式化したことで最も有名。

オプション

このシェーダが持つオプションは、デフォルトの、光の反射をどのくらい利用するか決定するものだけです。デフォルトでは0.8で、他のオブジェクトが明るくなるようにできます。

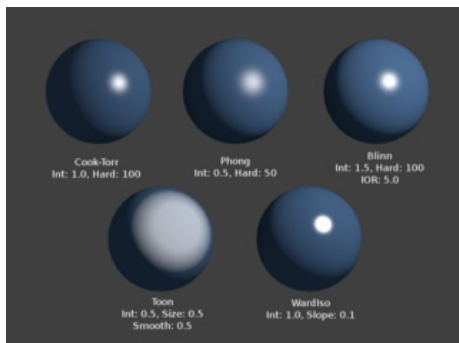


The Lambert diffuse shader settings.

Oren-Nayar

Mode: All Modes

Panel: Shading/Material Context → Shaders



Oren-Nayar Shader

オレン - ナイヤルは、散乱現象に多少 '物理的な' アプローチを採用していて、表面の微細な粗さの多寡を考慮に入れています。

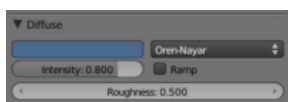
[Michael Oren](#) and [Shree K. Nayar](#)

Their [reflectance model](#), developed in the early 1990s, is a generalization of Lambert's law now widely used in computer graphics.

オプション

Rough

表面の粗さ、それゆえ、散乱の散らばりの量

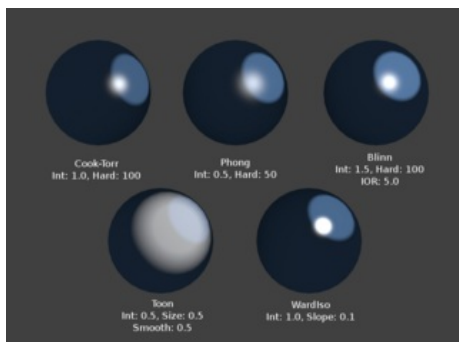


The Oren-Nayar diffuse shader settings.

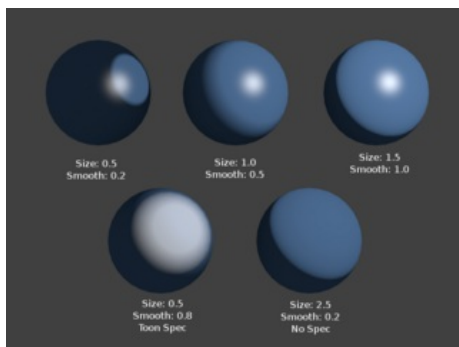
Toon

Mode: All Modes

Panel: Shading/Material Context → Shaders



Toon Shader, Different Spec



Toon Shader Variations

トゥーンシェーダは、非常に '非物理的な' シェーダで、その点でリアリティを装うことを意図したものではなく、漫画のセルスタイルのレンダリングを生成し、光と影の間に明確な境界をひき、明/暗の領域を均一にします。

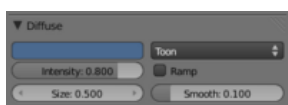
オプション

Size

明るい領域のサイズ

Smooth

明るい部分と暗い部分の境界のソフトさ

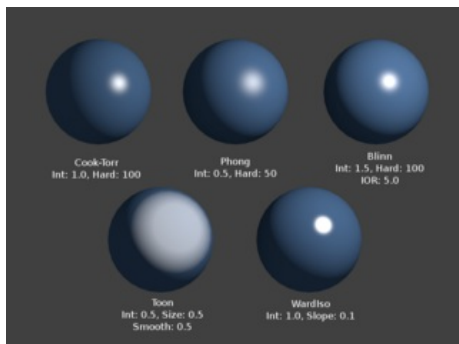


The Toon diffuse shader settings.

ミナルト

Mode: All Modes

Panel: Shading/Material Context → Shaders



Minnaert Shader

ミナルトは、標準的なランバートシェーダの部分で暗くする働きをします。なので *Dark* が 1 である場合、ランバートの結果とぴったり同じになります。darknessの値を高くすると、オブジェクトの中心のほうが暗くなります (見る側に向かって指した場合)。darkness値を低くすると、オブジェクトのエッジが明るくなり、ピロードのような見た目になります。

マルセル・ミナルト (1893-1970)

はベルギーの天文学者で、大気が光や画像に与える影響に興味を寄せた。1954年に *The Nature of Light and Color in the Open Air* と題する本を出版した。

オプション

Dark

'点灯' 領域の暗さ (higher)、または光源からそっぽを向くエッジの暗さ (lower)。

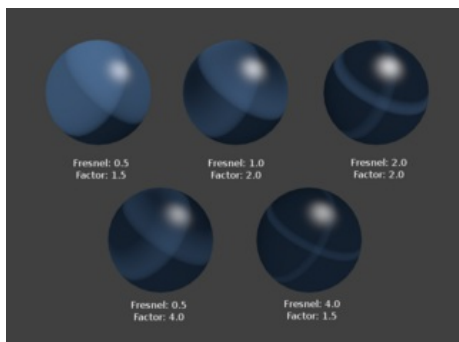


The Minnaert diffuse shader settings.

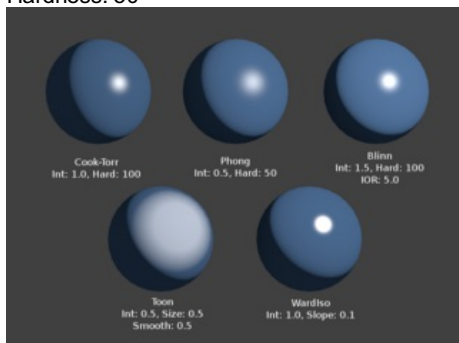
フレネル

Mode: All Modes

Panel: Shading/Material Context → Shaders



Various settings for the Fresnel shader, Cook-Torr Specular shader kept at Intensity 0.5, Hardness: 50



Fresnel Shader, Different Spec

フレネルシェーダでの拡散反射光の量は、光源の方向、すなわち、入射角に依存します。光源のほうを直接向いている領域は暗く、入射光に垂直な領域は明るく表現されます。

[オーギュスタン・ジャン・フレネル \(1788-1827\)](#)

はフランスの物理学者で、波動光学の理論の確立に大きく貢献した。

オプション

Fresnel

フレネル効果のパワー、5.0がmax。

Factor

フレネルの要素が溶け込むようにブレンドする係数、5.0がmax。



The Fresnel diffuse shader settings.

Specular Shaders

Mode: All Modes

Panel: Shading/Material Context → Specular

概要

Specular shaders は明るいハイライトを作成し、光沢のある表面に見えるようにして、光源の反射を模倣します。diffuse shadingとは異なり、鏡面反射は *視点に依存*します。スネルの法則によると、鏡面に当たる光は入射光の角度でミラー反射される (表面の法線から見て)ので、視る角度が非常に重要になります。

Note

重要なので強調しておきますが、ここでいう *鏡面反射* 現象は我々が鏡で見る反射ではなく、光沢のある表面に見える光のハイライトです。本当の鏡のような反射を得るためには、内部レイトレーサを使用する必要があります。マニュアルの [RENDERING](#) のセクションを参照してください。

オプション

各鏡面シェーダには共通の一般的なオプションがあります:

Specular color

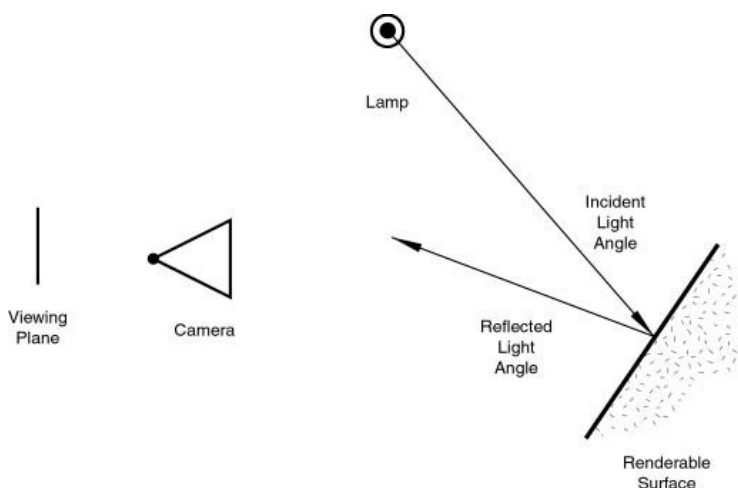
スペキュラハイライトの色

Intensity

明度、またはスペキュラハイライトの明るさ。この範囲は [0-1]です。

結果として、マテリアルは少なくとも2つの異なる色を持ちます。ディフューズのものとはスペキュラのもので、スペキュラカラーは、通常、純粋な白色に設定されていますが (反射光ソースと同じ "純粋なホワイト")、異なる値に設定して様々な効果にすることもできます (例えば、金属のような色のついたハイライト)。

技術詳細



Specular Reflection.

実をいうと、拡散反射と鏡面反射は、まったく同じ光の散乱プロセスによって生成されます。拡散反射する表面は、波長に関して、とても小さいデコボコのある表面で占められていて、光は表面の小片から、角度を少しずつ変化させながら多くの異なる方向に反射されます。

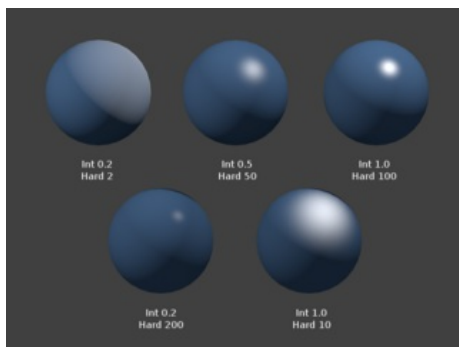
一方、鏡面反射のほうは、波長に関して、滑らかな表面で占められています。これは、表面の各点から散乱される光線のほとんどが同じ方向に向いていて、拡散的に散在するのではないことを示しています。これは細かいスケールでの話です。表面の粗さが入射光の波長よりもはるかに小さい場合、表面はフラットに表現され、鏡面としての役割を果たします。

粗さのスケールと光の波長との関係を理解することが難しいのであれば、ボールを想像してみてください (たとえば、センチメートルスケールのね):石積み の壁にそれを投げつけると (センチメートルのスケールの粗さで)、毎回別々の方向にはねかえり、すぐに見失ってしまうはず! 一方、コンクリートの壁に投げつけた場合は (粗さはたとえば、ミリスケール)、その跳ねかえりは非常に簡単に予測でき、追えるはず (多かれ少なかれ!) 光の反射もこれと同じような原理なのです...

CookTorr

Mode: All Modes

Panel: Shading/Material Context → Shaders



CookTorr Shader (Lambert 0.8)

概要

CookTorr (クック - トーランス) は基本的な鏡面反射シェーダで、光沢のあるプラスチック表面を作成するときに最も有用です。これはPhongの少し最適化されたバージョンです。

Robert L. Cook (LucasFilm) and Kenneth E. Torrance (Cornell University)

In their 1982 paper [A Reflectance Model for Computer Graphics](#) (PDF), they described "a new reflectance model for rendering computer synthesized images" and applied it to the simulation of metal and plastic.

オプション

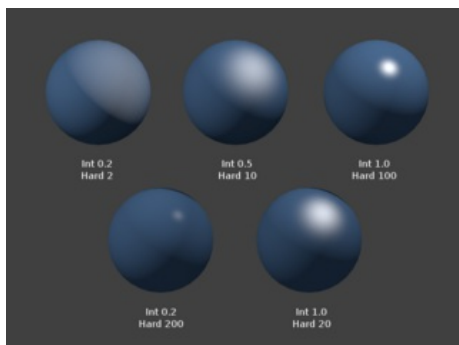
Hardness

スペキュラハイライトのサイズ

Phong

Mode: All Modes

Panel: Shading/Material Context → Shaders



Phong Shader (Lambert 0.8)

概要

Phongは基本的なシェーダで、CookTorrに非常に似ていますが、皮膚と有機表面に優れています

[Bui Tuong Phong](#) (1942-1975)

was a Vietnamese-born computer graphics pioneer that developed the first algorithm for simulating specular phenomenon. [His model](#) included components for not only specular lighting, but also diffuse and ambient lighting.

オプション

Hardness

スペキュラハイライトのサイズ

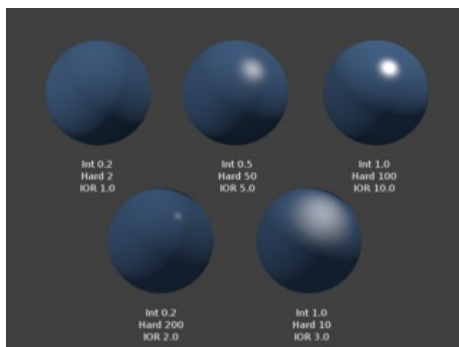
惑星の大気

そのぼやけから、このシェーダは惑星の周りの大気に適しています。惑星の周りに、惑星よりわずかに大きい球を追加します。そのマテリアルに、Phongスペキュラのシェーダを使用します。非常に低い α (0.05)、拡散性ゼロ、低いhardness(5)、ただし高い鏡面性(1)に設定しましょう。

Blinn

Mode: All Modes

Panel: Shading/Material Context → Shaders



Blinn Shader (Oren-Nayar Int 0.8, Rough 0.5)

概要

ブリンは、より'物理的'な鏡面反射シェーダで、多くの場合 Oren-Nayar 拡散シェーダと併用されます。4つのオプションが追加されるので、よりコントロールラブルになります。前述の3つに、*屈折率* (IOR)です。

ジェームズF.ブリン

はNASAのジェット推進研究所に勤務し、彼の功績はカール・セーガンのテレビドキュメンタリー *Cosmos*において広く知られるようになった。そのモデルは彼の1977年の論文 [Models of Light Reflection for Computer Synthesized Pictures](#) (PDF) に記され、光の方向にともなう鏡面反射強度の変化や、表面上のハイライトのより正確な位置決めなども盛り込まれている。

オプション

Hardness

スペキュラハイライトのサイズ。ブリンシェーダでは、ハイフォンやCookTorrよりはるかにタイトなスペキュラハイライトが可能です。

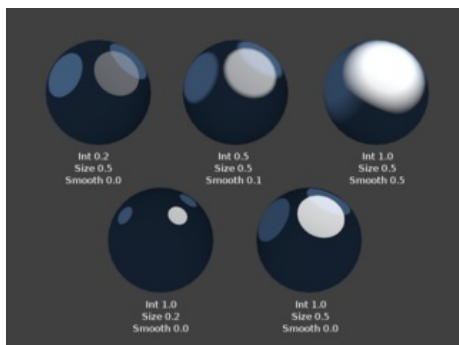
IOR

'屈折率'。このパラメータは、実際に材料を通りぬける光線の屈折を計算するためのものではありませんが(そのためにはレイトレーサーが必要になる)、スネルの法則によって鏡面反射の輝度と広がりを正しく計算します。

Toon

Mode: All Modes

Panel: Shading/Material Context → Shaders



Toon Specular Shader (Toon Diffuse, Int 0.8, Size & Smooth match)

概要

トゥーン鏡面シェーダは、トゥーン拡散シェーダの対偶のものです。漫画のセル画のような、シャープで均一なハイライトを生成するように設計されています。

オプション

Size

スペキュラハイライトのサイズ

Smooth

ハイライトのエッジの柔らかさ

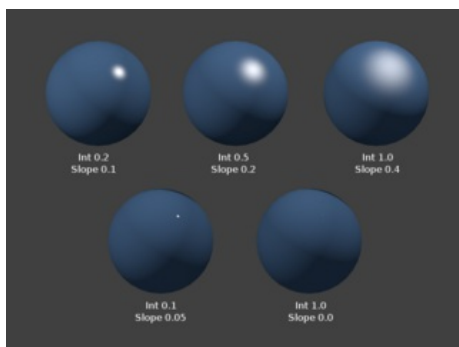
ヒント

トゥーンシェーダはまた、ColorRampsを使用することで、よりコントロールラブルな方法で遂行することもできます。

WardIso

Mode: All Modes

Panel: Shading/Material Context → Shaders



WardIso Shader

概要

WardIso は、金属やプラスチックに役立つ柔軟な鏡面反射シェーダです。

Gregory J. Ward

は物理学の最も基本的な法則に従った比較的単純なモデルを開発しました。彼の1992年の論文 *Measuring and modeling anisotropic reflection* でWardが双方向反射率分布関数 (BRDF) を紹介して以来、パラメータが少なく制御が簡単なことから、広くコンピュータグラフィックスで使用されるようになりました。彼のモデルは、等方性表面 (光の方向に依存しない) と異方性表面 (方向に依存する) の両方を表現することができます。Blenderでは、ワード鏡面シェーダはまだ **Ward Isotropic** と呼ばれていますが、実際には異方性です。
([PDF](#))

オプション

Slope

面勾配の標準偏差。以前は [二乗平均平方根](#) またはrms値と呼ばれていたもので、このパラメータの影響内では、スペキュラハイライトのサイズの制御を、他のスペキュラシェーダとは別の方法を使用して行ないます。

Ramps

Mode: All Modes

Panel: Context Shading → sub-context Material → Ramps

実生活のさまざまな状況 — 肌や金属のような — では、拡散反射と鏡面反射の色が若干変わることがあります。これは表面が受け取るエネルギー量や入射光の角度がベースとなるものです。Blenderの Ramp Shader オプションでは、Materialの色の範囲を設定したり、その範囲が表面上でどのように変化するか、そしてどのようにブレンドされて '実際の色' になるかを定義できます (典型的にはマテリアルから、またはテクスチャとして出力される)。

概要


色勾配では、明るい色から暗い色へ、最も強く発光する領域から暗く発光する領域へ、というような単純なブレンドだけでなく、マテリアル全体の色のグラデーションを精密に制御することができます。発光領域から影の領域までのグラデーションの制御につかう各オプションと同様に、色勾配には '法線' の入力が備わっていて、カメラのほうを向く面からカメラにそっぽを向く面にかけてのグラデーションを定義できます。これは多くの場合、見る角度に基づいてその色が変わるような、車のメタリック塗料のようなタイプのマテリアルに使用されます。

Blenderのテクスチャ計算はシェーディングの前に行われるので、Ramp Shader はテクスチャやマテリアルの色を完全に置き換えることもできます。とはいえ、ミキシングオプションとアルファ値を使用することで、Blenderのマテリアルに、追加のシェーディングレイヤーを作成することは可能です。

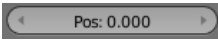
オプション



Ramps Panel

Blender 2.5 では、拡散 シェーダと 鏡面 シェーダに独立したRampパネルがあり、 ボタンを使用してそれぞれのオンとオフを切り替えることができます。

デフォルトでは、色勾配パネルは、2つの色を両端に広がっています。ひとつめ (stop 0) は黒と透明 (Alpha=0) で、ふたつめ (stop 1) は白と不透明 (Alpha=1) です。

カラーストップマーカの位置の変更は、(1) カラーバンドでストップマーカをドラッグする、(2)  ボックスで Pos 値を変更する、のいずれかの方法で行えます。

各マーカの色とアルファ値は、 ボックスをクリックして設定できます。



Input popup menu

Input

入力メニューには、グラデーションの定義につかう以下のオプションが含まれています:

Shader

色を定義するマテリアルシェーダ (Lambert, CookTorr) によって渡される値。ここで光の量は、色には影響せず、光の方向にだけ影響します。

Energy

Shaderと同様に、ここでもまた光源のエネルギー、色、距離が考慮されます。照らす光が多くなるときに、これがマテリアルの色を変化させます。

Normal

カメラに対して相対的な、表面の法線で、Ramp Shaderに使用されます。これはテクスチャでも同様のことが可能ですが、利便性のために追加されました。

Result

前の3つのオプションはすべて、ランプごとに作用しますが、このオプションはすべてのシェーディング計算の最後で動作します。これはシェーディングを全体を完全に制御でき、全体を 'Toon' スタイルで出力させるといったこともできます。ここにあるアルファ値を使用すると、マテリアルへ最後の仕上げとして微調整をするときに非常に便利です。



Blend popup menu

Blend

さまざまなブレンドモードのリストになっていて、Inputからの色を色勾配シェーダとブレンドするときに使用できる。

Factor

このスライダには、Inputからの色に対する色勾配シェーダの要素全体が表示されます。

Colorbands


Mode: All Modes

Panel: Context Shading → sub-context Material → Ramps

カラーバンドには多くの色の連続を通した勾配を内包させることもでき(アルファ付き)、各色のスペクトルの特定の位置を越えて作用させることができます。Colorbandsは、マテリアルとテクスチャの両方で使用され、色の範囲を計算したり表示したりするさまざまな場所でも同様に使用されます。

オプション

Add

新しいマークをカラーバンドの中心にデフォルトの色(ニュートラルグレー)で追加します。新しいマークは、カラーバンド自体で Ctrl LMB  クリックでも追加することができます。これでクリックした位置にマークが追加され、マウスポインタの下にあるものと同じ色になります。

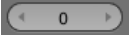

Delete

現在選択されているマークをカラーバンドから削除します。


F

カラーバンドを反転します。

0


アクティブなマークの数。このマークの値は表示されているものの値で、カラーバンドには、アクティブなマークが破線として表示されます。別のマークを選択することもでき、(1)  のスライダー矢印を使用する、(2) 表示されている番号をクリックして色マークの番号を入力する、または (3) カラーバンドのマーカを LMB  クリックする。

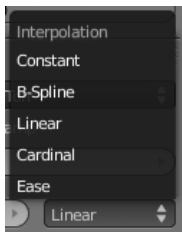
Pos

カラーバンドのアクティブなカラーマークの位置 (0.0–1.0 の範囲)。カラーマークの位置は、カラーバンドで LMB  ドラッグでも変更できます。

Reordering colors

カラーマークの位置が並び変えられた場合に、自動的に番号が付けなおされ、常に左から 0 で始まり、右にインクリメントするようになります。

Positionスライダの右にあるColorswatchには、アクティブなマークの色が表示されます。LMB  クリックするとカラーピッカーが表示され、そこでカラー(RGB)と透明度(Alpha)の値を設定できます。



Interpolation
popup menu

Linear

マーカーの値の間の補間の様々なモードはInterpolationメニューで選択できます:

Ease

二次方程式によってゆるやかにします。

Cardinal

基数。

Linear

線形(デフォルト)。色間をスムーズに、一貫して遷移させます。

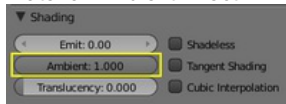
B-Spline

B スプライン。

Constant

一定。

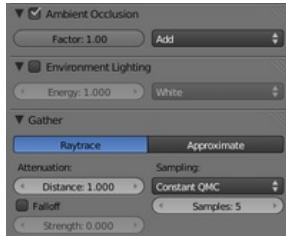
Material Ambient Effect



The Ambient slider.

各オブジェクトのマテリアル設定 (拡散シェーダ) には Ambient スライダーがあり、オブジェクトが受けとる環境光の量を選択することができます。

Blender 2.5において、これはデフォルトで1.0に設定されています。オブジェクトに受け取らせたい環境光の量に応じて、このスライダーを設定するとよいでしょう。洞窟の奥深くのものには環境光は届かず、それに対して入り口に近いものには多くの光が届きます。注目点としては、この効果はアニメーション化することもでき、オブジェクトが影から現れて光のもとに出てくるように変化させることもできます。



World settings for Ambient Occlusion and Environment Lighting.

アンビエントオクルージョンと環境照明の設定は World メニューにあり、これらの照明コンポーネント両方に影響を与えるパラメータも一緒に World Gather メニューにあります。

レイトレースによる鏡面反射

Mode: All Modes

Panel: Shading/Material Context → Mirror Transp

概要

レイトレーシングは、マテリアルに周りのものを反射させて、鏡のようにする使い方もできます。レイトレース反射の原理は非常に簡単です。光線はカメラから発射され、それがオブジェクトにぶつかるまでずっと、シーンを通過します。光線が最初にヒットするオブジェクトに反射性がない場合は、光線がオブジェクトの色になります。オブジェクトが反射するものであれば、光線は現在の場所からバウンドして、別のオブジェクトまで移動する、というように、最終的に非反射オブジェクトに出会うまで、光線の色のチェーン全体をつなぎ続けます。

最終的に、初めの反射オブジェクトは、その Reflectivity の値に比例して、その環境の色を継承します。当然、シーン内に反射物しかない場合は、レンダリングが永遠に続く可能性があります。この理由から、単一の光線の進行には、制限メカニズムが備わっていて、これはDepthの値によって設定されます。このパラメータで、単一の光線に許可する跳ね返りの最大数を設定します。

Note

レイトレースによる反射を使用したい場合は、シーンの設定でレイトレーシングを有効にする必要があります。これは、Scene/Render context → Render Panelで行います。レイトレーシングはBlender 2.37以降ではデフォルトで有効になっています。

ミラーパネルの *Color Swatch* は、反射されて返ってくる光の色です。一般的に、通常ミラーには、白を使用します。しかし、反射に色がつくミラー(例えば金属)では、色見本をクリックして色を変更することができます。ミラー化による反射の量は、Reflectivityの値によって決まります。0より大きい値に設定されている場合、ミラーの反射性が有効化され、反射は色見本で設定した色に染められます。

オプション



The Mirror Panel

Enable raytraced reflections

レイトレースによる反射を有効または無効にする

Reflectivity

オブジェクトの反射率の量を設定します。完全なミラーを必要とする場合は1.0の値を使用し、また、反射がいらぬ場合は0.0に設定します。



Picking a mirror color

Color swatch

鏡面反射の色

デフォルトでは、クロムめっきのようなほぼ完全な反射性マテリアルや、またはミラーオブジェクトは、その周辺の色を正確に反映します。しかし、他のいくつかの同様の反射材は、自身の色をつけて反射します。これは、たとえば、よく磨かれた銅や金の場合です。ブレンダー内でこれを再現するためには、ミラーの色を適宜設定する必要があります。ミラーの色を設定するには、シンプルにミラーパネルの色見本をクリックして、色を選択します。

Fresnel

フレネル効果のパワーを設定します。フレネル効果は、表面の法線と視線方向のなす角度に応じて、マテリアルがどのように反射するかを制御します。一般的に、この角度が大きいほど、反射性の高いマテリアルになります(これは一般的にオブジェクトの輪郭に発生します)。

Blend

反射エリアと非反射エリアとの間で、ブレンドがどのように起こるかを調整するための制御因子。

Depth

光の相互反射の最大許容数。シーンにたくさんの反射物があつたり、そのような反射物にカメラがズームしている場合、周囲の反射を反射物に映して見えるようにしたいのであれば、この値を増やす必要があります(!)。このケースでは、Depthは4か5の値にするのが一般的に良いでしょう。

Max Dist

反射光線がカメラから遠ざかることができる、Blender単位での最大距離 (Z-Depth)。この範囲より遠くの反射は、計算時間を減らすためにフェードアウトされます。

Fade to

Max Distance 内では交点を持たない光線の色。Material color は屋内シーン、Sky color (World settings) は屋外シーンに最適です。



Suzanne in the Fun House
([blend](#))

Gloss

塗装において、高光沢仕上げは非常に滑らかで光沢があります。フラット、または低光沢では、光を分散し、非常にぼやけた反射が得られません。また、凹凸があつたり磨かれていてもザラザラな表面 (例えば車の塗装など) は完全ではないので、少し Gloss < 1.0 にする必要があります。右の例では、左のミラーの Gloss が 0.98 で、真ん中は Gloss = 1.0、右は Gloss が 0.90 です。反射をリアルにするために、この設定を使用して、ミラーを徹底的に曇らせましょう。また、この値を使用すると、ミラー内の被写界深度を模倣することができます。

Amount

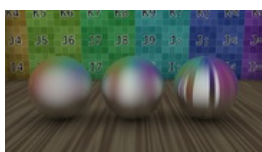
反射の輝き。Values < 1.0 では、拡散性の、ぼやけた反射を与え、そして以下の設定が有効になります。

Threshold

adaptive sampling のしきい値。サンプリングの作用が、この量未満になった場合 (パーセンテージ)、サンプリングが停止されます。しきい値を上げると、adaptive sampler が頻繁にスキップするようになりますが、反射のノイズが多くなる可能性があります。

Samples

ぼやけた反射を平均した cone samples の数。サンプルをより多くすると、結果が滑らかになりますが、レンダリング時間も長くなります。



Anisotropic tangent reflecting spheres with anisotropic set to 0.0, 0.75, 1.0. ([blend](#))

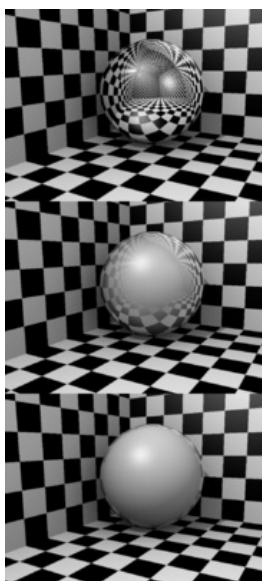
Anisotropic

反射の形状、0.0 (circular) から 1.0 (接線に沿って完全に引き伸ばす)。Tangent Shading がオンになっている場合、Blender がぼやけた反射を自動的に異方性反射としてレンダリングします。

Tangent をオンに切り替えると、Anisotropic スライダーで、この異方性反射の強さを制御できるようになります。1.0 (デフォルト) では完全に異方性で、0.0 では完全に円形の範囲になり、マテリアル上の接線シェーディングがオフになっている場合と同様になります。異方性レイトレス反射は、接線シェーディングと同じ接線ベクトルを使用しています。なので、角度とレイアウトの修正を同様の、自動生成による接線を使う方法、またはメッシュのUV座標に基づく方法で行えます。

例

フレネル



Demonstration of Fresnel effect with values equal to (from top to bottom) 0.0, 2.5 and 5.0

フレネルが本当はどんなものなのかを理解するために、ちょっとした実験を実施してみましょう。雨の日の後、外に出て水たまりを上の方から見てみましょう。水たまりを通して地面を見ることができます。水たまりのすぐ前でひざまずいて、顔を地面に近づけ、ある距離の点で水たまりをもう一度見てみると、近い水面では地面を見ることができますが、水たまりのもう一方の端に目を移していくと、空の反射しか見えなくなり、地面が徐々に見えなくなっ

ていきます。これがフレネル効果です:視野角や表面の法線に応じて、表面で反射と非反射の特性を併存させることができます。

*Demonstration of Fresnel effect with values equal to (from top to bottom) 0.0, 2.5 and 5.0*では、この振る舞いが、完全な反射性をもつマテリアルで実演されています (ミラー反射率 1.0)。

フレネル 0.0 は、完全なミラーマテリアルを表し、一方フレネル 5.0 は光沢のある素材を表します。下の画像ではほとんど目立たせていませんが、エッジの周りではマテリアルが完全に反射性になっています。

フレネルの境界線のスムーズさは、Blend スライダを使用してさらに制御することができます。

Transparency

Mode: All Modes

Panel: Shading/Material Context → Transparency

Blenderのマテリアルは、透明に設定して、光を透過させることもできます。マテリアルに設定できる透明には以下の3つの種類があります:



Z-Transparency Render



Raytraced render, with IOR=1.2

Mask

単純に背景をささぎる。パイプラインの下にさらに組み合わせるときに便利— [Mask Transparency](#)参照。

Z Transparency

これは透明な面にアルファマップを使用します。このオプションでは、基本的な設定しか使用できず、屈折を計算することはできません。

Raytrace

屈折の計算にレイトレーシングを使用します。レイトレーシングを行えば、複雑な屈折、減衰、およびぼかしが可能になります。

(右にある画像の生成につかった.blendを見るには、こちらをクリックしてください: [File:Manual-Materials-Properties-RaytracedTransparency Comparison.blend](#)).

レイトレースによる透明

レイトレーシングを使用して、レンズのように透明な材質を通り抜けるときの光線の屈折をシミュレートします。光線が、カメラから送り出され、それがオブジェクトに到達するまでシーンを通過します。光線が最初にヒットするオブジェクトに反射性がない場合は、光線はオブジェクトの色になります。

オブジェクトが透明である場合は、それを通り抜けて次のオブジェクトへ、というように、非透過的なオブジェクトにぶつかって最終的にチェーン全体の光線の色が決まるまで、光線の道のりは続きます。やがて、最初の透明オブジェクトが、自身の Alpha 値 (とヒットするまでの間の各透明なマテリアルの Alpha 値)に比例して、その背景の色を継承します。

ただし光線が透明なオブジェクトを通過すると同時に、材料の屈折率 (IOR) によってコースからそれることがあります。あなたが実際にガラスのまっさらな球をのぞいてみると、背景が逆さまになってゆがんでいることに気づくはずで: これはすべて、ガラスの屈折率のせいです。

レイトレーシングを有効にする

レイトレースによる透明を得るには、以下を行う必要があります:

1. レンダーの設定 (Render settings) でレイトレーシングを有効にしてください。これは、Render context → Shading Panelで行います。レイトレーシングは、デフォルトで有効になっています。
2. アルファ値を1.0ではない何か他の値に設定します。
3. バックグラウンドの素材が透明なオブジェクトを通過する光を受け取るようにするには、Material → Shadow panelでReceive Transparentをオンにする必要があります。

オプション



The Transparency Panel.

Alpha

材料の透明度の量を設定します。

Specular

スペキュラカラーのアルファ減衰を制御します。

Fresnel

フレネル効果のパワーを設定します。フレネル効果は、表面の法線と視線方向のなす角度に応じて、材料がどのように透明になるかを制御します。一般的に、角度が大きくなるほど、材料がより不透明になります (これは一般的にオブジェクトの輪郭で発生します)。

Blend

透明領域と不透明領域の間のブレンドを制御します。フレネルが0より大きい場合にもみ使用します。

IOR

屈折率。材料を通過する光線がどのくらい屈折されるかを設定するもので、したがって背景の画像が歪められて描き出されます。下の [IOR values for Common Materials](#) を参照してください。

Filter

透過光線トレースのフィルタリングの量。この値が高いほど、マテリアルのベースカラーがより多く表示されます。マテリアルはまだ透明なままですが、マテリアルの色を帯び始めます。デフォルトでは無効 (0.0) になっています。

Falloff

物質を通過する際に、どのくらい早く光が吸収されるか。ガラスに '深さ' と '厚さ' を与えます。

Limit

マテリアルがこの値より厚くなると、透明ではなくなります。これは、カラーフィルタが作用を開始した後に、しきい値の制御に使用します。

Depth

単一の光線が通過できる透過表面の最大数を設定します。標準的な値というのはありません。Depthが設定されている透明オブジェクトを通して見る場合、Depthの範囲外の透明オブジェクトは真っ黒にレンダリングされます。言い換えれば、透明オブジェクトの表面に黒い部分にあること気づいたら、解決策はおそらく、その Depth 値を増加させることです (これは透明なオブジェクトのレイトレーシングで共通の問題です)。また、背後のオブジェクトの透明な影をオンにする必要があります。

Gloss

マテリアルの光沢度の設定。

Amount

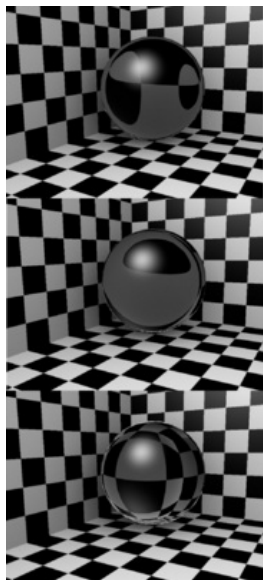
屈折の明瞭さ。ゼロよりも低く設定すると、ぼやけた屈折を得られます。

Threshold

adaptive samplingのしきい値。サンプリングの働きがこの量未満になった場合 (パーセンテージ)、サンプリングが停止されます。

Samples

ぼやけた反射における平均した cone samples の数。

例**屈折率**

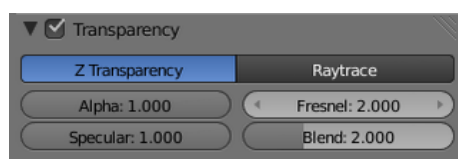
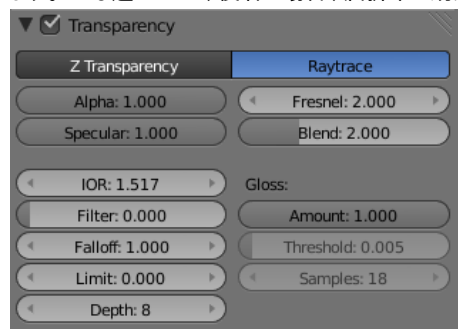
Influence of the IOR of an Object on the distortion of the background: spheres of Water, Glass and Diamond (top to bottom).

(*Influence of the IOR of an Object on the distortion of the background: spheres of Water, Glass and Diamond (top to bottom).*). 材料ごとに特有のそれぞれ異なる値があります: 空気は **1.000** (無屈折)で、アルコールは **1.329**、ガラスは**1.517**、プラスチックは**1.460**、水は**1.333**、ダイヤモンドは **2.417**です。

フレネル



16のガラスを様々な方向に回転させて、角度に依存したレイトレースによるフレネル効果 (左)と、アルファバツファによる透明度 (右)を実際に試しています。主な違いとして、後者の場合、屈折率の効果が無いことに注意してください。(Download [.blend](#).)

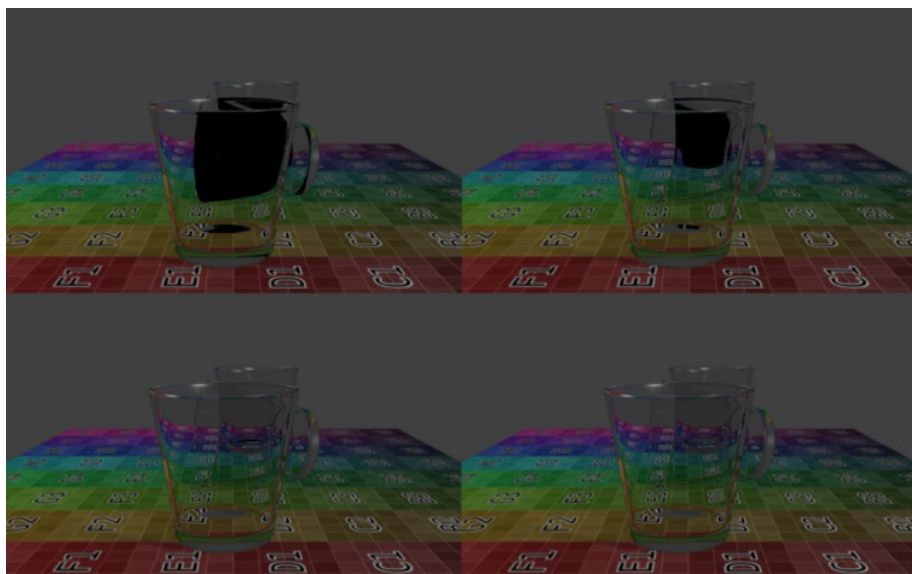


Settings for Fresnel using raytrace (left) and Z transparency (right).

F4のガラススタイルのスペキュラハイライトに注目してください(ライトとカメラの中間に向いている)。フレネル効果は、面がカメラからそっぽを向いている行Cと6列目で見るすることができます。

フレネル効果の量は、Blend 値の増加、または Alpha値の減少のいずれかによって制御できます。

Depth



A simple scene with three glasses on a surface and three lamps. Depth was set to 4, 8, 12, and 14, resulting in render times of 24 sec, 34 sec, 6 min, and 11 min respectively. (Download [.blend](#).)

Depthを増加させると、レンダリング時間も大幅に増加します。光線が面を通過するたびに、レイトレーシングアルゴリズムが再帰的に呼び出されます。上記の例では、各ガラスのそれぞれの側は、外面と内面を持っています。光線は、このように各ガラスの4面を通過する必要があります。

しかしそれだけでなく、表面上の各点で光の一部が反射されたり、または様々な方向の表面が鏡面反射されることがあります。複数の光線でのこの結果は、各点について計算する必要があります(多くの場合、[tree of rays\[1\]](#)と呼ばれる)。上記の各レンダリング画像には、 $640 \times 400 = 256\,000$ のピクセルがあります。Depthを増加させることにより、光線の少なくとも一つのツリーが、各ピクセルに追加されます。

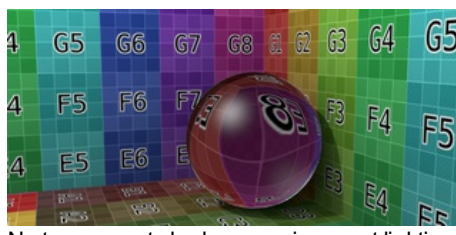
あなたのコンピュータに気遣いを。シーン内のオブジェクトを慎重に配置して、透明オブジェクトが重ならないようにすることも、興味深い代替手段となります。

ヒント

透過物の影



No transparent shadows



No transparent shadows, environment lighting enabled



Transparent shadows enabled, alpha set to 0.0



As previous, alpha set to 0.25



Transparent shadows with ambient occlusion set to multiply, distance 1 (radius of sphere)



As previous, distance increased to 2 (diameter of sphere)

デフォルトでは、透明オブジェクトの影は、オブジェクトがまったく透明ではないかのように、黒一色でレンダリングされます。しかし、現実には、オブジェクトがさらに透明になれば、その影は明るくなります。

Blenderでは、透過物の影は、透明オブジェクトから影を受けるマテリアルとして設定されています。これは、Material context → Shadow panel でReceive Transparent ボタンを使って有効にしたり、無効にしたりできます。影の明るさは影のかかるマテリアルのAlpha値に依存します。

透過物のレイトレースシャドウの代替としては、World コンテキストにある、Ambient Occlusion、Environment Lighting、Gather panelsがあります。また、テクスチャを使用して、影を受けるマテリアルの Intensity の値を制御することもできます。

一般的な材料のIOR値

次のリストには、レイトレースによる透明が、様々な液体、固体(宝石)、ガスに使用されるとき、屈折値の指標のいくつかを記載しています:

A	E	J	S
Acetone 1.36	Ebonite 1.66	Jade, Jadeite 1.64 - 1.667	Sanidine 1.522
Actinolite 1.618	Ekanite 1.600	Jade, Nephrite 1.600 - 1.641	Sapphire 1.757 - 1.779
Agalmatolite 1.550	Elaeolite 1.532	Jadeite 1.665	Sapphire, Star 1.760 - 1.773
Agate 1.544	Emerald 1.560 - 1.605	Jasper 1.540	Scapolite 1.540
Agate 1.540	Emerald Catseye 1.560 - 1.605	Jet 1.660	Scapolite, Yellow 1.555
Air 1.000	Emerald Synth flux 1.561	K	Scheelite 1.920
Alcohol 1.329	Emerald, Synth hydro 1.568	Komerupine 1.665	Selenium, Amorphous 2.92
Alcohol, Ethyl (grain) 1.36	Enstatite 1.663	Kunzite 1.660 - 1.676	Serpentine 1.560
Alexandrite 1.745	Epidote 1.733	Kyanite 1.715	Shampoo 1.362
Alexandrite 1.750	Ethanol 1.36	L	Shell 1.530
Almandine 1.83	Ethyl Alcohol 1.36	Labradorite 1.560 - 1.572	Silicon 4.24
Aluminum 1.44	Euclase 1.652	Lapis Gem 1.500	Sillimanite 1.658
Amber 1.545	F	Lapis Lazuli 1.50 - 1.55	Silver 0.18
Amblygonite 1.611	Fabulite 2.409	Lazulite 1.615	Sinhalite 1.699
Amethyst 1.540	Feldspar, Adventurine 1.532	Lead 2.01	Smaragdite 1.608
Ammolite 1.600	Feldspar, Albite 1.525	Leucite 1.509	Smithsonite 1.621
Anatase 2.490	Feldspar, Amazonite 1.525	M	Sodalite 1.483
Andalusite 1.640	Feldspar, Apatite 1.565	Magnesite 1.515	Sodium Chloride 1.544
Anhydrite 1.571	Labradorite 1.565	Malachite 1.655	Spessartite 1.79 - 1.81
Apatite 1.632	Feldspar, Microcline 1.525	Meerschaum 1.530	Sphalerite 2.368
Apophyllite 1.536	Feldspar, Oligoclase 1.539		Sphene 1.885
Aquamarine 1.575			
Aragonite 1.530			

Argon	1.000281	Flourite	1.434	Mercury (liq)	1.62	Spinel	1.712 - 1.717
Asphalt	1.635	Formica	1.47	Methanol	1.329		
Axinite	1.674 - 1.704	G		Milk	1.35	Spinel, Blue	1.712 - 1.747
Axinite	1.675	Garnet, Andradite	1.88 - 1.94	Moldavite	1.500		1.708 - 1.735
Azurite	1.730	Garnet, Demantoid	1.880 - 1.9	Moonstone	1.518 - 1.526	Spinel, Red	
B		Garnet, Demantoid	1.880	Moonstone, Adularia	1.525	Spodumene	1.650
Barite	1.636	Garnet, Grossular	1.738	Moonstone, Albita	1.535	Star Ruby	1.76 - 1.773
Baryocalcite	1.684	Garnet, Hessonite	1.745	Moonstone, Albite	1.585 - 1.594	Staurolite	1.739
Beer	1.345	Garnet, Mandarin	1.790 - 1.8	Morganite	1.594	Steatite	1.539
Benitoite	1.757	Garnet, Pyrope	1.73 - 1.76	N		Steel	2.50
Benzene	1.501	Garnet, Rhodolite	1.740 - 1.770	Natrolite	1.480	Stichtite	1.520
Beryl	1.57 - 1.60	Garnet, Rhodolite	1.760	Nephrite	1.600	Strontium Titanate	2.410
Beryl, Red	1.570 - 1.598	Garnet, Spessartite	1.810	Nitrogen (gas)	1.000297	Styrofoam	1.595
Beryllonite	1.553	Garnet, Tsavorite	1.739 - 1.744	Nitrogen (liq)	1.2053	Sugar Solution 30%	1.38
Brazilianite	1.603	Garnet, Uvarovite	1.74 - 1.87	Nylon	1.53	Sugar Solution 80%	1.49
Bromine (liq)	1.661	Gaylussite	1.517	O		Sulphur	1.960
Bronze	1.18	Glass	1.51714	Obsidian	1.489	Synthetic Spinel	1.730
Brownite	1.567	Glass, Albite	1.4890	Oil of Wintergreen	1.536	T	
C		Glass, Crown	1.520	Oil, Clove	1.535	Taaffeite	1.720
Calcite	1.486	Glass, Crown, Zinc	1.517	Oil, Lemon	1.481	Tantalite	2.240
Calspar	1.486	Glass, Flint, Dense	1.66	Oil, Neroli	1.482	Tanzanite	1.690-1.7
Cancrinite	1.491	Glass, Flint, Heaviest	1.89	Oil, Orange	1.473	Teflon	1.35
Carbon Dioxide (gas)	1.000449	Glass, Flint, Heavy	1.65548	Oil, Safflower	1.466	Thomsonite	1.530
Carbon Disulfide	1.628	Glass, Flint, Light	1.80	Oil, vegetable (50° C)	1.47	Tiger eye	1.544
Carbon Tetrachloride	1.460	Glass, Flint, Medium	1.62725	Olivine	1.670	Topaz	1.607 - 1.627
Carbonated Beverages	1.34 - 1.356	Glycerine	1.473	Onyx	1.486	Topaz, Blue	1.610
Cassiterite	1.997	Gold	0.47	Opal, Black	1.440 - 1.460	Topaz, Imperial	1.605 - 1.640
Celestite	1.622	H		Opal, Fire	1.430 - 1.460	Topaz, Pink	1.620
Cerussite	1.804	Hambergite	1.559	Opal, White	1.440 - 1.460	Topaz, White	1.630
Ceylonite	1.770	Hauyne	1.490 - 1.505	Oregon Sunstone	1.560 - 1.572	Topaz, Yellow	1.620
Chalcedony	1.544 - 1.553	Hauynite	1.502	Oxygen (gas)	1.000276	Tourmaline	1.603 - 1.655
Chalk	1.510	Helium	1.000036	Oxygen (liq)	1.221	Tourmaline	1.624
Chalybite	1.630	Hematite	2.940	P		Tourmaline, Blue	1.61 - 1.64
Chlorine (gas)	1.000768	Hemimorphite	1.614	Padparadja	1.760 - 1.773	Tourmaline, Catseye	1.61 - 1.64
Chlorine (liq)	1.385	Hiddenite	1.655	Painite	1.787	Tourmaline, Green	1.61 - 1.64
Chrome Green	2.4	Honey, 13% water content	1.504	Pearl	1.530	Tourmaline, Paraiba	1.61 - 1.65
Chrome Red	2.42	Honey, 17% water content	1.494	Pericase	1.740	Tourmaline, Red	1.61 - 1.64
Chrome Tourmaline	1.61 - 1.64	Honey, 21% water content	1.484	Peridot	1.635 - 1.690	Tremolite	1.600
Chrome Yellow	2.31	Howlite	1.586	Peristerite	1.525	Tugtupite	1.496
Chromium	2.97	Hydrogen (gas)	1.000140	Petalite	1.502	Turpentine	1.472
Chrysoberyl	1.745	Hydrogen (liq)	1.0974	Phenakite	1.650	Turquoise	1.610
Chrysoberyl, Cat's eye	1.746 - 1.755	Hypersthene	1.670	Q		U	
Chrysocolla	1.500	I		Phosgenite	2.117	Ulexite	1.490
Chrysoprase	1.534	Ice	1.309	Plastic	1.460	Uvarovite	1.870
Citrine	1.532 - 1.554	Iodocrase	1.713	Plexiglas	1.50	V-W	
Citrine	1.550	Iodine Crystal	3.34	Polystyrene	1.55	Wardite	1.590
Clinohumite	1.625 - 1.675	Iolite	1.522 - 1.578	Prase	1.540	Variscite	1.550
Clinzoisite	1.724	Iron	1.51	Prasiolite	1.540	Water (0° C)	1.33346
Cobalt Blue	1.74	Ivory	1.540	Prehnite	1.610	Water (100° C)	1.31766
Cobalt Green	1.97			Proustite	2.790	Water (20° C)	1.33283
Cobalt Violet	1.71			Purpurite	1.840	Water (gas)	1.000261
Colemanite	1.586			Pyrite	1.810	Water (35° C, room temp)	1.33157
Copper	1.10			Pyrope	1.740	Whisky	1.356
Copper Oxide	2.705			Quartz	1.544 - 1.553	Willemite	1.690
Coral	1.486					Witherite	1.532
						Vivianite	1.580
						Vodka	1.363
						Wulfenite	2.300
						Z	

Coral	1.486 -
	1.658
Cordierite	1.540
Corundum	1.766
Cranberry Juice (25%)	1.351
Crocoite	2.310
Crystal	2.000
Cuprite	2.850

D

Danburite	1.627 -
	1.641
Danburite	1.633
Diamond	2.417
Diopside	1.680
Dolomite	1.503
Dumortierite	1.686

Quartz, Fused	1.45843
R	
Rhodizite	1.690
Rhodochrisite	1.600
Rhodonite	1.735
Rock Salt	1.544
Rubber, Natural	1.5191
Ruby	1.757 -
	1.779
Rum, White	1.361
Rutile	2.62

Zincite	2.010
Zircon	1.777 -
	1.987
Zircon, High	1.960
Zircon, Low	1.800
Zirconia, Cubic	2.173 -
	2.21

動作の仕方

オブジェクトの表面下の光の経路を実際に計算することは、事実上不可能でしょう。しかし、これを行う必要はなく、異なるアプローチでの計算を使用できることを示していきます。

レンダラーは、2つのステップでSSSを計算します:

- 最初に表面の明るさの計算を、オブジェクトの前面からだけでなく、その裏側からも行ないます。これは通常のレンダリングと全く同じです。アンビエントオクルージョン、ラジオシティ、ディフューズシェーダの種類、光の色、などが考慮されます。
- 第二段階ではいよいよイメージがレンダリングされますが、この場合はSSSシェーダがディフューズシェーダを置き換えます。ランプの代わりに、計算によるライトマップが使用されます。表面上の点の明るさは、その周囲の点の明るさの "平均" として算出されます。設定に応じては表面全体が計算の対象になり、単に平均を計算するよりも少し複雑になりますが、その背後にある数学に煩わされることはそれほどありません。

それよりも、SSSが別個の光点に何をするのか見てみましょう。

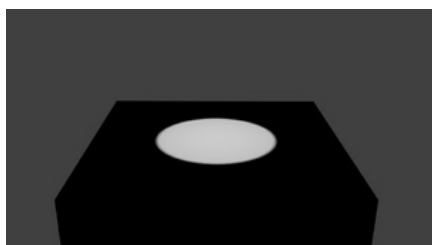


Image 3a: No SSS.



Image 3b: Small SSS radius.

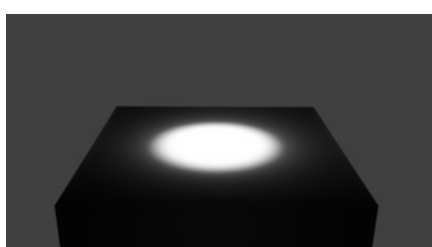


Image 3c: SSS radius enlarged.

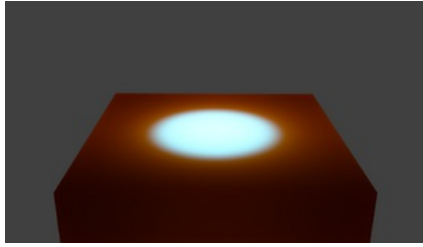


Image 3d: SSS with very large red radius value.

SSSをオンにすると、光が広い範囲に分散されます。この範囲の大きさは半径の値に依存します。すべての色で同じ量を散布するだけでなく、RGB-カラーごとに異なる半径の値を選択することもできます。

色の半径に非常に大きい値を使用する場合、光はオブジェクト全体に均等に分散されます。

表面下散乱を有効にする



Image 4: The SSS Panel.

SSS is already enabled.

- Subsurface Scattering ボタンをクリックして、SSSを有効にします。
- 上部でアクセスできるのは、様々なプリセットです。新規追加と、古いプリセットの削除は、+ と - のボタンのクリックで行えます。プリセットを選択すると、Radius の値、RGB Radius、そして IOR が自動的に設定されます。残りのオプションは設定しないままにしておきましょう(それらはほとんどがあなたのオブジェクトの大きさに依存するものだからです)。

表面下散乱はレイトレーシングを必要としません。ただし、それは入射光と影に依存するものなので、影の計算は適切に行う必要があります(こちらにレイトレーシングが必要になることはある)。

オプション

光の散乱の仕方は数値スライダで制御します:

IOR

Index Of Refraction の値は、入射光の減衰を決定します。値を高くすると光が早く減衰することを意味します。効果は非常にわずかで、分布関数が少し変わるだけです。多くの異なる材料の調査から、1.3 から 1.5 の値が適するとわかっています。あなたが今、正確な材料をシミュレートしようと試行錯誤しているのであれば、[私たちの IOR 一覧表](#)を参照してみましょう。

Scale

Blender単位でのオブジェクトのスケールで、散乱効果を起こしたい場所の幅。Scale 1.0は、1 Blender単位が 1 ミリメートルに等しいことを表し、scale 0.001 は 1 Blender単位が 1 メートルに等しいことを意味します。シーンのなかでどのくらいのスケールの値を使用したら良いかを導き出したいのであれば、ぜひ以下の式を使用しましょう: (レンダラー単位でサイズ)/(ミリメートル単位での現実世界のサイズ)=スケール。



The SSS Color Swatch

<Swatch>

色見本には、2つの効果があります。

1. SSSを一種の風変わりなランプと考えると、これはライトの色といえます。
2. また散乱にも影響を与えます — 暗い色にするほど多くの光が散乱されます。

なのであなたが緑に設定した場合、オブジェクトの点灯領域は緑色で表示され、緑が少しだけ散乱されます。したがって、暗い部分は、赤と青で表示されます。色の半径を大きく設定することによって、異なる散乱を補正することもできます。

RGB Radius

光をぼかす半径。光が物体を通過したり、反転して表面の他のどこかの点から現れるとき、パスの長さが生成されます。これらのスライダーで、そのパスの平均の長さを調整することができます。パスが長くなるほど、色の分散はより均等になります。

Blend

Color

これが制御するのは、R,G,Bオプションでディフューズ色とテクスチャを調節する度合いです。たとえこのオプションが **0.0** に設定されていても、まだR,G,Bオプションは散乱の挙動に影響を与えるということに注意してください。

Texture

シェーディングにともなって表面のテクスチャをぼやけさせる度合い。

Scattering Weight

Front

前方散乱を増減させる係数。オブジェクトの正面から光が入射するとき、どのくらいが吸収、または追加されるでしょうか? (通常は **1.0** または **100%**)。

Back

後方散乱を増減させる係数。後ろからオブジェクトに当たる光は、オブジェクトをはるばる通り抜けて、オブジェクトの前面に出てくることがあります。これは主に、手や耳のように、薄いオブジェクトに発生します。

Error

このパラメータは、周囲のポイントをどのくらい正確なアルゴリズムでサンプルするかを制御します。**0.05**のままにしておけば、乱れない画像が得られるはずですが、高く設定すると、レンダリングはスピードアップしますが、潜在的なエラーをとまいません。**1.0** に設定すると、外観をすばやくレビューするには良い方法となりますが、エラーをとまいません。

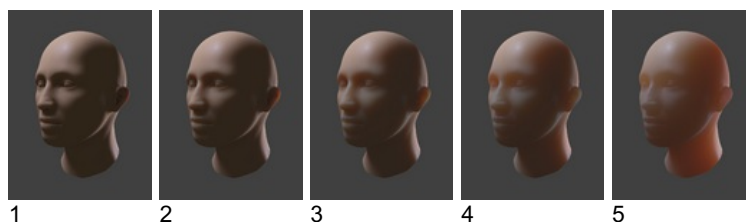
自作のSSSマテリアルの開発

自作のSSSマテリアルを作るためには、これらの簡単な手順に従いましょう:

- SSSの色をお好みの値に設定しましょう。通常、オブジェクトの支配的な色となります。色ごとに異なる半径を使用する場合は、暗くなりすぎないようにしましょう。
- スケール係数を設定しましょう。もっと透明に見えるようにしたい場合は、小さなオブジェクトにするかまたは値を大きくする必要があります。
- 半径値を設定しましょう。
- FrontとBackの値で明るさを調整しましょう。

例

肌



Increasing SSS scale ([.blend](#))

関連項目

- [Development Release Log: Subsurface Scattering](#)

- [Ben Simonds: *Three Layer SSS in Blender Demystified*](#)

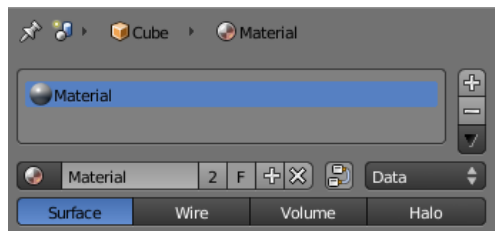
Materials

Mode: All Modes

Panel: Shading/Material Context → Material

マテリアルは、シェーディング/マテリアル・コンテキストの、マテリアルパネル内で、オブジェクトやオブジェクトのデータにリンクさせることができます。この場所では、マテリアルをどのようにオブジェクト、メッシュ、などにリンクするのかを管理でき、そしてパネルの残りの部分では、編集のためのマテリアルを有効化することができます。

Context

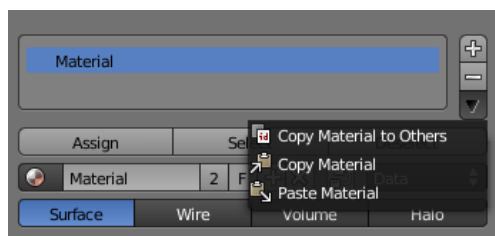


Material added in object mode

マテリアルメニューの上部にあるアイコンのリストは、マテリアルが編集されるときにコンテキストを説明するものです。上記の例では、Material というマテリアルが Cube というオブジェクトにリンクされ、キューブは Scene というシーンにリンクされています。

左側のピン留め記号のオンとオフを切り替えることで、選択しているマテリアルだけを表示したり、状況に従うように、Blender に指示することができます。

マテリアルスロット



Copy and paste dropdown menu

マテリアルがリンクされるか、または作成されると、1つまたは複数のマテリアルスロットが作成され、さらにオプションが利用できるようになります:

- + sign
新しくマテリアルスロットを追加したり、または選択したものをコピーする
- - sign
選択しているマテリアルスロットを削除する。
- Down arrow
選択しているマテリアルスロットをコピーして貼り付ける

Multiple materials

メッシュは、複数のマテリアルを持って扱うことができます。 [Multiple Materials](#) のページで詳述するように、マテリアルは、土台となるそれぞれの面の上にマッピングすることができます。エディットモードでは、以下のツールが表示されます:

Assign

選択しているマテリアルスロットにあるマテリアルを、選択している頂点に割り当てる

Select

選択しているマテリアルスロットに割り当てられている頂点を選択状態にする

Deselect

選択しているマテリアルスロットに割り当てられている頂点の選択を解除する

マテリアルの命名とリンク

マテリアルの名前欄

このフィールドをクリックして、マテリアルの名前を変更する

Number of users (number field)

マテリアルを使用するオブジェクトまたはオブジェクトデータの数。このマテリアルは、さまざまなオブジェクト間でリンクされており、編集すると、それらすべてで横断的に更新されます。この番号をクリックすると、'単一のユーザー・コピー'を生成し、マテリアルを複製して、それに併せてアクティブなオブジェクト/オブジェクトデータのみリンクします。

F (Fake user)

マテリアルの '偽ユーザー'を仕立てて、実際の使用主がない場合でも、マテリアルのデータブロックを.blendファイルに保存したままにします。

プラス記号

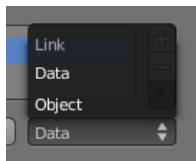
新規にマテリアルを追加します。

X sign

このマテリアルへのリンクを削除します。

Nodes

このマテリアルの指定を、Material/Ramps/Shaders の設定からではなく、マテリアルノードのnoodleで行うようにします。



Link material to object or to object's data

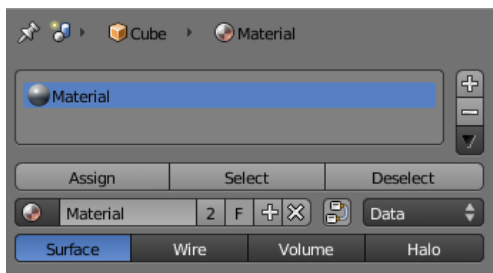
Datablock links

リンクのポップアップメニューには2つの選択肢があります。データとオブジェクトです。これら二つのメニューの選択肢は、マテリアルを、オブジェクトまたはデータ、(このケースでは)メッシュ (or curve, nurbs, etc.)にリンクさせるかどうかを決定します。Data のメニュー項目では、オブジェクトのデータブロックにリンクされているメッシュのデータブロックを、このマテリアルにリンクさせるのかを決定します。Object のメニュー項目では、マテリアルをリンクするオブジェクトのデータブロックを直接決定します。

これはもちろん重要です。たとえば、異なるオブジェクトが同じメッシュのデータブロックを共有することがあります。このデータブロックがオブジェクトの形状を定義するので、編集モードでのあらゆる変更はこれらのオブジェクトすべてに反映されます。そのうえ、そのメッシュのデータブロックにリンクされているあらゆるものが、そのメッシュを共有しているすべてのオブジェクトで共有されます。なので、マテリアルがメッシュにリンクされている場合、すべてのオブジェクトがそれを共有することになります。

一方、マテリアルが、オブジェクトのデータブロックに直接リンクされている場合、同じメッシュを共有していても、オブジェクトは異なるマテリアルを持つことができます。短い説明: オブジェクトに接続されている場合、異なる材料を使用して、同じobjDataの複数のインスタンスを持つことができます。メッシュデータにリンクされている場合、できません。

Material type



Material added in edit mode

これらのトグルは、このマテリアルをレンダリングパイプラインにフィットさせ、そしてどのような様相のマテリアルをレンダリングするのか、といったことをBlenderに指示します。

Surface

オブジェクトをサーフェスとしてレンダリングする

Wire

面のエッジを線として描画する (レイトレーシングではサポートされません)

Volume

オブジェクトを volumeとしてレンダリングします。 [Volumes](#)参照

Halo

オブジェクトを後光の粒子としてレンダリングします。 [Halos](#)参照

Material Properties Overview

マテリアルプロパティの各セクションの使用方法については、次のセクションで詳述されています。

Surface and Wire materials

サーフェスのマテリアルタイプは、最も一般的なマテリアルです。これらは、定義表面を持つオブジェクトを表します。

Wire マテリアルは、単にオブジェクトのすべてのエッジを棒に変えて、レンダリングに反映できるようにします。ただし、シェーディングオプションはサーフェスマテリアルと同じものを使用します。

プレビュー

このプレビューは、いくつかのオブジェクトの内の1つにマップされた現在のマテリアルです。

- Flat Plane
- Sphere
- Cube

- Monkey
- Strands
- Large Sphere with Sky

Diffuse

散乱シェーディングは、光が表面にあたって、非常に広い角度に跳ね返るところをシミュレートします。ディフューズシェーディングの色を設定したり、拡散の計算にどのモデルを使用するかを設定できます。

Specular

スペキュラは、多くの場合、鋭い、明るい点光源の反射をシミュレートします。スペキュラシェーディングの色を設定したり、鏡面反射の計算にどのモデルを使用するかを設定することもできます。

Shading

- Emit

余分な照明を追加して、マテリアルが光っているかのようになります。

- Ambient

マテリアルが受け取るグローバル環境光を設定します。

- Translucency

後ろ側に透けるシェーディングの量。葉や紙のように、薄いオブジェクトをシミュレートするために使用します。

- Shadeless

これは、任意のシェーディングの計算を無効にし、色の情報だけを表示させます。これは、本質的には "surface shader" を作ります

- Tangent Shading

シェーディングに法線の代わりにマテリアルの接線ベクトルを使用します — 異方性シェーディング効果のために (例えば、柔らかい毛ブラシや金属)。このシェーディングは2.42で導入されました。また、メニュー内のストランドのレンダリングの設定については、さらに下のパーティクルシステムのメニューを参照してください。

- Cubic Interpolation

ディフューズ値に対してキュービック補間を使用し、明るい領域と暗い領域の間の変わり目をスムーズにします

Transparency

オプションを設定すると、光がオブジェクトを通過できるようになる

Mirror

ここでは、反射のあるマテリアルのためのオプションを設定できます。

Subsurface Scattering

サブサーフェススキャタリングは、半透明のオブジェクトに光が入射し、周りにはね返り、その後異なる場所に出てくる場所をシミュレートします。例としては、ろうそく、人間の皮膚、チーズ、など。

Strand

毛皮や髪の毛のマテリアルをレンダリングする際にこれらの設定が使用されます

オプション

- Traceable

マテリアルを、レイトレーシング計算できるようにし、反射と屈折を可能にします。

- Full Oversampling

マテリアルが、アンチエイリアシングのサンプルすべてにおいて、シェーディングとテクスチャをフルにレンダリングするよう強制します。

- Sky

マテリアルをアルファなしでレンダリングし、背景を空と置き換える

- Use Mist

このマテリアルにミストを使用します。

- Invert Z Depth

マテリアルの表面を、反転したZバッファを使ってレンダリングします。

- Z Offset

Invert Z Depthを使用している場合、これがzの値の人為的なオフセットになります。

- Light Group

特定の照明グループに対して、マテリアルの照明計算を制限する

- Exclusive

マテリアルが照明グループを排他的に使用します

- Face Textures

オブジェクトのベースカラーを、画像テクスチャが割り当てられた面を持つ色で置き換えます。

- Face Textures Alpha

オブジェクトのベース・アルファ値を、画像のテクスチャが割り当てられた面を持つアルファで置き換えます。

- Vertex Color Paint

頂点カラーを使用してオブジェクトのベースカラーを置き換えます。

- Vertex Color Light

頂点カラーを、付加的な照明として追加します。

- Object Color

オブジェクトごとの色で結果を調節する。

Shadow

- Receive

他のオブジェクトからかかる影を、マテリアルが受けられるようにする

- Receive Transparent

他の透明なオブジェクトからかかる透明な影を、マテリアルが受けられるようにします。

- Cast Only

マテリアルを持つオブジェクトにだけ影を落とすようにし、レンダリングには表示されないようにします。

- Casting Alpha

影にかかるアルファを設定します。不規則な、深いシャドウのバッファリングに使用されます。

- Shadows Only

マテリアルのアルファ値と同様に影をレンダリングし、影の部分を除いて、マテリアルを透過的にします。

- Shadow Only Type

Shadows Onlyが有効になっているときに使用される影のタイプを設定する

- Shadow and Distance
- Shadow Only
- Shadows and Shading

- Cast Buffer Shadow

バッファランプからマテリアルに影がかかるようにできます。

- Buffer Bias

シャドウバッファを乗算する係数。

- Auto Ray Bias

滑らかな法線を持つ表面でのレイトレースシャドウのエラーを防止する

- Ray Bias

影の境界における、明暗境界線の歪みを防ぐための、レイトレーシングのバイアス値。

- Cast Approximate

Approximate Ambient Occlusionを使用するときに、マテリアルに影を落とすようにする

Volume Material

Volume materials は、雲や煙のように、小さな粒子からなるものの体積を表すものです。それらは標準的な材料とは非常に異なるのですが、[Volume Page](#)で詳述しています。

Halo Material

Halo materials は、オブジェクトの各点を、光るドットとして描画します。これはパーティクルエフェクトやレンズフレアをシミュレートするときに役立つマテリアルです。[Halo Page](#)に詳しく説明されています。

Introduction

マテリアルの作成は、前章までに説明したような、使用するすべての設定をマテリアルパネル上で行う方法に加えて、Blenderでは、ノードのセットを介して基本的なマテリアルをルーティングする方法でもマテリアルの作成を行えます。各ノードは、マテリアルに何らかの操作を実行し、メッシュに適用されるときの変数の仕方に変更を加え、そして次のノードに進みます。この方法では、非常に複雑なマテリアルの外観を達成することができます。

マテリアルの一般的なコンセプトや、マテリアルメニューを使用したマテリアル/テクスチャの作成方法はもうお馴染みでしょう。また、Blenderで利用可能なテクスチャの座標系についてもだいたい理解しているはずですが (例えば、Generated, UV, など)。また、ノードの多くの側面は、後のセクションで拡張機能として見ていくことになるので、ここではスキップします。各セクションは、以前のものを増築する形で構成されています。

はじめに言っておくと、ノードシステムによって、マテリアルメニューが使われなくなることはありません。多くの機能やマテリアルの設定は、いまでもマテリアルパネルを介してのみアクセス可能です (例えば、Ray Mirror)。しかし、ノードの登場で、大幅な制御が行えるようになるので、さらに複雑で幻想的なマテリアルを作成できるようになります。

念のため、コンセプトに (まだ) 馴染みがない方もいるかもしれないので: ノードのシステム (もしくは "noodle"とも言われる)を作成するときは、ある種のデータ処理パイプラインを描くことになり、そこでのデータは様々な *sources* を示すノード"から流れてくる"もので、様々な処理やフィルタリングの段階を表すノードを"流れぬげ"ながら、最終的に出力または目的地を表すノードに"流れ着き"ます。ノードをお互いに接続するには多くの異なる方法があり、"ノブ"、あるいはパラメータを調整することで、各ノードの動作を制御できます。これにより、とても大規模に製作時のコントロールを行えます。そして、非常に迅速で直感的になります。

前置きはこのくらいにして、通常のマテリアルから始めていきましょう。

ここにあるのは、標準的なマテリアルをキューブメッシュに追加したものです。いままでと同様のやり方でも、このマテリアルに色や様々な設定を追加することはでき、素敵な見栄えにはなるでしょう。しかし満足いくものを得られないとしたら? もっと緊密に製作を制御したり、今以上の複雑さを追加したい場合は? ここからがノードの出番です。

このノードマップを作成するには、[Node Editor window](#) で作業を遂行します。このセクションで扱うのは:

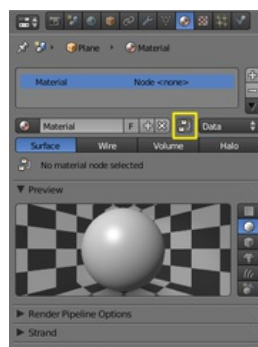
- マテリアルノードの有効化。
- ノードエディタウィンドウ、その基本的なコントロール、およびノードでの作業。
- マテリアルに使用可能な特定のタイプのノード。

ノードエディタへのアクセス

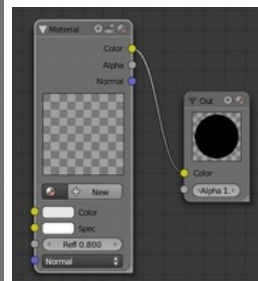
まず [node editor](#) に入り、ノードエディタにあるマテリアルノードボタン (球のアイコン) が押されていることを確認しましょう。コンポジットボタンやテクスチャノードボタンではないことに注意。

マテリアルボタンでノードマテリアルを有効にする

ベースマテリアルを探り当て、マテリアルパネルかノードエディタのマテリアル名の横にある Nodes ボタンを押してみましょう。マテリアルパネルに変更が表示されます。

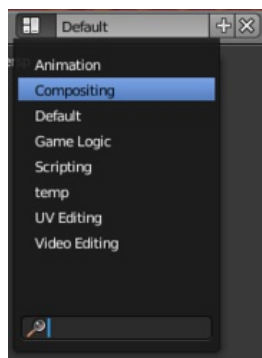


Material's menu with Nodes enabled



Default nodes

ちょうど行った操作がBlenderに伝えられ、今いるところにマテリアルが作成されて、ノードツリーになります。普段マテリアルメニューで見かけるパネルはほとんど、今は無くなっています。



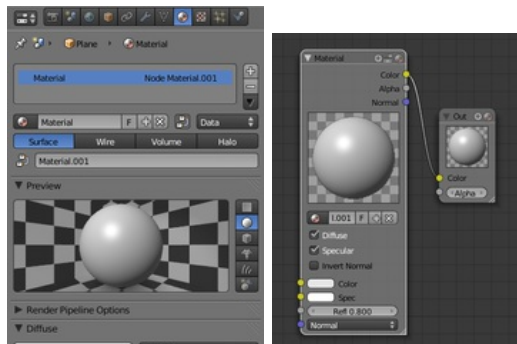
Accessing the Compositing screen

Compositing画面に切り替えると (Ctrl← デフォルトの画面上に表示されている場合)、画面の上半分に Node Editorが見つかるはずですが、マテリアルノードを有効化すると、マテリアルノードと出力ノードが自動的にノードエディタに追加されます。

また、デフォルトの画面で3Dビューを2つに分割し、1つを Node Editorに変更することもできます。

新しい素材の追加 (マテリアルパネル内の他のマテリアルと同様に編集して変更できる)、あらかじめ作成していたマテリアルの追加、または別のBlenderファイルからのマテリアルを付加、などを行えることを覚えておくのは重要です。そしてまた、ノードツリーの作成に使ったマテリアルを使用することもできます。

これで、Node editor (*Material.001*)に新しいマテリアルが追加されました。そして今やったように、このマテリアルのメニューで、マテリアルプロパティにアクセスすることができます。



Material's menu with a first material added to the Node Editor

外部リンク

- [Blender Material Nodes](#) - Changelog for the Blender version that introduced material nodes.

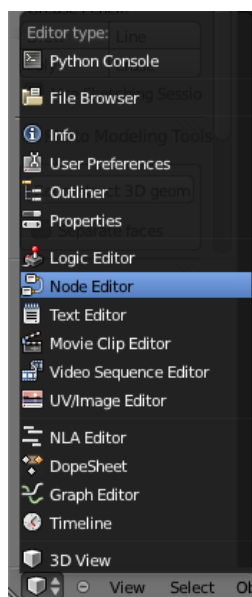
The Node Editor

(ノードエディタ)

このページでは通常のウィンドウと、ヘッダーメニューにあるオプション機能について説明します。Blender 内でノードを有効化する方法も説明します。

Accessing The Node Editor

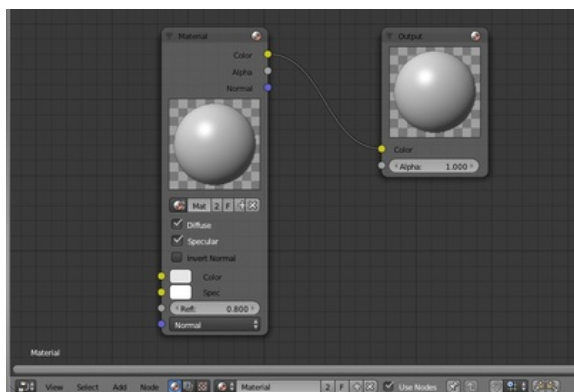
(ノードエディタを開く)



Node Editor ウィンドウの
選択

スクリーンレイアウトから「Compositing」を選ぶと、ノードエディタを含むウィンドウ配置になります。もしくは、任意のウィンドウのタイプを Node Editor に変えて、ノードエディタに入りましょう。図「Node Editor ウィンドウの選択」にあるように、ウィンドウタイプアイコンをクリックし、ポップアップリストから Node Editor を選びます。ノード図は非常に大きくなることもあるので、広いウィンドウを使うか、大きめのウィンドウを作ってください。ウィンドウには方眼紙様式の背景と、ヘッダーがあります。

blend ファイルにある各シーンは、複数のマテリアルノード図と「一つの」コンポジティングノード図を持つことができます。ノードエディタウィンドウにはノード種類セレクターで選択されている、いずれかのノード図が表示されます。



ノードエディタ

デフォルトではヘッダーの最初の表示は次のようになります:



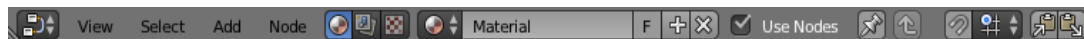
デフォルトのノードエディタのヘッダー

Activating Nodes

(ノードの有効化)

- どのノードを使うべきか？
 - マテリアル(シェーダー)ノード図で作業したいときは、ノード種類セレクターの球体のアイコンを選択します(図「マテリアルノードが有効なノードエディタのヘッダー」)
 - テクスチャノード図で作業したいときは、ノード種類セレクターのチェッカー模様のアイコンを選択します(図「テクスチャノードが有効なノードエディタのヘッダー」)
 - コンポジティングノード図で作業したいときは、ノード種類セレクターの写真の重なったアイコンを選択します(図「コンポジティングノードが有効なノードエディタのヘッダー」)
- ノードを実際にも有効化するには Use Nodes ボタンを押します

- 最初にマテリアル(シェーダー)、コンポジティングノード、テクスチャノード図のいずれかを選んだ時は、ノードエディタウィンドウにはすでに結合された入力ノードと出力ノードが置かれます。



マテリアルノードが有効なノードエディタのヘッダー



コンポジティングノードが有効なノードエディタのヘッダー



テクスチャノードが有効なノードエディタのヘッダー

Node Editor Window Actions

(ノードエディタウィンドウでの操作)

カーソルがウィンドウ内であれば、次のような Blender の標準的なホットキーやマウス操作を使えます:

ポップアップメニュー

- ⇧ ShiftA - 追加するノードを選ぶポップアップメニューを表示します。

削除

- X か Del - 選択ノードを削除します。

ボックス選択

- B - 範囲選択を開始します。カーソルを移動して LMB クリックし、ドラッグして複数ノードを選択します

接続の切断

- Ctrl LMB クリックしてドラッグ - アイコンが小さなナイフに変わり、ノード間を接続するスレッドをまたぐようにドラッグして切断します

取り消し (Undo)

- CtrlZ を押し忘れてボックス選択をしようとしたとき、便利です。

やり直し (Redo)

- CtrlY か Ctrl⇧ ShiftZ - ちょっと「Undo」しすぎたときに使えます :)

複数選択

- ⇧ Shift LMB か ⇧ Shift RMB - 複数ノードを選択します。

つかむ/移動

- G - 選択中のものを動かします

通常のウィンドウ操作

ノード図は非常にひげもじゃ(広くて入り組んでいるという意味です)になる可能性があります。ウィンドウの中身(ノード図)は他の Blender のウィンドウと同じように、MMB クリックしてドラッグすると平行移動することができます。Wheel を上下させるかテンキーの + NumPad/- NumPad を使うと、ズームイン/ズームアウトができます。通常のウィンドウと同じようにウィンドウのサイズを変えたり、結合させたりできます。

Node Editor Header

(ノードエディタのヘッダー)

At a glance

(項目一覧)

ウィンドウのヘッダーには次のオプションがあります:

- View (ビュー) - はっきり見たいものを選びます
- Select (選択) - 条件をつけてノードを選択します
- Add (追加) - ノードを追加します。種類別にまとめられています
- Node (ノード) - 選択したノードに対する操作です。頂点のメニューに似ています
- マテリアル/テクスチャ/コンポジティングノードセレクタ
- Use Nodes (ノードを使用) ボタン
- ピン止め ボタン
- 親ノードに移動 ボタン
- スナップ ボタン
- スナップノードセレクタ
- コピーノード ボタン
- ペーストノード ボタン
- Free Unused (未使用ノードを解放) ボタン
- Backdrop (背景) ボタン
- Auto Render (自動レンダリング) ボタン

Menus

(メニュー)

View, Select and Add

(ビュー、選択、追加)

基本機能を持つメニューです:

View(ビュー)

ウィンドウの視点を変えるメニューです。+ NumPad (ズームイン)、- NumPad (ズームアウト)、 \wedge Home (すべてを見渡せるまでズーム)といった標準的なキーボードショートカット、または同等のマウス操作の代わりになります。

Select(選択)

ノードやノードグループを選ぶメニューです。全選択 A やボックス選択の開始 B といったホットキーの入力と同じ操作ができます。

Add(追加)

ノードを追加するためのメニューです。追加できるノードの種類やその役割については、次の節をご覧ください。このメニューにある項目は、ウィンドウ内にカーソルがあるときに \diamond ShiftA を使うのと同じものです。

Node

(ノード)

Hide(隠す)

H - 選択ノードを隠します。メッシュ操作と同じ要領です。

Grouping(グループ作成)

最重要項目で、ユーザー定義のノードグループを作ることができます。作成したグループは編集したり、ノード図に追加したりできます。グループを作るには、対象ノードを選び、Node(ノード) » Make Group(グループ作成) を選ぶか、キーボードショートカット CtrlG を使います。名前はグループ内にある小さな入力欄で変更できます。緑色のヘッダーや、あなたのつけたクールな名前によってグループが簡単に見分けられます。

Delete(削除)

X - 選択ノードを削除します。

Duplicate(複製)

\diamond ShiftD - 元のものと同じ設定を持ち、元のものとはつながっていない複製を作ります。

Grab(つかむ/移動)

G - ノードをマウスで動かします。メッシュ操作と同じです

複製されたノード

ノードを複製すると、新しいノードは **ぴったり元のノードの上に重なります**。下にあるノードにスレッドがつながっているように見えたとしても、複製はつながっていないので、コントロールをいじっても画像には何も起きません。複製を動かして下にあるつながったノードを見えるようにする必要があります。

Buttons

(ボタン)

Material/Composite/Texture Selector

(Material/Composite/Texture セレクター)

ノードは操作対象によってグループ分けされています:

- [Material Nodes \(マテリアルノード\)](#) / [Blender 2.67 では Shader Nodes \(シェーダーノード\)](#) で作業するには球体のアイコンをクリックします。
- [Compositing nodes \(コンポジットングノード\)](#) で作業するには写真の重なったアイコンをクリックします。
- [Texture nodes \(テクスチャノード\)](#) で作業するには、チェッカー模様のアイコンをクリックします。

Use Nodes Button

(ノードを使用)

有効にするとマテリアル色の計算や最終的な画像のレンダリングにノード図を使うよう、レンダリングエンジンに指示します。チェックを外すとノード図は無視され、マテリアルタブやシーンの基本のレンダリングが行われます。

Use Pinned Button

(ピン止めを使用)

レンダーエンジンにピン止めされたノードツリーを使うよう指示します。

Go to Parent Button

(親に移動)

親ノードツリーに移動できます。

Snap Button

(スナップ)

ノードエディタウィンドウ内でのノードのスナップモードを切り替えます。

スナップノード要素 セレクタ

スナップ対象要素を次から選べます:

Grid (グリッド、デフォルト)

ノードエディタウィンドウのグリッドにスナップ

Node X (ノードX)

ノードの左右の縁にスナップ

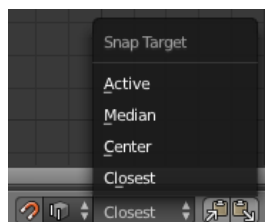
Node Y (ノードY)

ノードの上下の縁にスナップ

Node X/Y (ノードX/Y)

ノードの縁にスナップ

ノード縁へのスナップは次のスナップターゲットを考慮します:



Snap Target

Snap Target

どの部分をスナップさせるか選べます。

Closest (近接): 一番近い位置をスナップ

Center (中心): 中心をスナップ

Median (中点): 中点をスナップ

Active (アクティブ): アクティブ要素をスナップ

Copy Nodes Button

(コピーノードボタン)

選択ノードをクリップボードにコピーします。

Paste Nodes Button

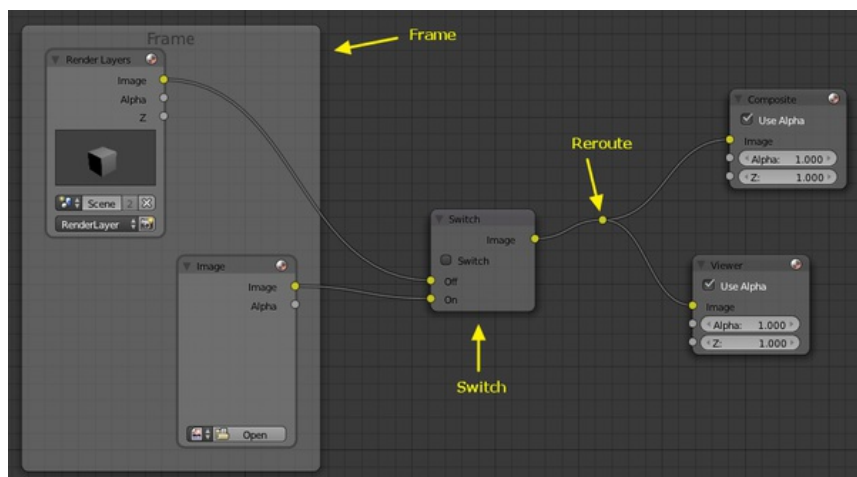
(ペーストノードボタン)

アクティブなノードツリーに、クリップボードからノードをペーストします。

Layout Nodes

(レイアウトノード)

Layout ノードはノードエディタウィンドウのノード配置の改善用に設計されています。Add (追加) » Layout (レイアウト) から利用できます。



レイアウトノードの例

Blender は次のレイアウトノードを備えます:

Frame (フレーム)

このノードは単に1箇所に複数ノードをまとめるために使います。

Reroute(リルート)

複数スレッドを枝状にしてノードエディタウィンドウのスペースを節約するためのものです。

Switch(スイッチ)

カラー(色)値の切り替えに使われるノードです。コンポジットノードでのみ使えます。

Free Unused Button

(未使用ノードを解放)

(コンポジットノード専用)非常に複雑なノード図で、未使用ノードのメモリ空間を解放します。お勧めです。

Backdrop

(背景)

(コンポジットノード専用)アクティブな Viewer ノードの出力を背景に使用します。有効にすると、ヘッダーとプロパティパネルにさらに設定が現れます:


Backdrop Channels(背景に使うチャンネル)

画像の Color、Color and Alpha または Alpha だけを表示します。

Zoom

背景画像の大きさを指定します

Offset

背景の画面に対する位置を変更します。Move ボタンを押すかショートカット Alt MMB  を使って実際に動かしながら変えることもできます。

Auto Render

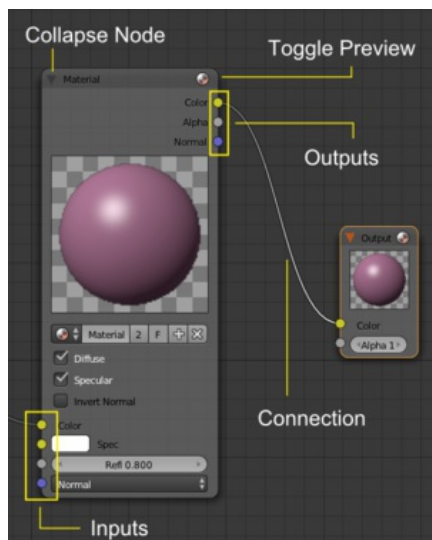
(自動レンダリング)

(コンポジットノード専用)3Dシーンで編集が行われると、変更のあったレイヤーの再レンダリングと合成を行います。

Node Controls

(ノードのコントロール)

このページではノードを操作するウィジェットを説明します。



ノードの主なコントロール

上から Collapse Node: 折り畳み Toggle

Preview: プレビュー切り替え Outputs: 出力

Connection: 接続 Inputs: 入力

タイトルバー

ノード名が表示され、折り畳み用のボタンがあります。

入力 (Input) ソケット

ノードの左側には入力ソケットがあります:

- 青色ソケットはベクトルを受け入れます
- 黄色ソケットは色を受け入れます
- 灰色ソケットは単独の値を受け入れます (例: アルファ)

出力 (Output) ソケット

ノードの右側には出力ソケットがあります:

- 青色ソケットはベクトルを作ります
- 黄色ソケットは色を作ります
- 灰色ソケットは単独の値を作ります (例: アルファ)

画像プレビュー

ノード内にはこのノードが出力する画像のプレビューやノードの動作を制御する曲線 (例えば RGB ノードにあるもの) を表示する領域があります。

ボタンとメニュー

画像プレビューの下には、ノードの動作を制御するボタンとメニューがあります。

スレッド

曲がった線は出力ソケットから入力ソケットへの接続を示しています。ソケットの種類は一致しなければなりません。

アクティブノードに関連した接続は見やすくなるようにハイライトされます。

Collapsing toggles

(折りたたみ表示の切り替え)

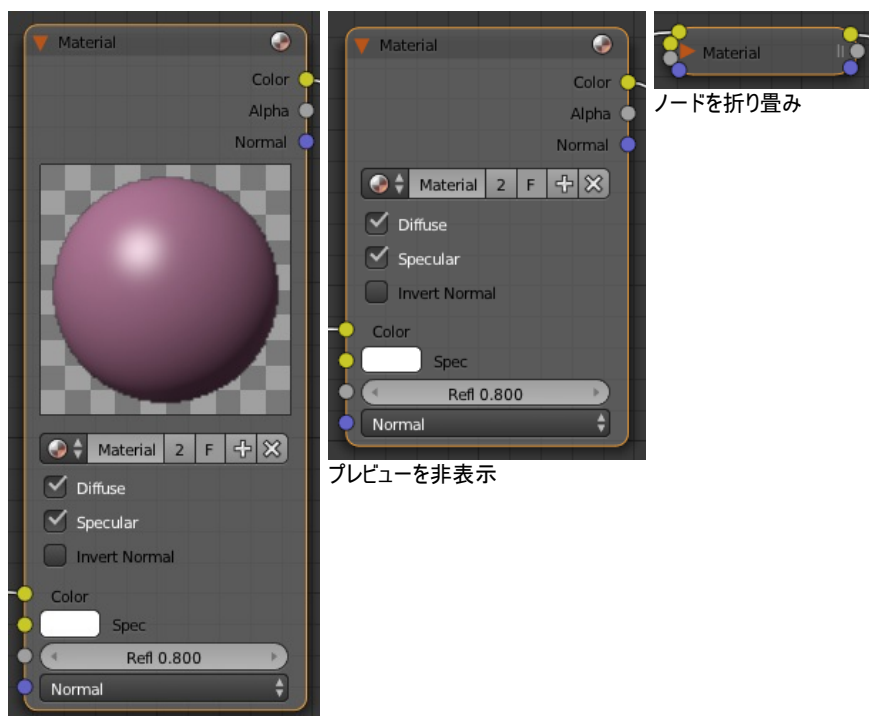
ノードの最上部には最大4つの外見に関わるコントロールがあります。クリックするとノードが表示する情報量に作用します。

ノード切り替え (▼▶)

左にある三角形はノードを折り畳み/展開します。

プレビュー画像の切り替え

タイトルバーの一番右にある球のボタンはプレビュー画像の表示/非表示を切り替えます。




すべて表示

プレビューを非表示

ノードを折り畳み

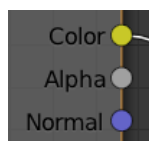
Sizing the node

(ノードのサイズ調整)

ノードの左右の端を LMB  クリックしてドラッグするとノード個別にサイズの調整を行えます。

Sockets

(ソケット)



ノードソケット

ノードウィンドウにある各ノードは色のついた小円形の「ソケット」(「コネクタ」ともよく呼ばれます)を持ち、入力データや出力データをつなげられます(図「ノードソケット」)。

ノード左側にあるソケットは *入力*を、右側にあるソケットは *出力*を表します。

便利のようにノードは送信/受信することを期待される情報の種類によって *色分け*されています。三つの色があります:

- **黄色ソケット**
色 情報が入力に必要、またはノードから出力されることを示します。
- **灰色ソケット**
値(数) 情報を示します。単独の数値かいわゆる「値マップ」を使えます(値マップは各点の値が明暗に反映されるようなグレースケールのマップだと考えることができます)。「値マップ」ソケットの入力として単独の値が使われると、マップのすべての点がこの値と同じ値になります。
一般的な用途: アルファマップやノード用の値オプション。
- **青/紫色ソケット**
ベクトル/座標/法線 情報を示します。

ノード間は、コンバータ(*converter*)を使わない限りは、黄は黄に、灰は灰に、青は青に繋げなければなりません。コンバータについては後ほど説明します。

ノードの色名の隣にはソケット名が表示されています。常にとは言えませんが、ソケットが何の情報を表すものか名前から想像できます。しかし必ずしも「従わねばならない」ものではありません。例えば Alpha と書かれた灰色ソケットから Material ノードの灰色の Reflection ソケットにつなぎ、結果を得ることができます。重要なのはこれが「灰色から灰色」への接続であるということです。

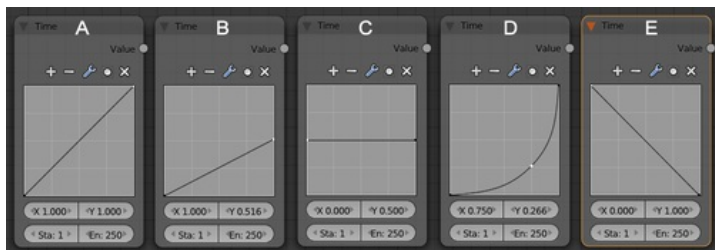
黄色(例: カラー画像)と灰色(例: グレースケール)をコンバータなしで組み合わせられる、例外があります。必要なら普通は Blender がコンバータを置くので、気軽に試してみてください。この後で説明する "Viewer" 出力ノードを使うと、動作するかどうかや、動作の様子を見ることができます。

Curves

(曲線)

ノードによっては入力値を出力値に変換するための曲線領域を持つものがあります。曲線の形状は制御点をクリックして動かすか、制御点を追加し

て編集できます。以下は例です:



Curve ノードの変形

どの曲線も傾き 1 の直線が始まります。曲線は始め両端に小さな黒い制御点を持っています。制御点を LMB クリックすれば選ぶことができ、制御点は白くなります。

曲線の変更は出力の生成のされ方に影響します。入力 X は普通 底の軸に交差して線形に(一定の間隔を持って)進みます。曲線にぶつかるまで上に進み、ぶつかれば右に進んで、X に対応する Y 出力を決めます。二つ目の例では、X が底に交差して 0 から 1.0 に進むに連れ、Y は 0.0 から 0.5 まで変化しています。3つ目の例では X は 0.0 から 1.0 に進みますが、Y は 0.5 に留まり続けています。したがって、上の図の曲線は次のような時間的効果を持ちます: **A** は影響なし、**B** は遅延させ、**C** は停止、**D** は加速、そして **E** は時間を逆向きにします。

「曲線」ウィジェットはどこでも使える Blender UI の組み込み機能で、与えた曲線データはこのウィジェットに渡されています。現在のところノードエディタと UV ウィンドウ内で使われています。

ウィジェットは水平方向に入力値をマップし、曲線の高さで示される新たな値を返します。

註釈: スクリーンショットでは曲線上の点が一「白く」なっていますが、深い意味はありません。単に、著者がチュートリアルを準備しているときに、たまたま最後に選択した点だったというだけです。ここで重要なのは曲線の形状であって、それを定義するための制御点の位置(または色)ではありません。

RGB Curves

(RGB 曲線)

ひとつのウィジェット内で複数の曲線を編集できます。その代表的な使用法が RGB Curves ノードで、このノードは「すべてを組み合わせた」結果である「色 (C)」を最初の曲線として持ち、R、G、B 要素それぞれの曲線も持ちます。4つの曲線すべてが有効で、「C」の曲線が最初に評価されます。

Selecting curve points

(曲線の制御点の選択)

- LMB は常にひとつの点を選び、残りの選択を解除します。
- Shift を押しながらかlickすると、選択範囲を増やすか少なくすることができます。

Editing curves

(曲線の編集)

- 点を LMB クリックしてドラッグすると点を動かせます
- 曲線を LMB クリックすると新たな点を追加します
- 他の点にちょうど重なるように動かすと一つにできます
- Shift を押しながらかlickするとグリッド単位にスナップします
- Ctrl LMB は点を追加します
- 選択した点を削除するには X アイコンをクリックします

Editing the view

(ビューの編集)

デフォルトではビューは 0.0 から 1.0 の範囲に固定されています。デフォルトの設定である clipping が有効なら、ビューの拡大やドラッグは行えません。丸いアイコンを押して clipping を無効にできます。

- 曲線の外側を LMB クリックしてドラッグするとビューを動かせます
- + と - のアイコンを使ってズームイン/アウトを行います

Special tools

(特別なツール)

レンチアイコンはビューのリセット、各点の補間の定義、曲線のリセットを行うメニューを表示します。

Using nodes

(ノードの利用)

Adding Nodes

(ノードの追加)

ノードは次のいずれかの方法でノードエディタウィンドウに追加します。

- ノードエディタのツールバーにある Add メニューをクリックし、目的のノードタイプを選びます
- カーソルをノードエディタウィンドウに置いた状態で、⇧ ShiftA を押し、ポップアップの Add メニューからノードを選びます


Arranging Nodes


(ノードの並べ替え)

ふつうはウィンドウ内でノードを並べ替えて、画像が左から右、上から下に流れるようにしてください。コントロールなどに影響しない部分をドラッグしてノードを動かします。ノードはほとんどどこでもクリックし、ドラッグすることができます。すべての接続は可能な限りベジエ曲線として再整形されます。

Connecting nodes

(ノードの接続)



ソケットを LMB -クリックしてドラッグすると、枝が伸び出てきます。これは「スレッド」と呼ばれます。

ドラッグしたまま他のノードの入カソケットにスレッドをつなげ、それから LMB  を離します。

この場合各出力のコピーがスレッドに沿って送信されます。一方、ひとつの入カソケットにはひとつのスレッドだけをつなげることができます。

Disconnecting nodes

(ノード接続の解除)

ソケット間のつながりを切るには、切りたいスレッドに近い何も無い空間を Ctrl LMB -クリックしてドラッグします。マウスポインタは小さなカッターナイフのアイコンになります。スレッドを横切るようにマウスを動かして、LMB  を離します。

Duplicating a node

(ノードの複製)

目的のノードをクリックして ⇧ ShiftD を押し、元のノードが見えるようにマウスを動かします。

わかった！

ノードを複製すると、新しいノードは **正確**にもとのノードの上に重なります。そのままにしておく(そうするのは本当に簡単です)、ふたつノードがあることが簡単にはわからなくなります！ 疑わしいときは、ノードをつかんで少し移動し、何か隠れているものがないか確かめてみてください。

Node Groups

(ノードグループ)

マテリアルとコンポジットのどちらのノードもグループ化できます。ノードをグループ化すれば、ノードエディタ上でノード回路網の配置を単純化でき、ヌードル(ノードの回路網)が扱いやすくなります。またノードグループ(.blend ファイル内)またはノードツリー(アペンド時)と呼ばれるものが作られます。

概念的には「グループ化」と「単体のノード」であるかのように扱えるノードの *セット* を定義することができます。ノードグループは同じ .blend ファイル内でも別の .blend ファイルからも、何度でも再利用できます。

例えばノードを使って作ったマテリアルを、別の .blend ファイルに単に追加することができます。しかし新たに作ったマテリアルから、既存のマテリアルノードの回路網から一部の分岐を使いたいときはどうでしょう？ 分岐を再作成することもできるでしょう。もしくはそのマテリアルを新たな .blend ファイルに追加して、使いたい分岐をカット&ペーストして新たなマテリアルに足すこともできるでしょう。どちらの方法もうまくいきますが、.blend ファイル間で作業するときはそれほど効率が良くありません。「被写界深度」のコンポジットノードの回路網を作り、別の .blend ファイルで使いたいときにはどうでしょう？ 一連の操作を正確に何十回も繰り返したいときは？ この場合も回路網の再作成はできますがそれほど効率が良くありません。マテリアルノードの分岐とコンポジットノード回路網のどちらにおいても、ノードグループを作って再利用するのが上手なやり方と言えるでしょう。

一旦定義したグループは、再利用可能なソフトウェア部品になり、ブラックボックス化します。定義を厳密に知らなくても、働きを知れば(何度でも好きなだけ)利用できます。グループは [Blender のライブラリと標準的な追加\(アペンド\)方法](#) を通じて利用可能な状態にできます。

Grouping Nodes

(ノードのグループ化)

パネル: [Node Editor](#)

メニュー: ◊ ShiftA » Group (グループ) » Make Group (グループ作成)

ノードグループを作るには、ノードエディタでグループに含めたいノードを選択し、CtrlG か ◊ ShiftA » Group » NodeGroup を使います。ノードグループは緑色のタイトルバーを持ちます。選択したノードはすべて最小化され、ノードグループ内に入ります。ノードグループのデフォルトの名前は *NodeGroup*、*NodeGroup.001* といったものです。ノードグループには名前を入力欄があり、クリックして変更できます。名前を意味のあるものに変えてください。別の .blend ファイルにノードグループをアペンドしたときに、Blender はマテリアルノードグループとコンポジットノードグループの区別をしません。したがってこの種類を見分けやすくする命名規則を作るのがよいでしょう。例えばマテリアルノードの分岐には *Mat_XXX*、コンポジットノードの回路網には *Cmp_XXX* と命名します。



グループに含むべきではないもの(すべてのタイプのノードエディタ)

覚えておきたいのは基本思想として、グループは簡単に再利用できる、自己完結したソフトウェア部品であるべきだということです。マテリアルノードグループは以下のノードを **含むべきではありません**：

Source nodes (ソースノード)

グループ内にソースノードを含めると、最終的にソースノードがグループ内とグループ外の新たなマテリアルノード回路内の二箇所にてきます。

ソースノードの例としてマテリアルノードの *Material* ノードや、コンポジットノードの *Render Layers* ノードがあります。

Output node (出力ノード)

グループ内に出力ノードを含めると、このグループからの出力ソケットが使えなくなります！

出力ノードの例としてマテリアルノードの *Output* ノードや、コンポジットノードの *Viewer* ノードがあります。

Editing Node Groups

(ノードグループの編集)

ノードグループを選択して ⇄ Tab を押すか、ノードグループ右肩の「ノード」アイコンをクリックすると、ノードがウィンドウ枠いっぱいになり、グループ内にある各ノードが表示されます。各ノードに対して移動や、コントロールの調整、スレッドのつながりなど、通常のエディタウィンドウの一部であるかのような操作ができます。外部にあるノードと直接スレッドをつなぐことはできず、グループノードの端にある外部ソケットを使う必要があります。ノードをグループに追加したりグループから削除するには、一旦グループを解除する必要があります。

Ungrouping Nodes

(ノードグループの解除)

AltG コマンドはグループを壊して、各ノードをエディタの作業領域に置きます。内部の接続は保持され、内部にあったノードを作業領域内にある外側のノードと接続できるようになります。

Appending Node Groups

(ノードグループの追加)

ノードツリーをグループとして .blend ファイルに一旦追加すれば、ノードエディタ内で ◊ ShiftA » Add (追加) » Group (グループ) を選び、追加したグループを選ぶとそのグループを利用できます。グループの「コントロールパネル」はノードグループ個別にあります。他のノードと同じように操作できます。

マテリアルノードの種類

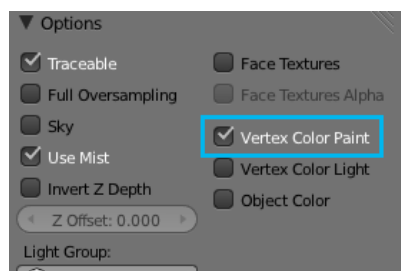
機能別にノードの種類をまとめます:

- [Input/入力 \(eng\)](#)- マテリアルや部品をノード図に取り入れます
- [Output/出力 \(eng\)](#)- 進行中の結果を小さな画像として表示します
- [Color/色 \(eng\)](#)- マテリアルの色を操作します
- [Vector/ベクトル \(eng\)](#)- マテリアルの光の反射の仕方を変えます
- [Convertors/変換 \(eng\)](#)- 色を他のマテリアルカラーに変換します
- [Groups/グループ](#)- ユーザー定義のノードのグループ
- [Dynamic/動的 \(eng\)](#)- Python で定義されたカスタムのノード。PyNodes とも呼ばれます。

Vertex Painting

頂点ペインティングは、オブジェクトの上に色を塗るシンプルな方法です。テクスチャではなく、頂点の色を直接操作するもので、そしてかなり率直です。

頂点カラーを塗るにはVertex Paint Modeに切り替えます。ただし、マテリアルパネルの "Vertex Color Paint" をチェックしない限り、レンダリングに表示されません。



Settings

ショートカットTでよびだすTools Shelfには、頂点ペイントのオプションのほとんどが含まれています。

Brush

最上部のイメージをつかうと、ツールプリセットを選択したり、ブラシの名前を変更するだけでなく、カスタムブラシを追加したり、それらを削除することもできます。

Radius

ブラシの半径を設定します。

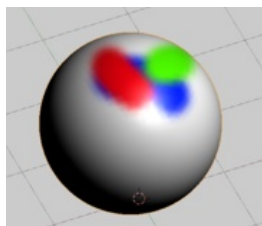
Strength

ブラシの効果の強さを設定します。

Tool

Mix

RGB値を混ぜあわせます。1.0の強さに設定されている場合、下地となる"ペイント"を覆い隠します。



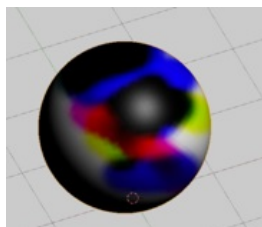
Mix overlay with full strength

Add

RGB値を加算します。最終的には、オブジェクト全体が白色に変わっていく、つまりRGB値が蓄積されて1.0-1.0-1.0になります:ピュアホワイトです。

Subtract

RGB値を減算します。通常は黒に至ります。



Subtract with full strength

Multiply

ブラシの色を頂点色で乗算します。

Blur

頂点の色をぼかします。

Lighten

頂点の色を明るくします。

Darken

頂点の色が暗くなります。

Stroke

Airbrush

ブラシの流れは、マウスクリックが持続する限り継続され、Rate の設定によって決まります。無効になっている場合、ブラシが行えるのはブラシが場所を変えるときの色の変更だけです。

Smooth stroke

ブラシがマウスから遅れて、よりスムーズなパスをたどります。有効にすると、次のオプションもアクティブになります:

Radius

ストロークを続行する前の最後のポイントからの最小距離を設定します。

Factor

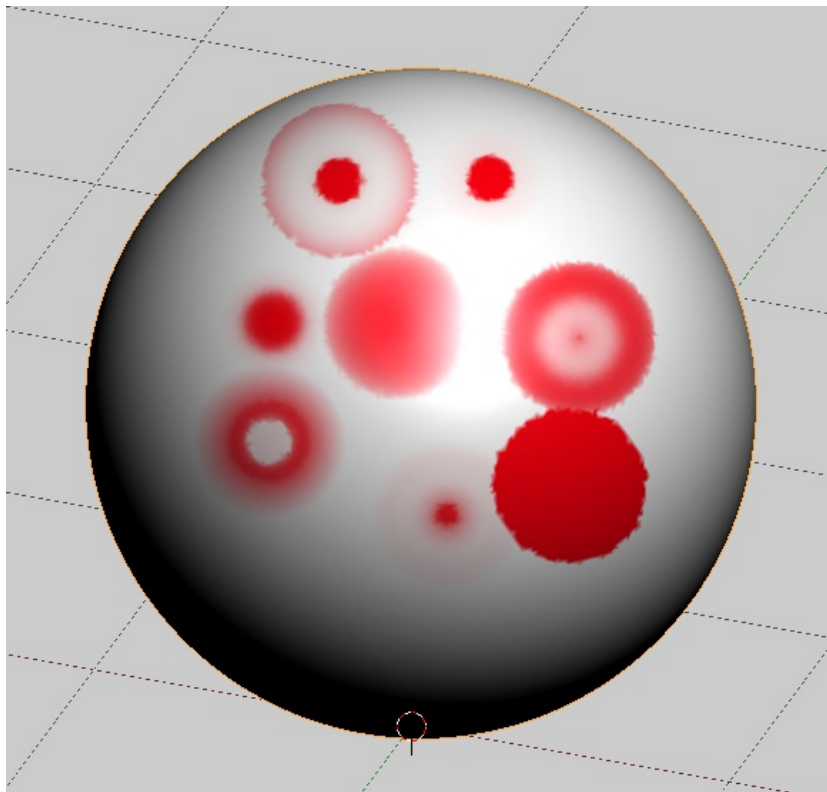
スムージングの量を設定します。

Space

ブラシストロークを点を連ねて作成します。間隔は Spacing の設定によって決定されます。Spacing は、ブラシの直径の割合で表現します。

Curves

ブラシのカーブは、色の適用の強さに、ブラシの中心からの距離に応じて影響を与えます。言い換えれば、ブラシ強度の減衰をあなたが編集することができます。



Appearance

ブラシの範囲の輪郭の色をカスタマイズするだけでなく、カスタムアイコンを指定することができます。

Options

All Faces

ブラシの範囲内のすべての面を描画します。オフにすると、ブラシの中心の面だけが着色されます。

Normals

塗装前に頂点の法線を適用します。これは通常、ペイントに影響を与えません。

Spray

バグがあります! マウスが保持されている限り、ペイントを続けます。

Unified Settings

Size

すべてのブラシが同じサイズを使用します。

Strength

すべてのブラシが同じ強度を使用します。

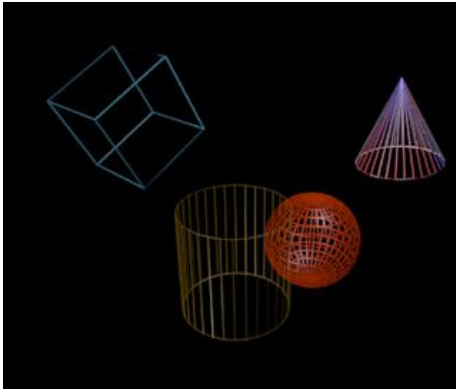
Wire Render

(ワイヤーレンダラー)



Wire Render

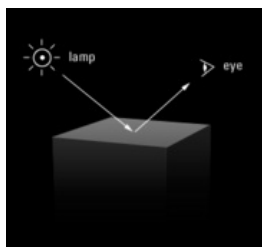
マテリアルタブにある Wire オプションを使うと、オブジェクトの辺をレンダリングできます。メッシュを構成する辺がすべて、単一ピクセルの画像としてレンダリングされます。色やアルファ、その他線に関係したプロパティは Surface Render と同じパネルで選べます。



Wire Render

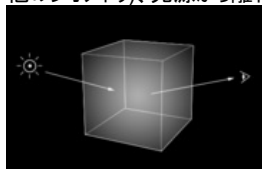
Volume Rendering

Volume rendering は、光が3D領域内の関係する媒体を通過するところをレンダリングするための手法です。Blenderのsim-physicsプランチの実装は物理ベースのモデルで、ボリュームでの光の様々な相互作用を比較的リアルに表現します。



Solid rendering

固体表面のレンダリングプロセスには、ジオメトリの一部をとらえるカメラが関わり、次にカメラに向かって跳ね返る光を計算を (ランプオブジェクト、または他のジオメトリ)、光源から離れ、表面への向きへと行ないます。カメラに到着した光が、レンダリングの最終的な色となります。



Volume rendering

ボリュームのレンダリングは、動作が独特です。煙、霧、雲などのような、小さな粒子で充填された、3次元領域の空間 (ボリュームとして定義されている) に、光が入ります。

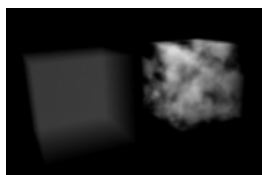
光は、様々な微粒子で周り跳ね返され、散乱されたり吸収されたりし、ボリュームを通り抜けた一部の光だけがカメラに到達します。そのボリュームが表示されるようにするには、マテリアルがどのくらい光を通すか、そしてボリューム内でどのように反応するか、ということレンダラが把握できる必要があります。ボリュームオブジェクトは3次元領域の空間を内包するもの、立方体などで例えば水をつまった密閉メッシュでなければならず、平面のような平らな面だけでは認識してくれません。画像を取得するには、レンダラにその領域をステップ実行させ、光がどのくらい吸収または散乱されるかを確認するために、'物質'(密度)がどの程度あるのかを確認する必要があります。これは時間がかかることもあるプロセスで、それは空間内の多くの点をチェックし、それぞれに密度を評価する必要があるからです。

Texturing Volumes

ボリュームへの色情報を追加するには、[Voxel Data](#) テクスチャを使用しましょう。

オプション

密度



Constant density vs textured density

光がボリュームを通過する際にはいろいろなことが発生し、カメラに到達する最終的な色に影響を与えます。これらは、実世界で起こる物理的相互作用を表現します。これらのほとんどはボリュームの密度に依存し、密度は一定にしたり、変化させたり、またはテクスチャによって制御することもできます。密度を制御することで、雲や濃い煙のような典型的な 'volumetric' 効果を得ることができます。

Density

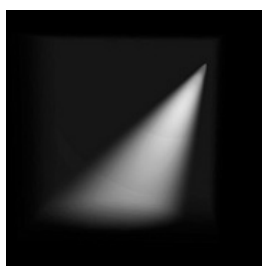
マテリアルの基本密度 - テクスチャからの他の密度はすぐ上に追加されます

Density Scale

見掛けの密度を増減するグローバル乗算器。スケールの異なるシーンを越えて一貫性のある結果を得たいときに、これが役立ちます。

シェーディング

散乱



Spot lamp scattering in a constant volume

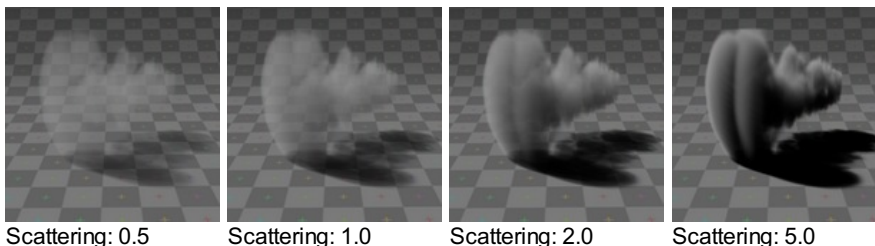
光が外部ソースからボリュームに入るとき、そのままストレートに通過することはありません。光がボリューム内の小さな粒子で偏向され、その一部の割

合の光がカメラに到達します。このプロパティをつかうと、ボリュームを通り抜ける光束が、眼のほうに散乱されるところが見えるようになります。

Scattering

ボリュームの外に散乱される光の量。ボリュームの外に散乱される光が多くなるほど、ボリュームの残りの部分を突き抜ける光は少なくなります。このパラメータを上げることで、ボリュームがより高密度に見なされるようになり、光がボリュームの表面の外に急速に散乱され、ボリューム内部の領域は暗く、光が到達しにくくなる、といった影響を持ちます。

以下の例では、ボリュームの外に散乱される光の量が少なくなるほど、光がボリュームを突き抜けて影のところまで届きやすくなることにご注目。



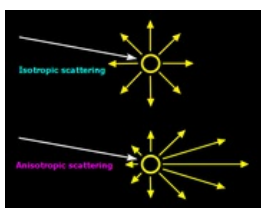
Scattering: 0.5

Scattering: 1.0

Scattering: 2.0

Scattering: 5.0

非対称



Isotropic and Anisotropic scattering

ボリューム内の光の散乱に用いるデフォルトのメソッドは、すべての方向に均等に光を逸らせるものです - いわゆる等方性散乱です。実生活の中でのさまざまな種類の媒体は、さまざまな角度の方向に光を散乱させることがあり、異方性散乱として知られています。後方散乱は、光が入射してくる方向に向かってより多く散乱されることを意味し、前方散乱は光の進行と同じ方向に沿って散乱されることを意味します。

Asymmetry

Asymmetry は、後方散乱 (-1.0) と前方散乱 (1.0) の範囲の間で制御します。デフォルト値の 0.0 では、等方散乱になります (すべての方向に均一)。

Transmission

トランスミッションは、ボリューム全体を透過する光の総称です。

この透過光は、さまざまな相互作用を起こすことがあり、例えば:

- ボリュームの外に反射/散乱された後の、入射光が残す結果
- ボリュームによって吸収された後、光が残す結果 (と熱に変換)

ここで、トランスミッションの色が使用されて、ボリュームを透過した後の光の最終結果の色が設定されます。

Transmission Color

ボリュームを透過した後の、光の最終的な色。

以下の例で注目してほしいところは、より多くの光がボリュームの外に散乱されるほど、透過できる光が少なくなってしまうことです。



Transmission color:
Yellow, Scattering:
0.5

Transmission color:
Yellow, Scattering:
1.0

Transmission color:
Yellow, Scattering:
2.0

Transmission color:
Yellow, Scattering:
5.0

Emission

いくつかのボリュームには、何も無いところから、化学的プロセスや熱的プロセスを介して光を発する、炎などのようなものがあります。この光は、ボリューム自体が発するもので、外部ソースから来る光とは独立するものです。

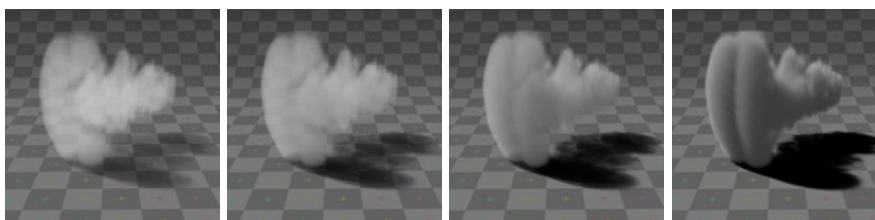
現在、この放出される光は、他のボリュームまたはサーフェスには影響しません (表面マテリアルの種類と同様に、'Emit' オプション)。

Emission Color

ボリュームによって放出される光の色。

Emission

スケールアップとスケールダウンのための、発光色の輝度乗数。



Emission 0.25,
Scattering: 0.5

Emission 0.25,
Scattering: 1.0

Emission 0.25,
Scattering: 2.0

Emission 0.25,
Scattering: 5.0

Reflection

'reflection'のパラメータは、ボリュームの外に散乱する光の、色づけやスケールに使用できます。これが影響を与えるのはランプと散乱から来る光だけで、透過光や発光の色に影響を与えることはありません。

これらの設定は物理的には不正確ですが、それはエネルギーを節約するためです - 散乱光は、ボリュームの残りの部分で透過される残りの光には影響を与えません。例えば、物理的な話では、光のオレンジ色の成分が、ボリュームからカメラに向かって散乱された場合、その反対の成分の色 (青) だけがボリュームを貫通し続け、ボリュームがマルチカラーを取る現象を引き起こして、使い難くなることがあります。ボリュームの色設定を率直に行えるようにして少し簡単になるように、反射パラメータを使用して、全体の色合いを素早く設定できるようになっています。

Reflection Color

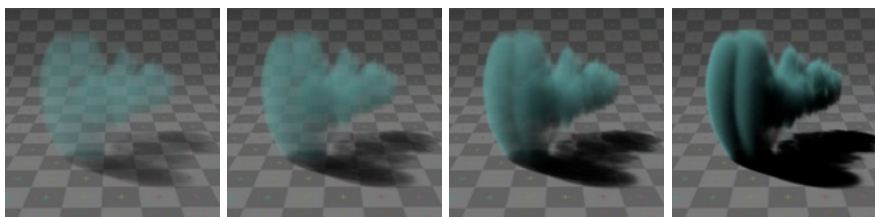
ボリュームの外に散乱される光の色。

Reflection

スケールアップとスケールダウンのための、反射の輝度の乗数。

ヒント

理想的には、反射のコントロールを使用する前に、他のボリュームの設定や照明をできるだけやり遂げておきましょう。物理的なものもらしさにこだわれば、マテリアルは正しく振る舞い、照明のシナリオの広い範囲でより予測しやすく使用しやすくなります。もちろん、ルールを破ることも、いつでもできます!

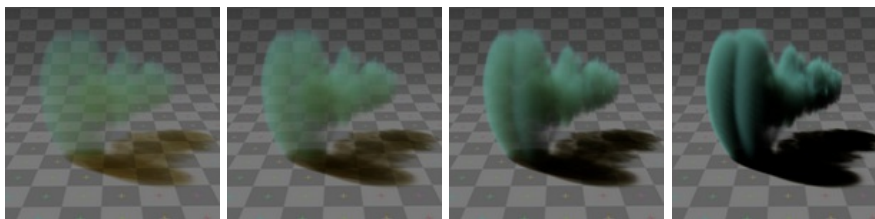


Reflection: Green,
Scattering: 0.5

Reflection: Green,
Scattering: 1.0

Reflection: Green,
Scattering: 2.0

Reflection: Green,
Scattering: 5.0



Reflection: Green,
Transmission: Yellow,
Scattering: 0.5

Reflection: Green,
Transmission: Yellow,
Scattering: 1.0

Reflection: Green,
Transmission: Yellow,
Scattering: 2.0

Reflection: Green,
Transmission: Yellow,
Scattering: 5.0

照明

複数のシェーディングモードが利用可能で、レンダリングの迅速さと物理的な正確さの間の範囲でオプションが用意されています。

Lighting Mode

Shadeless

Shadelessは、最もシンプルで、薄い、かすかな霧や蒸気に便利です。

Shadowed

Shadowedは同様のものですが、外部のオブジェクトへの影をともしません。

Shaded

Shadedは、ボリュームに一重の散乱法を使用して、セルフシェーディングにおいてボリュームを光が貫通するようにします。

Multiple Scattering

多重散乱の計算を可能にします。

Shaded+Multiple Scattering

シェーディングと多重散乱の機能を兼ね備えます。

Shaded Options:

External Shadows

ボリュームの外部ソースからの影を受けとります (一時的)。

Light Cache

ボクセルグリッドの陰影情報を事前に計算し、シェーディングをスピードアップしつつも、精度はやや下がります。

Resolution

ボクセルグリッドの解像度で、低解像度にするると速くなり、高解像度にするるとより多くのメモリを使用します。

Multiple Scattering Options:

Diffusion

拡散係数、ぼかし効果の強さ。

Spread

この距離に比例して、光が拡散される。

Intensity

複数の散乱光のエネルギーの乗数。

Transparency

Mask

背景をマスクします。

Z Transparency

透明な面にアルファバフアを使用します。

Raytrace

透明物の屈折のレンダリングにレイトレーシングを使用します。

Integration

Step Calculation Method

...

Randomized

...

Constant

...

Step Size

ボリュームの深さのサンプルの次との間の距離。Step Sizes で、ボリュームにどの程度ノイズが増えるかが決まります。高い値だとレンダリング時間は短くなり、高ノイズになります。

Depth Cutoff

透過光がこの輝度を下回った場合に、光線の進行を早期に停止します - より高い値にすると、精度を犠牲にして、ボリュームの密度計算をスピードアップさせます。

Options

Traceable

この材料でレイトレーシングを計算できるようにします。

Full Oversample

アンチエイリアシングのすべてのサンプルで、シェーディング/テクスチャをフルにレンダリングするよう、この材料に強制します。..

Use Mist

この材料でミストを使用します (worldの設定で)。

Light Group

この材料の照明を制限して、このグループのランプだけが当たるようにします。

Exclusive

材料がこのグループを排他的に使用します。ランプは、他のシーンの照明から遮断されます。

Examples

<これらは、Wikiのシェーディングイントロセクション全体のサンドボックス編集で、材料とテクスチャについてまとめてあり、ボリュームシェーディングを始める入り口になってくれます。メッシュオブジェクトの制限に注意してください。他のオブジェクトタイプのシェーディングは検証が必要です...>

シェーディングは、最終的なレンダリング出力においてオブジェクトが見えるようにする手順と規則です。Blenderには、メッシュオブジェクトに陰影を付ける、4つのメソッドがあります:

1. Surface
2. Volumetric
3. Halo
4. Wire

Surfaceシェーディングの方向性は、実体的なオブジェクト、固体 (ただし、おそらく柔軟な)表面を持つスキニングしたオブジェクトで、椅子、剣、または桃のようなものです。表面は、拡散、鏡面、ミラー、および透明性を有するものとして描き出されます。また、半透明の面でその中の光を散乱する、表面下散乱と呼ばれる現象を起こすこともあります。反射性を持つこともでき、クロム、ツヤのあるプラスチック、または金属のようにしたり、さらに部分的に透明にして、ガラスや液体のようにもできます。

Volumetric シェーディングは、オブジェクトを、雲、煙、霞、霧、魔法の呪文、蒸気のような、微細粒子が充満する領域のボリュームとして扱います。

光がボリュームに入ると、こういった粒子によって散乱され、その散乱光の一部が私たちの目/カメラに到達して見えるようになります。ボリュームは密度、xxxの用語で記述されます。粒子は均一に着色されますが、ボリューム内の密度が変化するので、その形状によっては暗い部分ができる場合があります。密度をボリューム全体に均一に分散させることも、あるいは凝集させて形を認識できるようにすることもできます。これらの微細粒子自体を光発させることもでき、赤熱する残り火やスパークを内包するようしたり、雲の内部に何らかのエナジーフィールドを媒介させたりもできます。その密度をパーティクルシステムによって吹き動かして、ジェットや排気を巧妙に作成することもできます。

Haloシェーディングは、オブジェクトの各頂点を光の塊へ変えるもので、こういったエフェクトは、火花、妖精の粉、きらめき、そして閃光で見られるもので、例えるなら、明るい日光の下でダイヤモンドです。Haloはまた、レンズフレアを大まかに近似することもできます。これは実際のカメラのレンズが太陽などの明るい光源を直接向くときに見られるものです。

Wireシェーディングは、オブジェクトの各辺を、ワイヤーケージ、またはネットのように、細い線としてレンダリングします。Wireレンダリングは非常に高速で、より複雑な表面の代理マテリアルとして、中間レンダリングにかかる時間を節約するために使用できます。

シェーディングには、主要な構成要素が2つあります: マテリアルとそのテクスチャです。貴方が見る色は、光と陰影の作用によるものなので、照明のセクションも同様に確認しておく必要があります。影をつけられるタイプのオブジェクトはBlenderには5つあります: メッシュ、カーブ、サーフェス、メタ、およびTextです。以下の表は、オブジェクトの種類ごとに、どのタイプのシェーディングが使用できるかを示しています。非メッシュオブジェクトのすべてのタイプは、そのタイプからメッシュへと変換することができるので、突き詰めて言えば、すべての種類のシェーディングがすべての種類のオブジェクトで使用可能だということ覚えておきましょう

		Shading available per Object type			
		Surface	Halo	Wire	Volumetric
Mesh	yes		full	yes	yes
Curve	if cyclic or extruded		no	no	
Surface	yes		no	yes	
Meta	yes		no	no	
Text	yes		no	no	

Options

Halo Materials

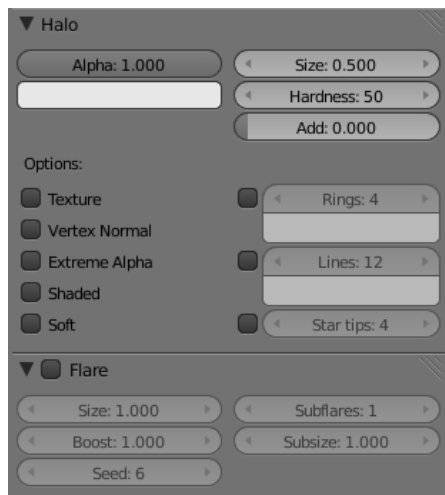
(ハローマテリアル)

Blender には面シェイダー (face-shader) の例に背き、各面ではなく各頂点に用いられるマテリアルがあります。光の輪に見えることからこのマテリアルを Halo (ハロー/ハイロー) と呼びますが、実体は持ちません。小さな光の集まりのようですが、ランプのようにシーンを照らすことはなく、本当の照明ではありません。

特殊効果を作りたいときやオブジェクトを発光させたいとき、または目に見える光や霧/大気を本当の光源の周りに作りたいとき、Halo は非常に便利です。

Options

(オプション)



Haloパネル

Halos を有効にするには、Material の最上部にある Halo ボタンを押します。

3Dビューを見てみると、メッシュの面がもう描画されていないことがわかります。代わりに頂点だけが描画されていますが、これが各 Halo の開始点になります。Halo は混みあったシーンでは見つけにくくなるので、[アウトライナー](#) で見つけやすいように、うまく名前をつけてください。

プロパティウィンドウの、通常 Diffuse、Specular、Shading といったパネルがある箇所には、Halo の特性に関連したパネルがあります:

Halo Panel

(Halo のパネル)

Alpha (アルファ)

透明度

Color Swatch (色)

Halo 自身の色

Size (サイズ)

Halo の大きさ

Hardness (硬さ)

Halo の硬さ。Specular の硬さと似たものです



Add (追加) の効果

Add (追加)

Add スライダーは Halo の色が後のオブジェクトや他の Halo の色に (混ぜ合わせるのではなく) 「加算」される量を決めます。Add 値を増やすほど Halo は、その後ろに移動するオブジェクトや Halo 領域を通過するオブジェクトを照らし出すように見えます。

Texture (テクスチャ)

Halo にテクスチャを与えます。

デフォルトではテクスチャはオブジェクト座標を使ってオブジェクトに用いられ、Halo の源になる頂点色の基準値にテクスチャ全体が影響を与え、Halo に反映されます。テクスチャを Halo の **範囲内** に影響させて色や透明度を変えたいとき、この機能を有効化してください。これを使うと、テクスチャ全体が Halo ごとにマップされます。パーティクルを使って写実的な雨の効果を作るような場合、またはそれに似た状況を作る場合に、この方法がとても便利だとわかります。

Vertex Normal (頂点の法線)

Halo の大きさを指定するのに使います

Extreme Alpha (アルファ強調)

アルファを増大させます

Shaded (陰影付き)

他のオブジェクトから光と影を受けるようにします

Shaded が有効なら、Halo はローカルの照明の影響を受けます。ランプは Halo をもっと明るくし、Diffuse Color (拡散色) や

Intensity (強度) に影響します

Soft (ソフト)

他のジオメトリとの交差する Halo の輪郭をやわらかくします。

Effects

(エフェクト)

エフェクトを好きなように組み合わせることができます。点やリングの数、色をエフェクトごとに決められます:

Rings (リング)

Halo を囲む 円形の輪を追加します

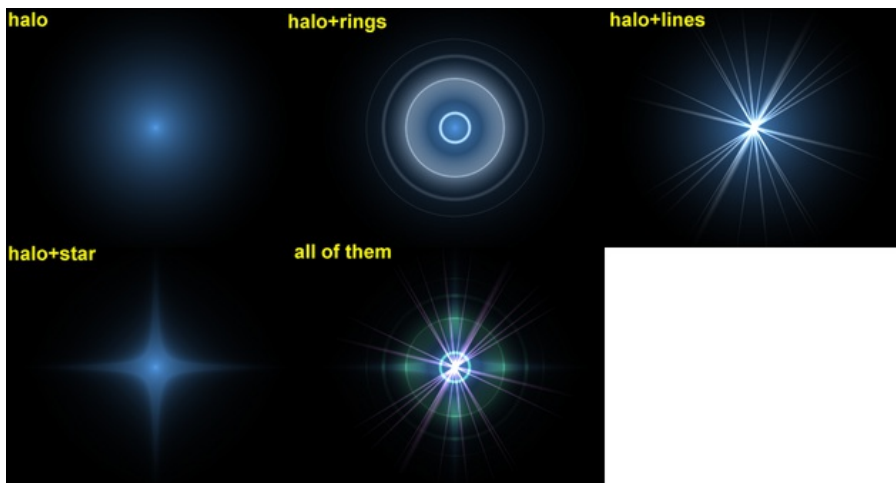
Lines (ライン)

Halo の中心から延びる線を追加します

Star tips (星の先端/ぎざぎざ星)

Halo を星形にします

Color Ramp は使えません。関連するボタンで各種特殊効果の切り替えができます。例えば Flare (レンズフレア)、Ring (リング)、Line (ライン)、Star (星)、Texture (テクスチャ)、Extreme Alpha (アルファ強調)、Shaded (陰影付き) があります。図「Halo のバリエーション」は、頂点がひとつだけのメッシュに Halo マテリアルを用いた例です。



Halo のバリエーション

Halo の大きさ、硬さ、アルファはスライダーで調整できます。これは、一般的なマテリアル設定とよく似ています。



Ring (リング)、Line (ライン)、Star tip (星の先端) はそれぞれ、チェックボックスで有効化すると数値ボタン Rings:, Lines:, Star: を使って数を指定できます。リングとラインはランダムな位置と方向に配置されます。ランダムのパターンを変えたいときは Seed: 数値ボタンを使ってシード値を設定します。

Flare Panel

(フレアパネル) 有効化すると Halo をレンズフレアとしてレンダリングします。

Size

Flare の大きさを Halo に対する比で指定します

Boost

Flare の強さをさらに増します

Seed

シードテーブル内のオフセット値

Subflares

サブフレアの数

Subsize

サブフレア、点、円の大きさ

Lens Flares

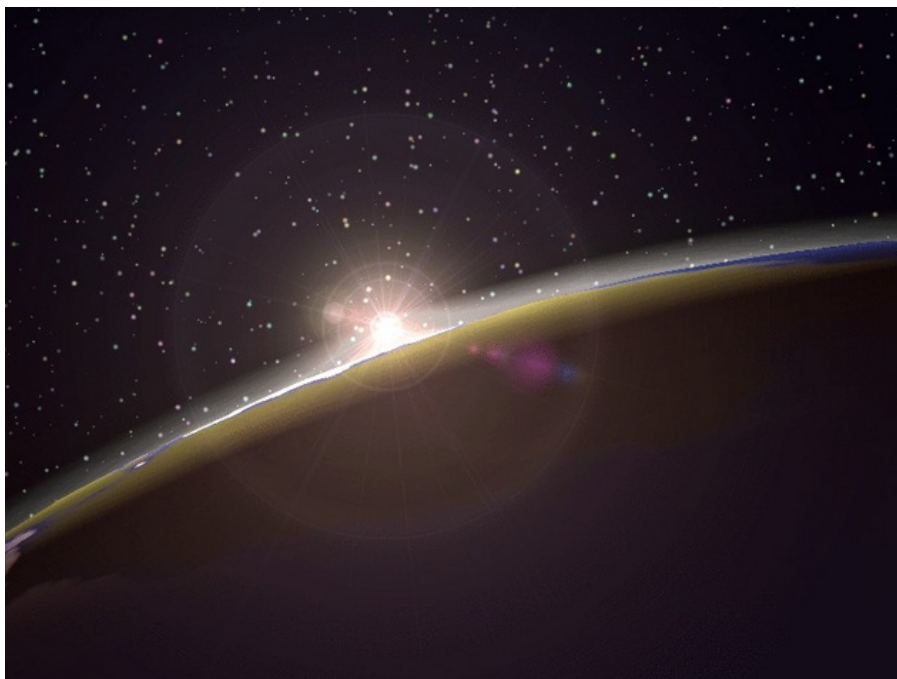
(レンズフレア)

我々の目は写真技術の機械的処理が原因の歪み (artifacts) が画像にあると、その画像はリアルだと信じるような訓練を受けてきました。Motion blur (モーションブラー)、Depth of Field (被写界深度)、および lens flares (レンズフレア) はこうした歪みの 3 つの例です。はじめの二つは *レンダリングの章* で取り上げます。最後の一つは、特殊な Halo を使って作り出すことができます。画像を見た人はシミュレートされたレンズフレアにこれがカメラで撮影されたものだと思わなくて、画像は本物だと考えます。

下の図のようにメッシュオブジェクトからレンズフレアを作ります。初めにプロパティウィンドウのマテリアルタブで Halo ボタンを選び、Flare オプションを有効にします。Rings と Lines を有効にしますが、これらの設定色はかなり薄いままにしてください。Flares: 値と Fl.seed: 設定を変えて、見た目に満足できるまで調整してください。効果を強めるために Boost: を動かす必要があるかもしれません。

なおこのツールはガラスレンズを通過する光子の物理的シミュレートを行いません。単に目を楽しませるものです。

Blender のレンズフレアは動いている時もきれいに見えますが、他のオブジェクトに Flare メッシュを遮られると消えます。



レンズフレア

例

Dotmatrix display

(電光掲示板)

電光掲示板を作るのに Halo マテリアルを使ってみましょう。

- まずは 32x16 の大きさのグリッドを追加します。次にカメラを追加し、掲示板がうまく見えるように調整します。
- 2D の画像編集プログラムを使って、黒地に赤字の文章をシンプルなボールドフォントで描きます (あなたがものぐさとかなら [お気を悪くしませんように。この響きが好きなだけです!]、下の画像をハードディスクに保存することもできます... (訳注: 右クリックで保存を選ぶと開けない画像が保存されてしまうので、一旦別のウィンドウに開いてから右クリックで保存します))。図「電光掲示板のテキスト」をご覧ください。画像の幅は 512 ピクセル、幅は 64 ピクセルで、両側に少し黒の余白があります。



電光掲示板のテキスト

- 電光掲示板用のマテリアルを追加し、種類を Halo にします。Halo の Size (サイズ) は 0.06 にし、レンダリングすると白い点の格子が見えるようになります。
- テキスチャを追加し、種類を Image or Movie (画像または動画) にします。画像を読み込んで再度レンダリングすると、赤く塗られた点が格子状に並ぶのが見えるでしょう (訳注: このときマテリアルのオプション「テキスト」にはチェックを入れないままです)。
- テキスチャタブの Mapping (マッピング) にある、Size (サイズ) の X 値をだいたい 0.5 にして再度レンダリングします。テキストが掲示板の中央に来るはずですが。
- 白点を取り除くために、マテリアルの色を暗い赤色にしてレンダリングします。赤い点だけが見えるように成りましたが、掲示板がまだ暗すぎます。編集モードに入り、全ての頂点を ◊ ShiftD で複製します (動かさないようにご注意ください!)。マテリアルの Add (追加) 値を使って明るさを調整し

ます。



電光掲示板

これでテクスチャタブの Mapping (マッピング) にある Offset (オフセット) の X 値を使えば、テクスチャが掲示板を動くアニメーションが作れます (もっと解像度の高い画像を使うこともできますが、Halo を縮小して調整しないと重なってしまうでしょう (図「電光掲示板」))。

マテリアル番号について

Halo マテリアルは、最初のマテリアル番号を使って適用されたときにだけ動作します。あとに続くマテリアルはレンダリングされません。

Introduction to Textures

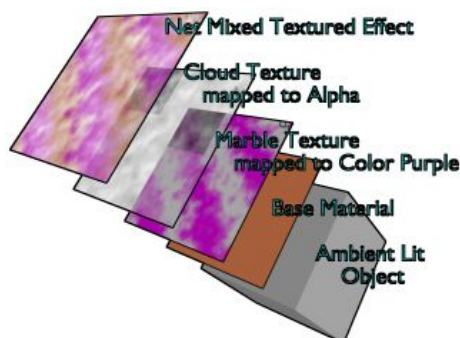
CGIにおいて、テクスチャマッピングは、表面に画像やパターンを投影することによって表面にディテールを追加する手法です。投影画像とパターンに影響を与えられる設定は、色だけでなく、スペキュラ、反射、透明性、さらには見せかけの3次元の奥行きなどもあります。ほとんどの場合、画像やパターンは、レンダリング時に投影されますが、テクスチャマッピングは、スカルプト、ペイント、オブジェクトの変形にも使用されます。

Blenderでは、Texturesは次のようになります:

- *Material*に適用される
- [World Background](#)に適用される
- *Brush*に適用される、例えば以下を参照してください:
 - [Sculpt Mode](#)
 - [Painting the Texture](#)
- Modifiersに関連付けられる、以下参照:
 - Particles textures
 - Ocean textures

Material Textures

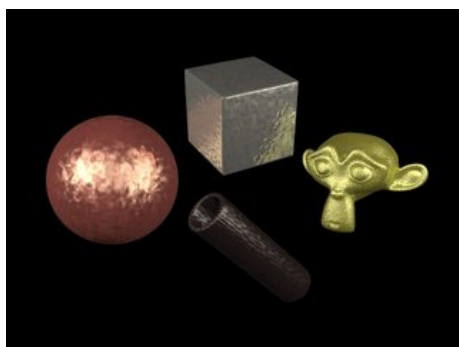
ここまでに見てきたマテリアルの設定は、滑らかで、均一なオブジェクトを生成するものでしたが、そのようなオブジェクトはリアリティに乏しく、そういった均一さは場違いになってしまったり周囲から浮いてしまう傾向があります。この非現実的な均一性に対処するために、Blenderでは、ユーザーが反射、スペキュラ、粗さ、などマテリアルさまざまな表面特性に修正を加えて、*textures*に適用することができます。



Textures Layer on base Material

テクスチャは、基材の上に追加される層のようなものです。テクスチャは、オブジェクトの最終カラーの要素の1つ以上に影響を与えます。あなたが見る正味の色は、この例の画像に示すように、レイヤーの重なるの効果を加味したものです。レイヤーは、使う場合、以下のとおりになります:

1. あなたのオブジェクトは、あなたのworldの設定に基づいて **環境** 光に照らされます。
2. **ベースマテリアル** は、表面全体を均一な色で塗り、光に反応して、拡散、鏡面といった、様々な陰影をつけ、そして光が透過したり物体に吸収される振る舞いに基づいて、色をミラーします。
3. **第一のテクスチャレイヤー**があり、ここでは紫色のマーブル色でオーバーレイするものです。
4. 次にある **第二の雲のテクスチャ**は、アルファ値に影響を与えることによって表面を霞/霧のような透明にします
5. これら2つのテクスチャは、ベースマテリアルと **混合** され、最終的な効果となります; 立方体は紫がかった茶色に曇ります。



Some Metal Textures

この観点をふまえて、**複数のテクスチャ**を使用して相乗効果を得ることは、リアルに見えるオブジェクトを作成する"秘訣"の一つです。あらゆる現実の物体を調べるときに注意深く"光を見ると"、その物体の最終的な外観は、さまざまな方法、さまざまな量の、異なる複数の基本的な視覚特性の組み合わせとして最もよく言い表せるものだというのが観察できるはずですが。これらの特性は、異なる角度で、異なる照明条件下で、そして遠ざかることで、より強く(または弱く)表現されます。Blenderでは、多くの方法でこれを達成することができます。 [this section](#)で説明するように、"テクスチャレイヤーのスタック"を使用することができ、また[here](#)で話すように、arbitrarily-complex networks ("noodles"...)を "texture nodes" として使用することもでき、選択はあなた次第です。

マテリアルのテクスチャは、3つの主要カテゴリに分類されます。:

Procedural Textures

数式によって生成されるテクスチャ。たとえば、木目、雲、そして歪んだノイズ

Images or Movies

オブジェクトに投影される写真や映画。たとえば、地球の平面地図を、球面にマッピングする。

Environment Maps

テクスチャを、反射や屈折の効果の作成に使用する。例えば、車のウィンドウに、街路のイメージを反射させる。

Data or Modifiers Textures

テクスチャを生データから取得したり、シーン内の特定のモディファイアによって取得する。

例:

- volumetric materials use Voxel Data textures, or Point Density textures
- テクスチャを、Oceanモディファイアから取得できます

note for editors

below we need to add more introductory text for all types of textures

World Textures

Todo

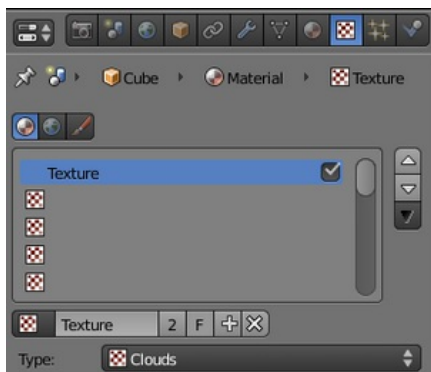
Brush Textures

Todo

Assigning a Texture

このページで示すのは、スロットにテクスチャを追加する方法です。テクスチャに共通するオプションについては [here](#) で説明されています。

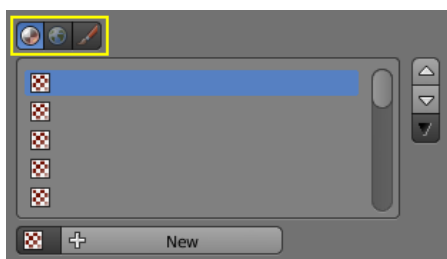
Choosing the Texture context



Texture panel

プロパティエディタで、テクスチャのコンテキストを選択しましょう: こうするとテクスチャーパネルが表示されます。

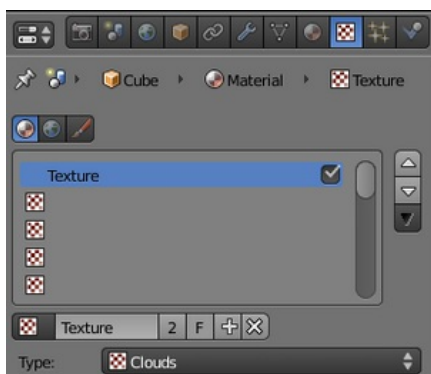
Choosing the Texture data type



Texture panel with buttons for Material, World, and Brush textures highlighted

テクスチャーパネルの上部にある3つのボタン Material, World, Brush が示すのはテクスチャデータの型、編集されるテクスチャの種類です。

Textures Slots



Texture panel

これらのボタンの下のリストは、我々が管理できるテクスチャの Stack を表します。これは、Texture Slotsを~18まで持つことができます:

- それを有効/無効にするには、テクスチャにチェックをつける、またはチェックを外す。
- 右側にある3つのボタンを使用すると、スタック内の個々のテクスチャを上下に移動させたり、またはスロット間でマテリアルの設定をコピーペーストできます。

Creating a new Texture Datablock in a new Texture Slot

空のスロットを選択し、 **+ New** ボタンをクリックしましょう。

こうすると、2つのことが行われます:

- 新しいテクスチャデータブロックを作成します。
- さらに、テクスチャスタック内に新しいスロットを追加します。

Creating a new Texture Datablock in a non-empty slot

空でないスロットを選択し、 **+** ボタンをクリックしましょう。

こうすると、2つのことが行われます:

- 新しいテクスチャデータブロックを、新しい名前で作成し、**選択しているスロットに割り当てられているテクスチャデータブロックのコピーを作成します**
- この新しいデータブロックを、選択しているスロットに割り当てます

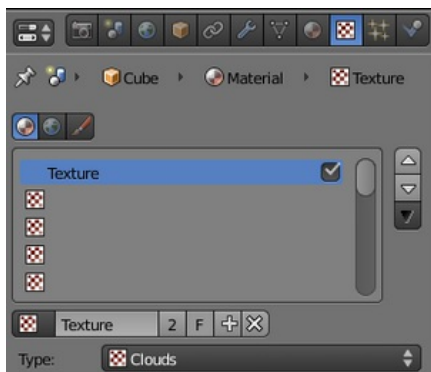
Sharing a Texture Datablock in a non-empty slot

- 空でないスロットを選択し、**Browse** ボタンをクリックしましょう。こうすると、メニューが開き、このファイル内の利用可能なテクスチャデータブロックがすべて表示されます。
- メニュー内のテクスチャデータブロックを選んで、選択状態のスロットにそれを割り当てましょう。これは、選択されたテクスチャを複数のオブジェクトで共有することになり、したがって、テクスチャデータブロックに示す *Number of users* が1ずつ増加します。

Textures common options

プロパティエディタで、テクスチャのコンテキストを選択しましょう: こうするとテクスチャーパネルが表示されます。

Textures Stack



Texture panel

下のリストのこれらのボタンは、我々が管理できるテクスチャの Stack を表しています。これは、Texture Slotsを 18個まで持つことができます:

- それを有効/無効にするには、テクスチャにチェックをつける、またはチェックを外す。
- 右側の3つのボタンを使用すると、スタック内の個々のテクスチャを上下に移動させたり、またはスロット間でマテリアルの設定をコピーペーストできます。

Texture Datablock

設定を確認するには、Textures Stackのスロットを選択しましょう。

スタックの下のボタンの最初のグループは、現時点でスタック内で選択されているテクスチャを表示します。

Browse

スタックの下の最初のボタンには、現在のファイルのtextureall利用可能なテクスチャが表示されます。テクスチャはグローバルに保存され、1つ以上のマテリアルにリンクさせることができます。再利用したいテクスチャをすでに作成している場合は、このリストから選択しましょう。

Name

マテリアルの名前を変更することができる名前欄。

Number of users

アクティブなテクスチャが別のマテリアルで使用されている場合、2つのボタンが現れ、これを使用してアクティブなテクスチャーのシングルユーザーコピーを作成することができます。既存のテクスチャに基づいて新しいテクスチャをすばやく作成するには、このボタンを使用しましょう。

Fake

F ボタンを押すと、アクティブなテクスチャに "Fake(仮の)" のマテリアルを割り当て、それが "本物の" ユーザーを持っていない場合でもテクスチャがファイルに保存されるようになります。

Add

新しいテクスチャでアクティブスロットのテクスチャを置き換えます。

Unlink

アクティブスロットからテクスチャを削除します。

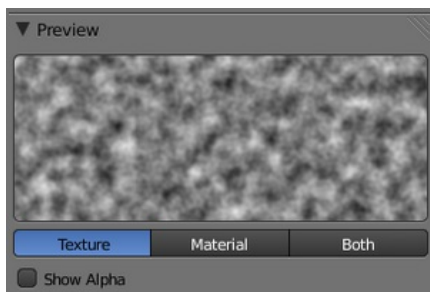
Texture Type

現在のテクスチャデータブロックに使用されるテクスチャの種類を選択します。

- [Procedural Textures](#)
- [Image](#) and [Video](#) Textures
- [Environment Map](#)
- [Volume Textures](#)
- Ocean Texture

これらのタイプは、[こちらセクション](#)で詳細に解説されています。

Preview



Preview panel

テクスチャプレビューパネルでは、テクスチャ自体がどのように見えるかのプレジュアライゼーションを、手早く、マッピングすることなしに行えます。

Texture, Material, or Both

テクスチャーのみ、マテリアルのみ、または両方を表示する

かを選択します。

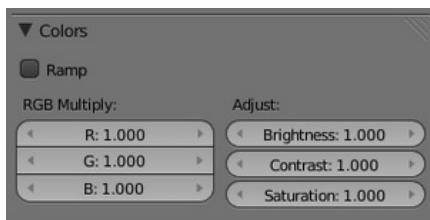
Show Alpha

プレビューで透過度を表示します。

Alpha: Use が [Image Sampling](#) パネルでチェックされている場合、画像のアルファチャンネルが表示されます。

Alpha: Useのチェックが外されている場合、アルファチャンネルは平均RGB値に基づいて表示され、[Influence](#) パネルで Alphaスライダーを使用するときのようになります。

Colors



Colors panel

Ramp ボタンは色勾配を起動して、テクスチャの色を新しいものに再マップすることができます。色勾配の使用方法については、[Ramps](#) を参照してください。

テクスチャの色は、Brightness, Contrast, Saturation のボタンで変更することができます。テクスチャすべての RGB-値 — Images と Environment Mapsを含む — は、RGB スライダーで変更することができます。

R, G, B

それぞれ赤、緑、青のチャンネルを明るくすることによって、テクスチャの色の濃淡を指定します。

Brightness

テクスチャの全体の明るさ/輝度を変更する

Contrast

テクスチャのコントラストを変更する

Saturation

テクスチャの彩度を変更する

Mapping

ここでは、テクスチャがオブジェクトにマッピングされる方法を制御できます。

Brushes

このオプションは、ブラシでは使用できません。認識できないからです

詳細については、[Mapping](#) のセクションを参照してください。

Influence

ここでは、テクスチャがどのプロパティに影響をするか、そしてどのくらい影響するかを制御できます。

[Influence](#) のセクションで詳しく説明されています。

Brushes

このオプションは、ブラシは使用できません。認識できないからです

UV Mapping a Mesh

(メッシュの UV マッピング)

最初のステップはメッシュの展開 (unwrap) です。メッシュが形状を表すのに必要な面をすべてそろえていると思えたら、展開を始めます。展開済みのモデルに対して面の追加や分割を行うと、blender は新たにできた面を追加してくれますが、マッピングや編集の追加作業が必要になるかもしれません。この方法で、ジオメトリの追加変更をガイドするために UV テクスチャ画像を使えます。

この章では UV のマッピング技法を扱います。次の章では [UV の編集](#) を、さらに次の章は [UV レイアウトの管理](#) 手法を、続いて [画像の適用](#) をそれぞれ扱います。

About UVs

(UV について)

UV マップにある点はすべて、メッシュの頂点に対応しています。UV マップの点を繋ぐ線はメッシュの辺に、UV マップの面はメッシュの面に対応しています。

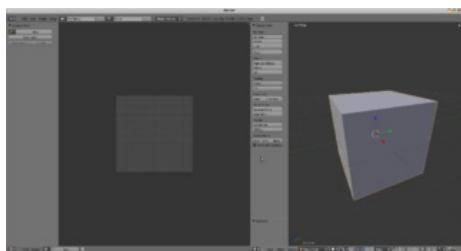
メッシュの各面はたくさんの UV テクスチャを持てます。UV テクスチャにはそれぞれ、別の画像が割り当てられます。UV/画像エディタで面を UV テクスチャに展開すると、メッシュの各面が自動的に 4つの UV 座標に割り当てられます。この座標は画像やテクスチャを面にマップする方法を定義します。これは2次元座標で、3次元の XYZ 座標と見分けやすいように U と V の名前で座標を呼んだのが名前の由来です。この座標はレンダリングやリアルタイムの OpenGL 表示にも使われます。

blender の面はそれぞれ別の画像にリンクすることができます。UV 座標はこの画像が面にどのようにマップされるのかを定義します。こうしてこの画像は、レンダリングやリアルタイム表示できるようになります。3D ウィンドウは、アクティブなメッシュオブジェクトの UV 座標を変更したり、画像を割り当てたりできるように、「面選択モード」にします。これで、面が複数の UV テクスチャに加わることができます。キャラクターの髪の毛の生え際にあたる面は、顔の UV テクスチャと頭皮/髪の毛の UV テクスチャの両方にあるかもしれません。

これについては次の章でもっと詳しく述べます。

Getting Started

(始めましょう)



UV Editing スクリーンレイアウト

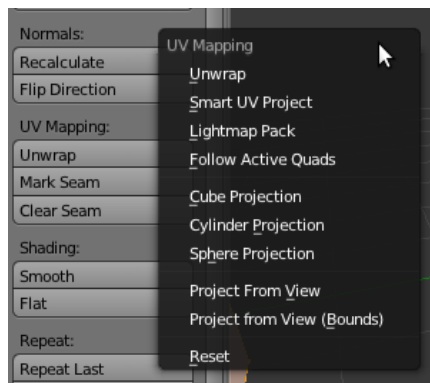
デフォルトでは、メッシュには UV がありません。始めに面をマップする必要があり、そうすれば [それを編集できます](#)。モデルの展開作業は 3D ビューウィンドウの編集モードで行います。この作業で、UV/画像エディタウィンドウに1つ以上の UV Island (島状のまとまり)ができます。

始めるには、画面最上部にある Info (情報) ウィンドウヘッダーの選択リストから UV Editing (UV編集) の [スクリーンレイアウト](#) を選んでください。これで UV/画像エディタウィンドウ (Shift+F10) の表示パネルと、3D ウィンドウ (Shift+F5) の描画域が並びます。

編集モードに入ってください。展開作業はすべて編集モードで行います。頂点、面、辺選択モードを使えます。

Workflow

(作業手順)



展開方式の選択

展開の手順はわかりやすいのですが、展開の結果に大きく影響するような選択肢が山ほどあります。選択肢の裏にある意味を理解すると、より効率的な展開ができるようになるでしょう。手順は次の通りです:

1. 必要なら seam (シーム、縫合線) をつけます
2. メッシュの構成要素をすべて選択

3. UV Unwrap メニューからUV マッピング方式を選びます
4. 展開の設定を調整します
5. テスト画像を追加して歪みが生じないか確かめます。[画像の適用 \(en\)](#) をご覧ください。
6. UV エディタでUVを調整します。[UV の編集](#) をご覧ください。

Mapping Types

(マッピングの種類)

blender には UV のマッピング方法が複数用意されています。シンプルな投影方式は、メッシュ表面のすべての点の位置を点/軸/平面方向に補間して、3次元空間を2次元空間に配置する換算式を使います。高度な方式ほど複雑なモデルに使うことができ、専門的な用途を持ちます。

基本:

[Cube\(キューブ投影\)](#)

メッシュを立方体の各面にマップしてから、立方体を展開図にします

[Sphere\(球状投影\)](#)

UV を球体に投影します。眼球や天体など、球か球に近い形にだけ役立ちます

[Cylinder\(円筒状投影\)](#)

UV を円筒状の面に投影します

[Project from View\(視点から投影\)](#)

3D ビューポートの現在のビューに見えているまま平らにします

高度:

[Unwrap\(展開\)](#)

生物の形状に役立ちます。seam に沿って切り分け、メッシュを平らにします

[Smart UV Project\(スマートUV投影\)](#)

メッシュを角度を基準にして island に分割します

[Lightmap Pack\(ライトマップパック\)](#)

面ごとに切り離して UV グリッドに詰めます

[Follow Active Quads\(アクティブ四角形面に追従\)](#)

アクティブな四角形から始めて、そこに環状に繋がる面に沿って uv をたどりませ

UV はリセットすることもできます。各面が UVグリッドを埋めるように配置され、どの面も同じマッピングになります。

タイル状に敷き詰めることのできる画像を使ったとすれば、メッシュ表面は、面の形にあわせて歪まされたこの画像の、滑らかな繰り返しで覆われるでしょう。この展開オプションは、マップをリセットし、すべての展開を取り消すときに使ってください(開始時の状態に戻ります)。

Basic Mapping

(基本のマッピング)

オブジェクトの基本となるジオメトリとその見え方に基いて、Mesh » UV Unwrap » Cube(キューブ)、Cylinder(円筒)、Sphere(球) の方式は最初うまくフィットしたとおりに面を広げようとしています。ここで、3D ウィンドウでの見え方は特に重要です。また、オブジェクトを包むように立方体のサイズや円筒の半径の設定(ツールシェルフ内)も調整しておくべきです(blender単位(BU)で)。

次の設定は キューブ、円筒状、球状の投影マッピングでよく使われます。

Correct Aspect(アスペクト比の補正)

画像のアスペクト比を踏まえて UV を配置します。画像がすでに正方形でないテクスチャ空間にマップされていれば、これを前提に正しく表示されるように配置をひずませ、投影します。

Clip to Bounds(境界にクリッピング)

0 から 1 の範囲外にある UV を一番近い UV 空間の境界に移動して、範囲内におさめます。

Scale to Bounds(境界に拡大縮小)

0 から 1 の範囲より UV マップが大きければ、マップ全体が入るように縮小します。

Cube

(キューブ投影)

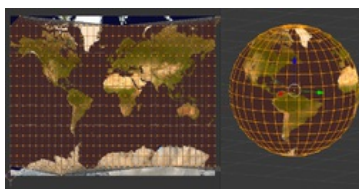
キューブマッピングはメッシュを6つのわかれた面に投影し、6つの UV Island を作ります。この島は UV エディタで重なって表示されますが、動かせませす。[UV の編集](#) をご覧ください。

Cube Size(立方体サイズ)

投影に使う立方体のサイズを設定します。

Cylinder and Sphere

(円筒状/球状投影)



球状投影でメルカトルの画像を使う

円筒状投影と球状投影は同じ設定を持ちます。相違点は円筒状投影が UV を円筒に対して投影するのに対し、球状投影は球の曲率を考慮し、緯度線が等間隔になります。

通常は円筒の展開を、縦に切込みを入れて平らに広げたかのように行うため、blender はその筒が直立していて、ビューが縦方向になることを求めます。ビューが変われば UV マップへの筒の投影も変わり、もし使っていれば画像を歪ませます。ただ、計算の行われる方向は手動で設定できます。球状投影でも同じ考え方ができます:

最初の地図製作者が世界地図を作った方法を思い出してください。blender でも、別の視点から球を展開すれば同じことを行えます。通常は球の展開のために、上下にある極から球を見ます。展開すると、blender が再現しているのはメルカトル投影図法です。あなたの側にある赤道上の点が、画像の中央に置かれるでしょう。この極視点 (polar view) は、非常に変わってはいるがよくある投影図を作ります。UV 画像としてメルカトル透視図法の世界地図を使うと、球上にマップされた素敵な惑星が作れます。

Direction(方向)

View on Poles(ビューを極に)

一番上(極)に視点を置き、視点からまっすぐ降りる軸を使います。

View on Equator(ビューを赤道に)

赤道を見る視点で、縦方向の軸を使います

Align to Object(オブジェクトで揃える)

軸の計算にオブジェクトのトランスフォーム(位置、回転、拡大縮小)を使います

Align(整列)

上になる軸を選びます

Polar ZX(極のZX)

極 (polar) 0 は x 軸上にあります

Polar ZY(極のZY)

極 (polar) 0 は y 軸上にあります

Radius(半径)

使用する円筒の半径です

Project From View

(視点から投影)

3D ウィンドウの Face->Unwrap UVs->Project from View(視点から投影/プロジェクション) オプションは、選択中の 3D ウィンドウに見えている通りに面をマップします。あなたに透視能力があれば、あるいはメッシュをつぶしてパンケーキのように平らにし、UV マップ上に置いたとすれば、こんな感じに見えるでしょう。モデリングしたオブジェクトの UV テクスチャに実物の写真を使うときは、このオプションを選んでください。モデルのあなたから遠い部分では、多少画像が引き伸ばされるでしょう。

Project from View (Bounds)(視点から投影(バウンド)) も同じ動作ですが、UV 空間の境界線に UV のサイズを合わせます。

Resetting UVs

(UV のリセット)

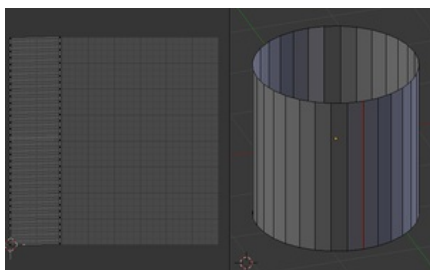
前に述べたように、3D ウィンドウの Face->Unwrap->Reset(リセット) は、選択面すべてを画像の同じ領域にマップします。オブジェクト(たとえば立方体)のすべての面を同じ画像にマップするには、立方体のすべての面を選んでからリセットメニューを使って展開します。

Advanced Mapping

(高度なマッピング)

Unwrapping Using Seams

(Seam を使った展開)



円柱の単純な seam

多くの場合、立方体、円筒形、球状、または best fit (訳注: おそらく「展開(Unwrap)」を指しています)の展開方式を使えば、良い UV レイアウトができます。しかしもっと複雑なメッシュ、とりわけ凹凸の激しいものには、上述のいずれかの展開処理の範囲制限と誘導用に、**seam** (シーム、縫合線) の定義が必要になるかもしれません。

縫い物と同じで、seam は画像/布地が縫い合わされる境界線です。展開処理を行ったメッシュは seam で切り開かれます。オレンジの皮をむいたり動物の皮を剥ぐようなものだと考えてみてください。皮に連続した切れ目を入れて、皮を剥ぎます。それから皮を多少引き延ばしながら平らにできます。この切れ目は seam と同じです。

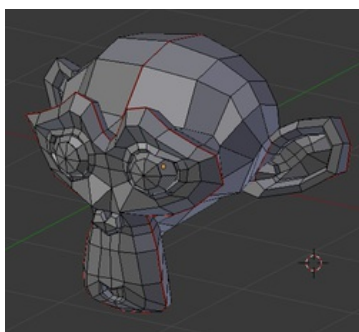
この方法を使うとき、引き伸ばしがどの程度あるか気を配る必要があります。seam が多いほど引き延ばしは少なくなりますが、これは大抵テクスチャ製作で問題になります。引き延ばしを最少にしつつ seam をできるだけ少なくするのが良い考えです。見えない部分に seam を隠すようにしましょう。3D ペイントを使う製作では、投影ペイントは seam を扱いやすいのでさしたる問題になりませんが、対照的に 2D テクスチャリングでは、別の UV island の境界線同士を合わせるのが難しくなります。

処理手順は次の通りです:

1. seam を作ります。編集モードで seam になる辺を選択し「シームをつける (Mark Seam)」を使います。
2. 展開 (unwrap) します。
3. seam を調整します。これを繰り返します。
4. 手動で UV を調整します。次章の UV 編集をご覧ください。

Marking Seams

(シームをつける)



seam のつけられた Suzanne

辺を seam に追加するには、単に辺を選択して Ctrl+E Mark Seam (シームをつける) を実行します。辺を seam から外すには、選択して Ctrl+E Clear Seam (シームのクリア) を実行します。

右の例では、seam を隠すために円柱の一番背面にある辺を seam に選び、デフォルトの展開方式を使いました。UV/Image Editor ウィンドウで、まるではさみで布地の縫い目を切って広げたように、すべての面がうまく展開されているのがわかります。

seam をつけるとき、面選択モードで Select (選択) -> Linked Faces (リンク) や Ctrl+L を使って作業状況を確認できます。このメニューは選択中の面に接続している面を seam を越えない範囲ですべて選択します。もし作りたい seam より外側の面が選択されていれば、seam がどこかで途絶えているのがわかります。ただし、seam が続くであろう部分を blender が解決できる限りは、seam が途絶えていても構いません。

動物の皮を剥ぐ方法が何通りもあるように、seam をつけるべき場所を決める方法もたくさんあります。普通は次のように考えてみてください。片手に (動物タイプの) オブジェクトを、片手にはさみを持っていて、オブジェクトをなるべく少ない回数切り開いてテーブルに広げようと考えています。体の前面と背面をわけたいので、耳の外側を通る seam をつけます。目は他とは別のメッシュなので、自動的にそれだけで展開されます。後頭部には縦に seam を入れて、頭の左右両側をそれぞれ平らに広げるようにします。

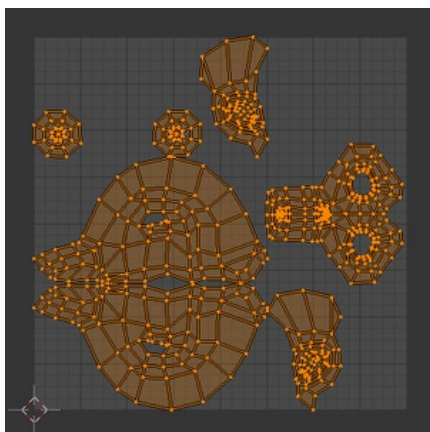
seam は、展開される面を制限する使い方もできます。例えば、頭のテクスチャリングをしているとき、頭頂と後頭部の頭皮は髪に覆われるため、実際にはテクスチャを必要としません。ですから、髪の生え際に沿って seam を定義します。こうすると、顔の正面を選択して Linked Faces (選択 -> リンク) を選択すれば、髪の生え際より先は選択されず、その部分の頭皮は展開されません。

頭や体のような左右対称のものを展開するときは、対称軸に沿って seam をつけます。例えば、正面図の真ん中で頭や体全体がわかるようにします。展開すると両半文を同じテクスチャ空間に重ね合わせることができ、右手の画像は左手の画像と共有され、顔の右側は左側と共有され、といった具合になります。

最後に、「どこでも何にでも完璧に動作する展開方法」を考えだす必要は「ない」ことを覚えておいてください。後述しますが、メッシュの部分部分に応じた別の方法を使うことで、簡単に複数の UV 展開を持つことができます。

Unwrap

(展開)



Suzanne の展開結果

展開すべき面を 3Dビューですべて選択することから始めましょう。面が選択できたら、次は展開です。3D ビューで Mesh(メッシュ) » UV Unwrap (UV 展開) または U を選択し、さらに Unwrap (展開) を選びます。

UV/画像エディタウィンドウからも、UVs(UV) » Unwrap (展開) または E を使って同じことができます。この方法は面すべてを展開し、それまでの作業をリセットします。

Face (面) » Unwrap (展開) » Unwrap (展開) オプションは、選択された面にあるすべての seam を踏まえ、面の接続の仕方を見てオブジェクトの各面が画像内にちょうどおさまるように展開します。可能なら、選択された面にはそれぞれ画像の別の領域が与えられ、どの面も他の面の「下に入り込む」ことがないようにします。もしオブジェクトのすべての面が選択されていれば、どの面も画像のどこかにマップされます。

blender は 展開処理の計算方式を2つ持ちます。3D ビューの ツールシェルフにある設定パネルで選べます。

Angle Based (アングルベース)

この方式はメッシュをうまく2次元で表現します。

Conformal (等角)

LSCM (Least Squared Conformal Mapping、最小二乗等角写像) を使います。通常はアングルベースに比べて正確性に劣る UV マッピングになりますが、オブジェクトが単純なほどうまく働きます。

Fill Holes (穴を埋める)

Fill Holes (穴を埋める) を使うと UV 領域内で重なりが発生を防ぎ、穴をうまく表現します。

Correct Aspect (アスペクト比の補正)

画像のアスペクト比を考慮した UV マッピングです。

Use Subsurf Modifier (細分割曲面モディファイアを使用)

subsurf モディファイア適用後の頂点位置を考慮した UV マッピングです。

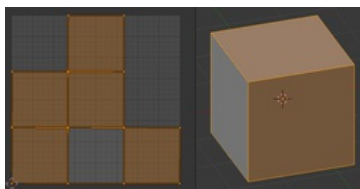
Margin (余白)

UV island の間隔

後々、マッピングを理解するために極めて重要になること: 面が使う UV 画像テクスチャは画像全体である必要はなく、一部で構いません。また、同じ画像の一部を複数の面で共有することができます。面は画像全体のごく一部にマップできます。

Smart UV Project

(スマートUV投影)



立方体にスマートUV投影を使う

スマートUV投影 (以前は Archimapper と呼ばれていました) を使うと、メッシュの角度変化に基づいた seam の自動生成の方法を、細かく制御できます。機械的なオブジェクトや建築物のような単純かつ入り組んだジオメトリ形状に適しています。

この機能はオブジェクトの形状、つまり選択されている面と面同士の関係を調べ、この情報と指定した条件に基づいて UV マップを作ります。

右の例ではスマートマッパーは、6面どれもが立方体の面と同じ四角形になるように、3面を上3面を下にした、きれいな並びで立方体をマップしています。

複雑度の高い機械的なオブジェクトに対して、このツールは非常に論理的でわかりやすい UV レイアウトを手早く簡単に作れます。

ツールシェルフの設定パネルでメッシュの展開方法を細かく調整できます:

Angle Limit (角度制限)

面のグループ分けを調節します: 制限値を上げるほどたくさんの小さなグループができ、歪みが減ります。制限値を下げるほど歪みの増加と引き換えにグループの数が減ります。

Island Margin(島の余白)

UV island の密度を決めます。数値が高いほど island の間隔が広がります。

Area Weight(エリアウェイト)

面ごとの投影ベクトルに、広いものほど重みづけします。

Lightmap

(ライトマップ)

ライトマップパック(Lightmap Pack)はメッシュの面や選択面を、1つずつ UV 境界線内に詰め込みます。ライトマップは初めゲーム開発で使われ、UV 空間を可能な限り多く利用するために、ライティング情報をテクスチャマップに焼きこむために使われていました。複数のメッシュに同時に作用させることもできます。ツールシェルフには複数のオプションがあります:

1つのメッシュで作業していれば、ツールのマップ対象を Selected Faces(選択面) だけにすることも All Faces(全ての面) にすることもできます。

Selected Mesh Object(選択したメッシュオブジェクト) オプションは複数のメッシュに対して使えます。使うには、オブジェクトモードで複数のメッシュオブジェクトを選択してから編集モードに入り、ツールを実行してください。

Share Tex Space(テクスチャ空間の共有)

複数のメッシュをマッピングするときに便利です。オブジェクトのすべての面を、重ならないように UV 領域内にフィットさせようとしています。

New UV Layer(新規UVレイヤー)

複数メッシュをマッピングするとき、このオプションはメッシュごとに新しい UV レイヤーを作ります。[レイアウトの管理](#) をご覧ください。

New Image(新規画像)

メッシュごとに新規画像を割り当てます。テクスチャ空間の共有 が有効なら全てのメッシュに1つだけ画像を割り当てます。

Image Size(画像サイズ)

新規画像のサイズを指定します。

Pack Quality(パック品質)

複雑な箱詰めの前に、事前詰め込みを行います。

Margin(余白)

UV island の密度を設定します。数値が高いほど island の間隔が広がります。

Follow Active Quads

(アクティブな四角形面に追従)

Face->Unwrap->Follow Active Quads(アクティブな四角形面に追従) は選択面から連続する面ループを(面の形が不揃いであっても)たどって、それらの面を配置します。なお、画像サイズは考慮されないで、画像領域にあわせて多少縮小する必要があるかもしれません。

Edge Length Mode(辺の長さモード):

Even(均一)

UV をすべて均等に置きます

Length(長さ)

各ループのUV辺の長さの平均スペース

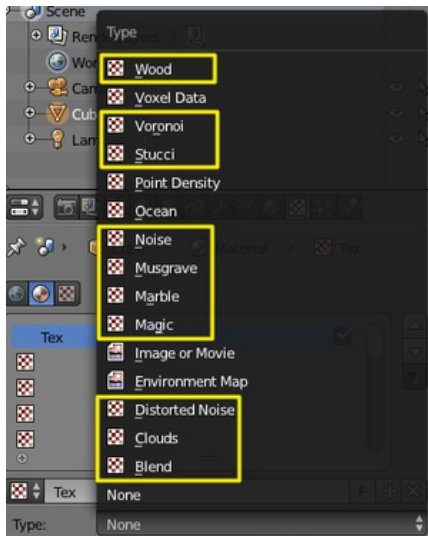
注意してほしいのは、出発点となるアクティブな四角形は UV 空間のもので、3D 空間のものではないということです。きれいな 90度に展開するには、「アクティブな四角形面に追従」を使う前にアクティブな四角形が UV 空間にあることを確認してください。

Texture types

(テクスチャの種類)

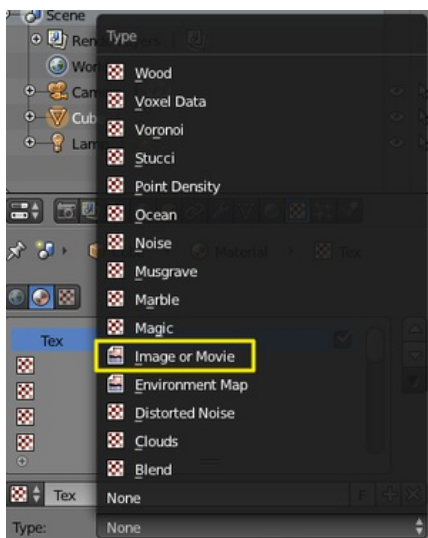
以下は利用可能なテクスチャの種類です:

- [Procedural\(プロシージャル\)テクスチャ](#)



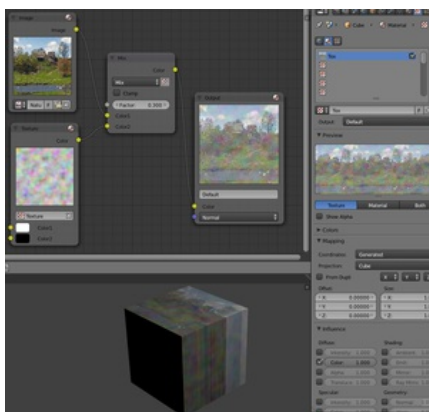
数式で生成されるテクスチャ。

- [Image\(画像\)テクスチャ](#)
- [Video\(ビデオ\)テクスチャ](#)



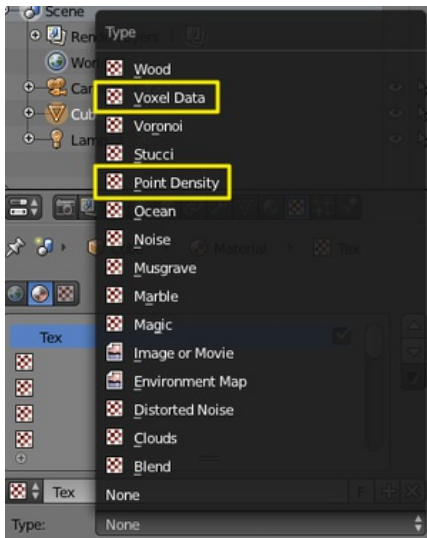
オブジェクトに投影される写真や映画。

- [Combined\(混合\)テクスチャ](#)



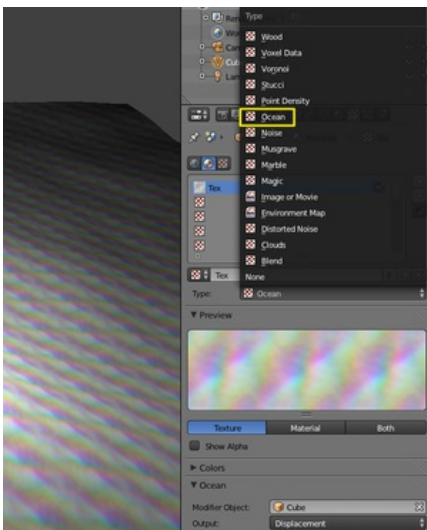
ノードにもとづいて混合されたテクスチャ。

- [Volume\(ボリューム\)テクスチャ](#)



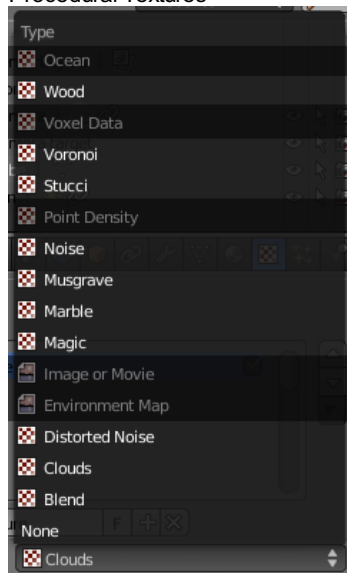
ボリュームデータに適用可能なテクスチャ。

- [Ocean\(海洋\)テクスチャ](#)



[Ocean\(海洋\)](#) モディファイアで生成されたテクスチャ。

Procedural Textures



The Texture Type list in the Texture panel of the Texture Buttons. (Non procedural textures darkened out.)

プロシージャルテクスチャは、数学的に定義されるテクスチャです。使用するの概ね比較的簡単ですが、特別な方法でマップする必要がないからではありません - これはプロシージャルテクスチャをそれほど複雑にすることができないという意味ではありません。

この種のテクスチャは '本物の' 3Dです。その意味するところは、端でぴったり一緒に合わさり、カットされて見えるはずのところでもつながって見えるようにしてくれるということです。木のブロックが実際に2つにカットされたかようになります。プロシージャルテクスチャは、フィルタやアンチエイリアスはされません。これが問題になることはほとんどありません: ユーザは指定された周波数を許容範囲内に容易に維持することができます。

これらが、利用可能なタイプです:

- [Blend](#)
- [Clouds](#)
- [Distorted Noise](#)
- [Magic](#)
- [Marble](#)
- [Musgrave](#)
- [Noise](#)
- [Stucci](#)
- [Voronoi](#)
- [Wood](#)

共通のオプション

Noise Basis

それぞれのノイズベースのBlenderのテクスチャは (ポロノイとシンプルノイズを除いて) ノイズの下地Noise Basis の設定を持ち、テクスチャの生成に使用されるアルゴリズムをユーザが選択できます。このリストには、オリジナルのBlenderノイズアルゴリズムが含まれています。Noise Basis の設定 はプロシージャルテクスチャを非常に柔軟にします (特にMusgrave)。

Noise Basis は、テクスチャの構造的な外観を支配します (ギャラリーはBlender 2.44から):

Cellnoise

Voronoi Crackle

Voronoi F2-F1

Voronoi F4

Voronoi F3

Voronoi F2

Voronoi F1

Blender Original

Noise Basisはさらに2つの設定が可能で、これは比較的 Blender Originalに近いものです:

- Improved Perlin
- Original Perlin

Nabla

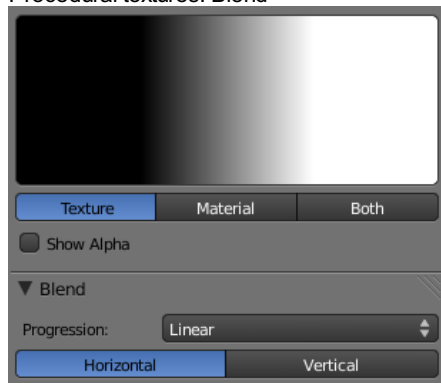
Blenderのほとんどすべてのプロシージャルテクスチャは、テクスチャマッピングの法線を計算するときにデリバティブを使用します (Blend と Magicは例外として)。これは Normal と Displacement Maps において重要です。効果の強さは Nabla Number Buttonボタンで制御されます。

ヒント

Mappingパネルのサイズボタンを使用して、プロシージャルテクスチャがマップされるサイズを設定しましょう。

プロシージャルテクスチャは、カラーテクスチャ、明度のみテクスチャ、アルファ値のついたテクスチャ、法線テクスチャ、のいずれも生成することができます。明度のみのもが使用されている場合、生成結果が白と黒のテクスチャになり、色勾配での使いやすさが大幅に向上します。一方、色勾配を使用していて明度値が必要になったら、Mappingパネルで No RGBに切り替える必要があります。

Procedural textures: Blend



Blend Texture Panels

Often used for

これは最も多用されるプロシージャルテクスチャの一つです。中間移行色のテクスチャを使用して、異なるテクスチャ同士をなめらかにつなぎ合わせたり (with Stencil)、または魅力的なエフェクトを作成したりできます (特に Mapping: Normal トリックを使用して)。これだけは覚えておきましょう: 色勾配を使用してカスタムブレンドを作成する場合、Mapping値に輝度を入力する必要があるなら、No RGBを使用しなければなりません。

Result(s)

明度。ブレンドテクスチャは、移り変わりを滑らかに補間して生成します。

Options

Progression

ブレンドのプロフィール

Linear

一次式的に移り変わる

Quadratic

二次式的に移り変わる

Easing

流れるように、非線形的に移り変わる

Diagonal

斜めに移り変わる

Spherical

三次元のボールの形で移り変わる

Quadratic Sphere

三次元のボールの形で、二次式的に移り変わる

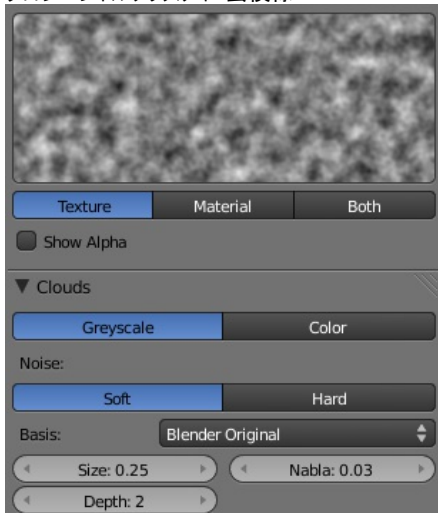
Radial

放射状に移り変わる

Horizontal/Vertical

進行方向を四分の一回転ひっくり返します。

プロシージャルテクスチャ: 雲模様



Clouds Texture Panels

よくある用途

雲、炎、煙。パンプマップとして使用して、マテリアルに全体的な不規則性を与えることに適しています。

Result(s)

グレースケール (デフォルト)、またはRGB カラー

オプション

Greyscale

標準的なノイズで、明度で表現する

Color

ノイズを、RGB値で表現する

Noise

Soft または Hard、コントラストとシャープネスを変更する

Size

ノイズテーブルの次元

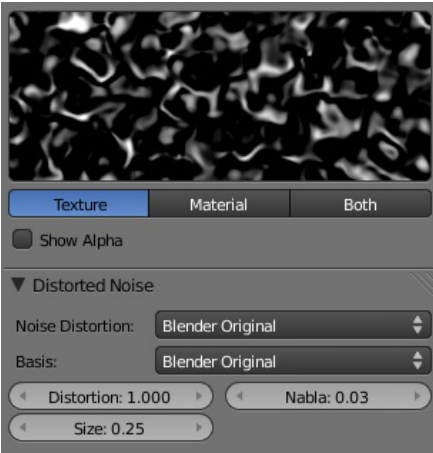
Depth

雲模様 の計算の深さ。数値を高くするほど、結果がでるまでの計算時間は長くなりますが、ディテールも細くなります

技術概要

擬似乱数の値を持つ3次元のテーブルを使用すると、そこからのなだらかさの補間値を使って各3次元座標を計算することができます (Ken Perlin氏と、SIGGRAPH 1985年の会議録からの、彼の見事な記事 "An Image Synthesizer"に感謝)。この計算手法は、Perlinノイズと呼ばれています。さらに、ノイズベースの各Blenderテクスチャ (ポロノイとシンプルなノイズを除く) は新しく "Noise Basis" 設定を持ち、テクスチャの生成にどのアルゴリズムを使用するかをユーザが選択できます。

プロシージャルテクスチャ: Distorted Noise



Distorted Noise Texture Panels

よくある用途

グランジ、非常に複雑で多彩

Result(s)

明度

オプション

Noise Distortion

別のテクスチャを歪ませるために使用するテクスチャ

Basis

ゆがめられるテクスチャ

Noise

発生するノイズの大きさ

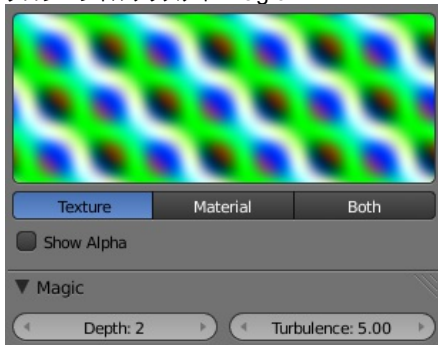
Distortion

Distortion Noise が Basis に影響を与える量

技術詳細

Distortion Noise は、Noise Basis とそこへのフィルタから選択するオプションをとり、ハイブリッドのパターンを生成します。

プロシージャルテクスチャ: Magic



Magic Texture Panels

よくある用途

あまり使用されません。"薄膜干渉"に使用することができ、その場合は Mapping を Reflection に設定し、比較的高い Turbulence で使用しましょう。

Result(s)

RGBカラー。RGB成分は、独立してサインの数式で生成されます。

オプション

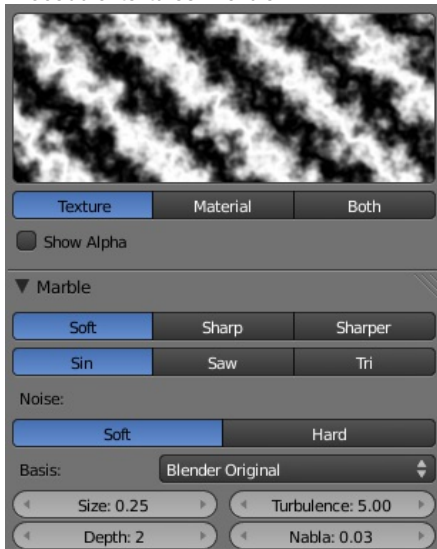
Depth

計算の深さ。数値を高くするほど、結果がでるまでの計算時間は長くなりますが、ディテールも細かくなります。

Turbulence

パターンの強さ。

Procedural textures: Marble



Marble Texture Panels

よくある用途

大理石、火、建造物でのノイズ

Result(s)

明度の値のみ

バンドは正弦波、表面波、または三角波のいずれかの数式に基づいて生成され、乱流のノイズがかかります。

オプション**Soft/Sharp/Sharper**

ソフトさにおいて、Marbleをより明確に定義する3つのプリセット

Sin/Saw/Tri

バンドを生成する波の形状

Soft/Hard

ノイズ関数を動作させる2つの方式。

Size

ノイズテーブルの寸法

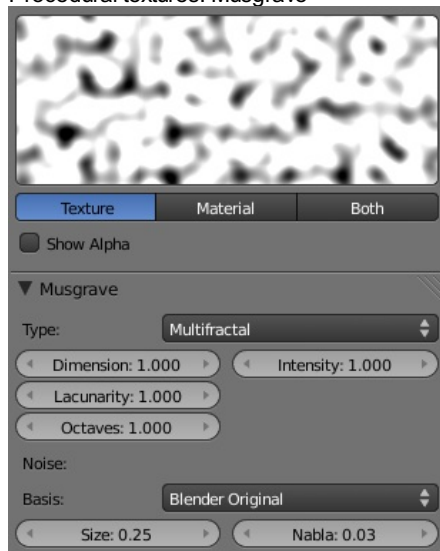
Depth

Marbleの計算の深さ。数値を高くするほど、結果がでるまでの計算時間は長くなりますが、ディテールも細かくなります。

Turbulence

正弦バンドの乱流。

Procedural textures: Musgrave



Musgrave Texture Panels

よくある用途

有機的な材料、ですが非常に汎用的です。ほとんど何でも行えます。

Result(s)

明度

オプション

Type

この procedural テクスチャにはノイズの種類が5つあり、それをベースにパターンを出力できます。そしてそれらはタブの上部にあるドロップダウンメニューから選択できます。5つの種類は、次のとおりです:

- Hetero Terrain
- fBm
- Hybrid Multifractal
- Ridged Multifractal
- Multifractal

これらのノイズの種類は、コントラストやスケールを変えた同じパターンをお互いの上に連続的にコピーするときの、Blenderのレイヤーの挙動を決定します。

主なノイズの種類は4つの特性をもっています:

Dimension

フラクタルな次元で、テクスチャ内のレイヤーのコントラストを、前のレイヤーと相対的に制御します。フラクタル次元が高くなると、各レイヤー間のコントラストが高くなり、したがってテクスチャもより詳細に表示されます。範囲: 0~2。

Lacunarity

Lacunarityは、マスグレブテクスチャの各レイヤーのスケールを制御します。各追加レイヤーが、ボタンに示した値の逆数のスケールを持つことを意味します。すなわち $Lacunarity = 2 \rightarrow Scale = 1/2 \text{ original}$ 。範囲: 0~6。

Octaves

Octavesは、フラクタル次元やlacunarity設定と併用して、オリジナルのノイズパターンをそれ自身の上にオーバーレイさせたりスケール/コントラストさせるときの回数を制御します。範囲: 0から8。

Intensity

光の強度。ヘテロ地形 のオフセットと呼ばれる。範囲: 0~10。

ハイブリッドマルチフラクタルとリッジフラクタルというタイプには、以下の追加設定があります:

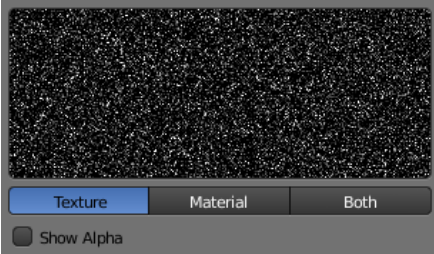
Offset

どちらにも "Fractal Offset" ボタンがあり、"海拔" での調整の役割を果たし、出力されるバンプマップのベースの高さを示します。このしきい値以下のバンプの値はゼロとして返されます。範囲: 0~6。

Gain

関数で作成される値の範囲を決定する設定。数値が高くなるほど、範囲は大きくなります。これは、通常は切り取られている端部のテクスチャの、追加的な詳細情報を高速に引き出す方法です。範囲: 0~6。

Procedural textures: Noise



Noise Texture Panel

Often used for

アニメーションによるホワイトノイズ。アニメーションにたくない場合には適しません。雲模様の代わりに、マテリアルに粗さをかけるとき向け。

Result(s)

明度

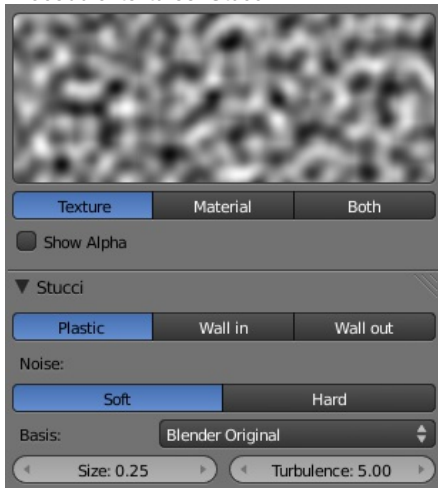
オプション

パネルもボタンもありません。切り替えてオンにするだけです。

技術詳細

よく出来ているように見えますが、これはPerlinノイズではありません! 実際の話、ランダムに生成されるノイズです。これは、画素ごとに、フレームごとに、毎回異なる結果になります。

Procedural textures: Stucci



Stucci Texture Panels

Stucci テクスチャは、ノイズ関数をベースにしています。

よくある用途

石、アスファルト、オレンジ。普段は、バンプマッピングにおいて粗い表面を生成します。

Result(s)

法線と明度

オプション

Plastic/Wall In/Wall out

Pプラスチックは標準のStucciで、一方 "walls" はStucciをつける場所の名前です。これは、穴やバンプのついた典型的な壁の構造です。

Soft/Hard

ノイズと併用できる2つの方式があります。

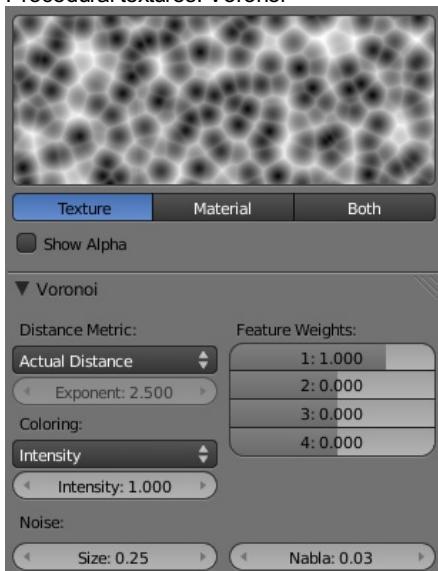
Size

ノイズテーブルの寸法

Turbulence

Stucci 計算の深さ

Procedural textures: Voronoi



Voronoi Texture Panels

よくある用途

非常に説得力のある金属、特に "ハンマー打ち出し模様" の効果。有機体のシェーダ (例えばうろこ、皮膚の静脈)。

Result(s)

明度(デフォルト)とカラー

オプション

Distance Metric

このプロシージャルテクスチャには、距離メトリックのオプションが7つあります。これらは、テクスチャのセル間の距離を見つけるアルゴリズムを決定します。これらのオプションは次のとおりです:

- Minkovsky
- Minkovsky 4
- Minkovsky 1/2
- Chebychev
- Manhattan
- Distance Squared
- Actual Distance

Minkovskyの設定にはユーザーが定義できる値があり (Exponent ボタン)、この値で距離関数の Minkovsky 指数 (ϵ) を決定します ($x^\epsilon + y^\epsilon + z^\epsilon$)^{1/ε}。値が1だと Manhattan 距離メトリックを生成し、1未満の値では星を生成し (**0.5** で Minkovsky 1/2になる)、そして高い値では正方形のセルを生成します (**4.0** で Minkovsky 4になり、**10.0** で Chebychevになる)。なので、ほぼすべての距離の設定は基本的に同じです - Minkowskyのバリエーションです。

Actual Distance/Distance Squared オプションを使用すると、不規則な形状の丸みを帯びたセルにすることができます。

Feature Weights

Voronoiパネルの下部にあるこれらの4つのスライダーは、4つのウォーリー定数の値を表しています。これは、テクスチャ内の各セル間の距離を、距離メトリックに基づいて計算するために使用されます。これらの値を調整することにより、興味深い効果を最終結果に持たせることができます。

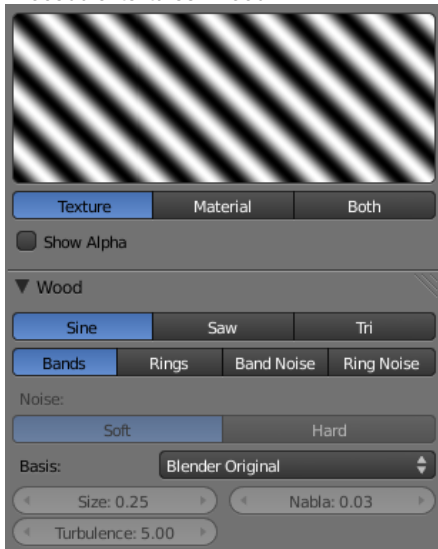
Coloring

4つの設定 (Intensity, Position, Position and Outline、および Position, Outline, and Intensity) があり、4つの異なるノイズ素地を、テクスチャ出力の色と明度の計算メソッドとして使用できます。このようにして作成したポロノイテクスチャは、"ウォーリースライダー" で姿をガラリと変え、他のテクスチャにあるノイズ素地の設定と同等になります。

技術詳細

For a more in depth description of the Worley algorithm, see: [Worley Documentation](#)(dead link).

Procedural textures: Wood



Wood Texture Panels

よくある用途

木目や環形のパターン。

Result(s)

明度のみ

オプション

Sin/Saw/Tri

バンドを生成する波の形状

Bands/Rings/Band Noise/Ring Noise

乱流の有無にかかわらず、ストレートまたはリング状のいずれかにバンドを設定します

Soft/Hard

ノイズ関数で利用できる2つの方式です。

Size

ノイズテーブルの寸法

Turbulence

バンドノイズタイプとリングノイズタイプの乱流

技術詳細

Generation

バンドはサインの公式に基づいて生成されます。また、ノイズの公式で乱流の程度を追加することができます。

Coordinates

バンドは正弦波の式に基づいているため、テクスチャは1.0単位ごとではなく、自身の π 単位ごとに繰り返します。これを補正するには、あなたが望む次元の π の値によってテクスチャを拡大縮小しましょう。

Image Textures

(画像テクスチャ)

用語「Image Texture (画像テクスチャ)」とは単に、RGBの各値(およびアルファ値)から成るピクセルを格子状に並べたもの(=グラフィック画像)を入力とするテクスチャです。他のタイプのテクスチャと同様、単に絵の「転写」を行うに留まらぬ用途がいくつもあります。

テクスチャ種類として Image or Movie (画像または動画) を選ぶと、パネル Image、Image Sampling、Image Mapping (画像/画像サンプリング/画像マッピング) が表示されます。この3つのパネルで画像テクスチャの使い方の大半を制御できます。

About Image Based Texturing

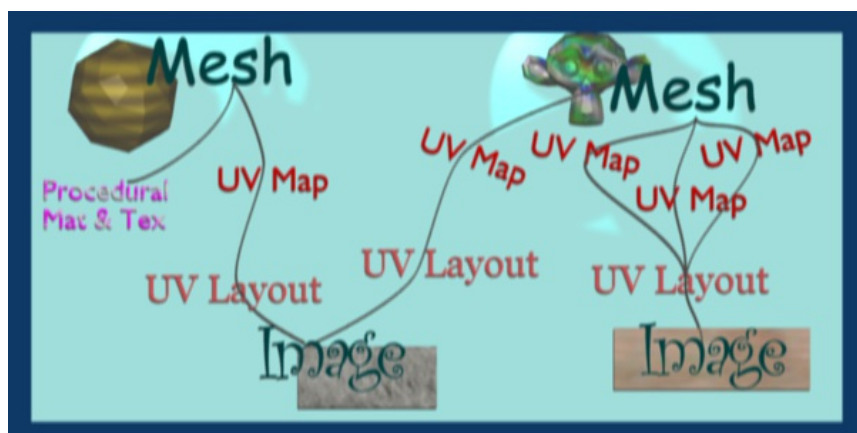
(画像を元にしたテクスチャ作成について)

テクスチャ画像は貴重なメモリ領域を消費します。高価なため非常に容量が小さい、高速な専用ビデオメモリに読み込まれます。したがって、画像は可能な限り小さくしてください。64x64 の画像は 128x128 の画像の 1/4 しか記憶域を消費しません。

アニメーション用の写実的なレンダリングでは、カメラの動きに合わせてズームが行われる可能性があるため、よく大きな画像テクスチャが使われます。通常は最終的なレンダリングのピクセル数に比例したサイズのテクスチャを使うことになるでしょう。結局のところ画像テクスチャとモデルを保持しつつ、レンダリング用の作業空間を提供するために、使える物理RAMの容量は限られています。

メモリを最大効率で利用するため、画像テクスチャは2の乗数で表せる寸法の正方形にすべきです。例えば 32x32、64x64、128x128、256x256、1024x1024、2048x2048、4096x4096 です。

メッシュ間で画像を再利用すると、必要なメモリを大幅に軽減できます。よく似たメッシュ領域を共通の画像を使うレイアウトに対応付けると、再利用可能です。下の概観図では左の画像が球とモンキーの一部で再利用されています。モンキーは2つのレイアウトを使い、1つは数枚の面用のUVマップを持ち、もう1つは3つのマップを持ちます。



UVテクスチャリングの部品すべてがどう連携しているか

ファイルのテクスチャを使うとき非常に重要なのは、メッシュの [UVマッピング](#) 作業が完了し、適切に配置されていることです。

UVマッピングはメッシュ全体に行う必要はありません。上図左の球には、UVマッピングされた面もあれば、生成式の(procedural)マテリアルやテクスチャを使う面もあります。写実的で高精細にしたいメッシュの部分にのみ、UVテクスチャを使ってください。例えば花瓶のモデルでは、装飾の施された縁の部分にのみ UVテクスチャが必要になるかもしれません。クッションの裏側に表と違う画像を用意する必要はありません。実際、クッションの背面が fabric(生成式のマテリアル)であることはよくあります。

別の例として、目は左右とも同じ画像に UVマップすべきです(片目だけを充血させたい場合を除く)。顔に関してはそばかすやえくぼの位置が左右対称でないため、両側を同じ画像にマッピングすべきでないことがあります。顔の片側だけ UVマップをわずかにずらすことももちろんできますが、すぐばれてしまうかもしれません。左右の耳は、似た面に画像や画像の一部をマップできるもう一つの例です。

Workflow

(作業手順)

次の段階を踏みます。

1. メッシュを作って、1つ以上の [UV レイアウト](#) に [Unwrap\(展開\)](#) します。
2. メッシュ用のマテリアルを1つ以上作ります。
3. UVレイアウトやテクスチャの特性用にそれぞれ1つ以上画像を作ります。次のいずれかの方法で行います。
 - 3Dウィンドウでテクスチャペイントを使い、メッシュに直接色を塗ります
 - UVエディタウィンドウに画像を読み込み(必要なら)編集します
 - 既存のマテリアルを UVエディタウィンドウ用の画像に Bake(焼き込み)します
4. できた画像がメッシュの1つ以上の特性に反映されるように UVテクスチャとして適用します。これはたくさんある Map To(対応付け)のオプションを1つ以上使って行います。例えば次のオプションです。
 - Color(色)に対応付けてメッシュの拡散色に作用させる
 - Normal(法線)に対応付けて法線の向きに作用させ、表面に凹凸やしわを与える
 - Specularity(スペキュラー/反射色)に対応付けて特定箇所につやを出し光らせる
5. 説得力のある結果を得るためテクスチャを重ね合わせます。

Using Images and Materials

(画像とマテリアルの利用)

テクスチャの色やアルファ(透明度)として画像を使うためには、UV/画像エディタで外部のペイントプログラムで作った画像をテクスチャとして開くか、新規画像を作ってテクスチャとして保存します。

画像を外部のペイントプログラムで作るなら、UVメニューにある Export UV Layout (UV配置をエクスポート) ツールで UV面の輪郭線を保存して使うことができます。詳しくは [画像の適用](#) ページの「UVレイアウトのエクスポート」をご覧ください。

Creating an Image Texture

(画像テクスチャの作成)

Blender で画像を作るには、まず単一色で塗りつぶされた [無地](#)の画像か、テストグリッドの画像を作る必要があります。その後、次の方法で画像に色を塗ることができます:

- 画像の土台として Vertex Color(頂点色)を塗る
- シーンでのメッシュの見た目に基いて画像を Bake する(焼き込む)

画像を作ったら、Blender 組み込みの [Texture Paint\(テクスチャペイント\)](#) や外部の画像ペイントプログラムで編集できます。

テクスチャが3Dビューでは見えるのにレンダリングされない

Textured 描画モードにすると、テクスチャを 3Dビューで見ることができます。これだけで Game Engine でもテクスチャを表示できます。ただし、レンダリングするときはマテリアルが必要です。UVテクスチャを使ってレンダリングするためには、メッシュに割り当てた Face Textures(面テクスチャ)マテリアルが必要です。マテリアルの設定パネルで選択オブジェクトにマテリアルを追加し、Face Textures を有効化します。

Examples

(例)

UVレイアウトがキャラクターの顔用に一つ、衣服用に一つあるとします。服にテクスチャをつけるには最低でも服の色用の画像と、できれば別に衣服の法線用の画像を作って見た目を模様のある布地にする「Bump Texture」(バンプテクスチャ)が必要です。服を着ると、例えばひじやひざなどで布地のつやや反射光の強さが変わります。別の画像を用意して、Specularity(反射光)の変え方を Blender に指示することができます。布地が折りたたまれたりしわになったりしている箇所では、Displacement(ディスプレイスメント)に対応付けた別の画像を使ってメッシュを物理的に変形させることができます。これらは、画像をテクスチャとしてメッシュに用いる例です。

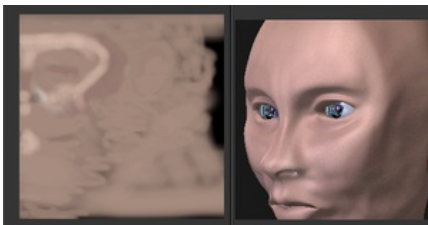
別の例として、顔はよく質問や学習のテーマになります。通常は基本的な肌の色を持つ、Shader に合った、Sub-Surface Scattering(サブサーフスキヤタリング)のマテリアルを作ります。それから次の用途を持つ UVテクスチャを追加し重ね合わせます:

- 色/Color と法線/Normal 用のそばかすマップ
- ディスプレイスメント/Displacement 用の皮下の静脈や腱のマップ
- 法線/Normal 用のしわや皮膚細胞の階層のマップ
- 色/Color 用の化粧のマップ
- スペキュラー/Specularity 用の つやマップ
- ゾンビ向けには、肉の削げ落ちた部分のアルファ透明度 (気持ち悪い...)
- 周辺光/Ambient Light が少ししか当たらないあごの下や鼻の穴
- やせた肌は透過が増えるので、そのためのマップ

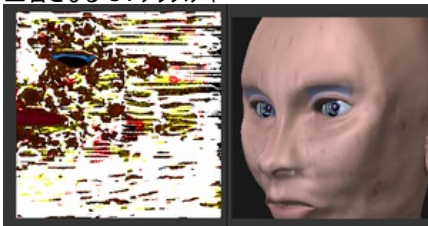
画像はそれぞれ、別のテクスチャチャンネルを使ってマップします。各マップは画像の別の特性(Color(色)、Normal(法線)、Specularity(反射光))に影響します。タイリングできる画像をテクスチャの Repeat(リピート)や Size(サイズ)を調整して繰り返せば、小さくて密度の高いパターンを作ることができます。

Layering UV Textures

(UVテクスチャの重ね合わせ)



土台となる UVテクスチャ



重ねあわせた UVテクスチャ

よくできたテクスチャは、画像を上下に重ねて作られています。まずは土台塗りが施された基本レイヤーから始めます。この上に重ねるレイヤーはすべて半透明にして下のレイヤーが透けて見えるようにし、細部を描き込みたい箇所だけ不透明にします。

大きな混乱を避けるため、1枚のメッシュ用の画像テクスチャはすべて同じ UVマップを使います。そうすれば、各画像をその下にあるものに整列させ、

右の例のように下から順に重ね合わせられます。このためにはまずこの章の記述に従い UV テクスチャ(マップ)を1枚作ります。次に Procedural Material(生成式のマテリアル)の章の記述に従ってマテリアル画像テクスチャを作ります。オリジナル座標ではなく、UV にマップしてください。

Material(マテリアル) » Texture(テクスチャ) » Mapping(マッピング) パネルで UV を選び、マップ名に同じマップ名を指定することを繰り返します。右の例では UVテクスチャを "Head" と名づけました。そうすれば、表示される画像テクスチャは UV座標を使ってマップされます。右の例「土台となる UV テクスチャ」では、顔は UVマップされたテクスチャを2つ持っています。1つは土台になる色用で、もう1つはしみ、傷、化粧用です。

どちらのテクスチャも Map Inputに 同じ UVテクスチャマップを使い、Color に作用させています。化粧のテクスチャは色のある部分以外は透明で、土台となる色のテクスチャが透けて見えています。色が強すぎたので2つ目のレイヤー(傷のレイヤー)の Color の作用値を 60% まで落としました。

画像テクスチャは通常、メッシュの色に作用するものと考えられています。現実的で写実的なレンダリングでは、光とメッシュ表面との作用の仕方を様々に組み合わせます。画像テクスチャは色だけでなく、法線(またはバンプ)、反射、あるいは Influence(作用)パネルで指定されたその他任意の属性に対応付けることができます。

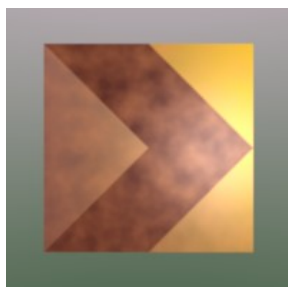
肌のつやがあって輝く部分を白で、そうでない部分を黒で塗ったグレースケール画像を描いたら、この画像を色ではなく Specularity(反射光)に作用させてUVマッピングするとよいでしょう。

メッシュの一部を透明にしてその下の別のメッシュを透かすには、透明にしたい部分を黒で塗ったグレースケール画像を作って UV の入力にし、(色ではなく)Alpha(アルファ)に作用させます。メッシュの一部を熱した金属片のように発光させたいときは、グレースケール画像を Emit(発光)に対応付けます。

驚くべきことにこれはただの氷山の一角です！ここで説明されたあらゆることでまだ足りない場合は、最近の Blender で導入された *Texture Nodes*(テクスチャノード)機能を使ってください。想像し得る限りのあらゆる方法でテクスチャを重ね合わせたり混ぜ合わせたりできます。

Mix and Match Materials

(マテリアルの混合と調和)



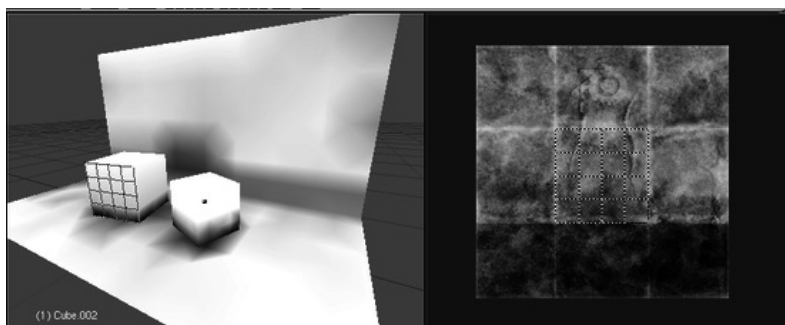
同じメッシュに別のプロシージャルマテリアルやテクスチャ、頂点ペイント、UVテクスチャを組み合わせることができます。

右の画像の World(ワールド)は、赤い ambient light/周辺光 を持っています。マテリアルは頂点ペイントと面テクスチャをどちらも有効にし、周辺光の半分を受ける設定です。色に作用する設定の弱めの Cloud/クラウドテクスチャで黄褐色を混ぜています。右側にある頂点は頂点ペイントで黄色に塗られており、左側は塗られていないプロシージャルの灰色です。UVテクスチャはパブリックドメインのテクスチャ CD にあった矢印画像です。シーンには白色の照明が右にずらして当てられています。この情報とユーザーマニュアルのここまでの内容を参照すれば、もうあなたも同じ画像を作れるはずです。

プロシージャルやテクスチャマップのどちらを使うのかに応じて、[複数のマテリアル](#) をメッシュに割り当てることもできます。プロシージャルにしたい面には、単に UVマッピングを行わないでください。

マテリアル設定で両方を有効にしていれば、UVテクスチャと頂点ペイント(3Dビューウィンドウで V)を同時に使えます。頂点色は UV画像テクスチャの明るさや色の調整に使います:

- UVテクスチャが一番底にあります (*Face Textures* (面テクスチャ))
- 頂点ペイントがその色に作用します
- プロシージャルテクスチャが一番上に置かれます
- エリア光源が表面に当たって影と影でない部分を作ります
- そして最終的に ambient light/周辺光 がそれを照らします

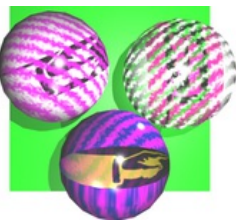


頂点色によるテクスチャ調整

1つの UVレイアウトが持てる画像は1つだけですが、画像のタイリングやアニメーションができます。レイアウトとは整理されたたくさんの UVマップであり、UVマップは多くのメッシュの面に対応づけられるので、したがって面は1枚の UVテクスチャ画像しか持たず、その面用の UV座標は画像全体にフィットする必要があります。1つの面に複数の画像を持たせたいときは面を複数の部品にわけて、それぞれに画像を割り当てます(またはノードを思い浮かべるかもしれませんが、それはまた別の話です...)

Using Alpha Transparency

(透明度の利用)



アルファ値を持つ UV テクスチャ

UV 画像のアルファ値 0.0 (透明) の領域は、黒でレンダリングされます。プロシージャルテクスチャと違って、UV テクスチャは土台となるプロシージャルマテリアルに働かないので、土台となるマテリアルは透明になりません。UV テクスチャは、その下にあるプロシージャルな色をすべて上書きします。プロシージャルテクスチャは UV テクスチャの上に適用されるので、プロシージャル画像テクスチャはどの UV テクスチャにも上書きします。アルファ値にマップされたプロシージャルテクスチャの透明 (黒色) の領域は、他の何よりも上に処理されて、その場所にあるオブジェクトを透明にします。UV テクスチャの可視部分を調整できるのは、頂点色だけです。右の例では、画像の袖口から指先までの指は透明で、UV テクスチャとして使われています。ボールの土台となるマテリアルは3つとも、青色の大理石のテクスチャです。面テクスチャが有効にされると、土台のマテリアル色は使われません。

左上のボールには一切頂点ペイントを使わず、指は中央の帯にマップされ、テクスチャはピンク色にマップされています。土台となるマテリアルでは頂点色ペイントと面テクスチャが有効にされていることがわかります。基本色の青色は使われず、テクスチャが使われています。頂点ペイントを使わなければ UV テクスチャの色を調整する方法はないので、指は白色で表示されます。UV 画像の透明領域は黒色で表示されます。

右上のボールでは、面の縦方向に、帯状にピンク色の頂点色を設定しています (3D ビューウィンドウで UV ペイントモードで面を選択し、頂点ペイントモードに切り替えてピンク色を選び、Paint (ペイント) » Set Vertex Colors (頂点色の設定)) を実行しました。指は面の中央の縦向きの帯にマップし、頂点色と面テクスチャを有効にしています。テクスチャはアルファの黒色にマップし、土台となるマテリアルのアルファ値 1.0 に乗算合成されています。テクスチャの白色部分は 1.0 で、1.0 × 1.0 は 1.0 なので (少なくとも私が最後に調べた時には)、この領域は不透明色で表示されます。プロシージャルテクスチャの黒色の領域、つまり 0.0 の部分は、基本マテリアルに乗算合成されて透明になります。マップを行っていない面 (ボールの左側と右側) では頂点ペイントが表示され (なし、灰色です)、ペイントされた部分はピンク色で表示されており、ペイントもマップもされている中央の縞は白色の UV テクスチャの部分がピンク色に変わっていることがわかります。プロシージャルテクスチャがオブジェクトを透明にさせた部分では、緑の背景が透けて見えています。UV テクスチャの透明領域は黒くレンダリングされました。

下のボールは複数のマテリアルを使っています。ボールのほとんど (中央の帯を除くすべての面) は土台のマテリアルで、面テクスチャは無効化しています (頂点色ペイントもされていません)。有効化しなければ、土台となる青色のマテリアル色が見え、ピンク色のテクスチャがその上に混ぜ合わせられます。中央の帯には頂点ペイントと面テクスチャを有効にした新しいマテリアル (2 Mat 2) が割り当てられています。面の中央の帯は頂点ペイントで黄色に塗られているので、指の白い部分は黄色になります。ピンク色のテクスチャが UV テクスチャに流れる部分では、ピンク色と黄色が混ざり合い、緑色になります。

二つの画像を混ぜ、同時に見えるようにするにはアルファ値を使う必要があります。土台となるマテリアルにアルファ設定のある画像テクスチャを使うと、その下にある UV テクスチャを透かして見せることができます。

複数の UV 画像を重ねる方法はいくつか考えられます:

- 同じマップを持つ複数の UV テクスチャを作ります。それから別々の画像 (アルファあり) を使うと Blender が自動的に重ね合わせます。
- [Composite Nodes \(コンポジットノード\)](#) の AlphaOver ノードを使って2枚の画像を合成して保存します。この画像を UV テクスチャとして開きます。
- 外部ペイントプログラムを使って画像をアルファ合成して保存します。この画像を面の UV テクスチャとして開きます。
- 2つのオブジェクトを、片方がもう一方の内側になるように作ります。内側のオブジェクトに土台の画像を割り当て、外側のオブジェクトに重ねたい画像のマテリアルを割り当てて、外側のマテリアルのアルファ値を 1.0 未満にします。
- [Material Nodes \(マテリアルノード\)](#) の AlphaOver か Mix ノードを使って2枚の画像を合成し、面のマテリアルとして使う3つ目のノードマテリアルを作ります。この方法では UV マップを行う必要がありません。複数のマテリアルを使って面にマテリアルを割り当てるだけです。

UV Textures vs. Procedural Textures

(UV テクスチャとプロシージャルテクスチャの比較)

マテリアルテクスチャのマッピング座標が UV で、Color にマップした画像テクスチャであるなら、UV テクスチャと同等になります。サイズやオフセットの変更が可能で、オブジェクトの色の影響度を Influence (作用) パネルで調整できるので、柔軟性が増しています。さらに、テクスチャチャンネルごとに別の画像を使うことができます。色、アルファ、法線、スペキュラー、反射光など、それぞれに別の画像を使えます。Clouds (クラウド) のようなプロシージャルテクスチャは、画像に本物らしさや詳細描写を与えるために、信じられないほど簡単に使え、役立ちます。

UV テクスチャ	プロシージャルテクスチャ
画像をメッシュの選択面の正確な座標に対応付け	パターンが動的に生成され、メッシュ全体 (またはマテリアルの覆う部分) に対応付け
画像を、メッシュのはっきり選択された面の範囲に一度対応付け	マテリアルが割り当てられたすべての面に一度対応付け。メッシュ全体であることも一部であることもあり
画像を面に一度だけマップ	Mapping の Size XYZ でテクスチャを面を越えて何回もタイリングできます。繰り返し回数はメッシュの大きさに依ります
オブジェクトの色とアルファに作用	それ以外に normals/法線 (バンプ)、reflectivity/反射、emit/放射、ディスプレイメントおよびその他多くのメッシュの見た目の特性にも作用。続くテクスチャを湾曲させたり型抜きしたりもできます
一つの面にたくさん割り当てられます	重ねることができ、テクスチャを10まで互いに重ね合わせて適用できます。複数チャンネルの混合用に、混合方式が多数用意されています
任意の画像タイプ (写真、ビデオ、レンダリング結果)。プリセットのテスト用グリッドを利用可能	多彩な表示: clouds/クラウド、wood grain/木目、marble/マーブル、noise/ノイズ、magic/マジック
動きのあるテクスチャ用の UV レイアウトを提供	Noise が動きのある唯一のプロシージャルテクスチャです

グラフィックスメモリをぐくわずかに消費 メモリを使わないか少ししか使いません。代わりに CPU の演算パワーを使います

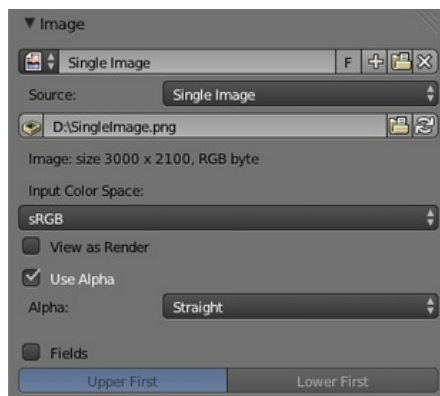
ある意味、1つのメッシュに1つの UV テクスチャを使ったほうがわかりやすくなりますが、メッシュの面に画像の細部描写を加えるというただ一つのことには専念するため、(UV 座標にマップされた) 複数のテクスチャを使うより制限が多くなります。プロシージャルテクスチャを UV レイアウトの定義する UV 座標に対応付けると、協調動作します。先述のように Mapping (マッピング) パネルの UV 座標マップシステムを使うと、複数の UV テクスチャを別の画像に対応付けられます。

Settings

(設定項目)

Image

(画像)



画像パネル

Image Sampling パネルでは入力ファイルを指定できます。

Image or Movie Datablock

Browse (閲覧)

.blend ファイルにリンクされたものから選択

Name field (名前欄)

画像の Blender 内での名前

F

画像テクスチャの Fake User の作成

+

現行テクスチャを新しいもので置き換え

Folder (開く)

コンピュータ内にある画像を閲覧

X

この画像のリンクを解除

Source (ソース)

使用する入力ファイルの場所と種類

Generated (生成)

Blender 内で生成された画像

Movie (動画)

動画ファイル

Image Sequence (連番画像)

連番の振られた複数の画像ファイル

Single Image (単一画像)

単独の画像ファイル

File for Image or Movie texture (画像か動画テクスチャ用のファイル)

[サポートされている画像形式](#) をご覧ください。

Pack image (梱包/パック)

画像を現在の .blend ファイル内に埋め込み

Path (パス)

ファイルパス

File Browser

クリックでファイルブラウザが開き、コンピュータ内にある別の画像を選択できます。⇧ Shift を押しながらかlickで現在の画像ファイルを開き、Ctrl を押しながらかlickでファイルのあるディレクトリを開きます

Reload (再更新)

ファイルの再読み込み。画像を外部アプリケーションで更新した際に役立ちます

Input Color Space (入力色空間)

ディスク上にある画像か動画の色空間

XYZ

XYZ 空間
 VD16
 gamma 2.2 sRGB 空間からの簡単な変換
 sRGB
 標準的な RGB 表示空間
 Raw
 Raw 空間
 Non-Color
 色でないデータを含む画像(法線マップ等)用の色空間
 Linear ACES
 ACES linear 空間
 Linear(リニア)
 709 (full range). Blender ネイティブでリニアな空間。

View as Render(ビューにも適用)
 この画像を画面上に表示するとき、レンダリングの表示変換を適用します。

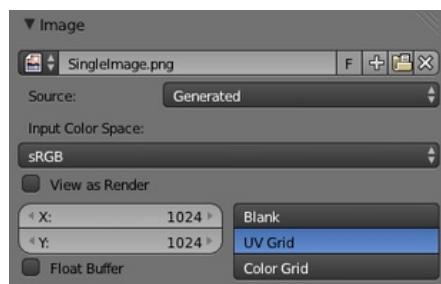
Use Alpha(アルファ使用)
 画像のアルファチャンネル情報を使うか、画像を完全に不透明にします

Straight
 透過した RGB とアルファピクセルは変更されません。
 Premultiplied(乗算済み)
 画像の透過した RGB ピクセルは画像のアルファ値に乗算されます。

Fields(フィールド)
 フィールド画像用です。映像のフレームは2つの別の画像(フィールド)が併合されてできています。このオプションは、レンダリング時に画像のフィールドがフィールドとして正しくレンダリングされることを保証するものです。

MIP Mapping cannot be combined with Fields.

Upper First
 ビデオフィールドの順序です。上位フィールドを先にします。
 Lower First
 ビデオフィールドの順序です。下位フィールドを先にします。



画像テクスチャの Generated(生成)ソース用
 Image(画像)パネル

Generated(生成)ソースには専用のオプションがあります:

XとYサイズ

生成画像の幅と高さ

Generated Image Type(生成画像タイプ)

生成する画像の種類

Blank(ブランク)

無地の画像を生成します。

UV Grid(UVグリッド)

UVマッピングのテスト用グリッドを生成します。

Color Grid(カラーグリッド)

改良されたUVマッピングのテスト用グリッドを生成します。

Float Buffer

浮動小数点のバッファを生成します。

動画と連番画像 ソース専用のオプションについては [こちら](#) をご覧ください。

Image Sampling

(画像のサンプリング)

Image Sampling(画像サンプリング)パネルでは、画像から情報を取得する方法を制御できます。

オプションの説明のために、いくつか例をお見せします。図「背景画像」は普通の JPEGファイル、図「前景画像」

は様々なアルファ値やグレイスケール値を含む PNG ファイルです。図「前景画像」の右側にある縦棒はアルファ合成され、横棒は 50% のアルファ値を持っています。



左: 背景画像
右: 前景画像

Alpha (アルファ)

透明度に関わるオプション

Use (使用)

透明度の情報を保存できる PNG と TGA ファイルに作用します (図「前景画像のアルファを使用」)。画像のアルファ値が 1.0 より小さい部分ではオブジェクトが部分的に透明になり、後ろにあるものが見えるようになります。

Calculate (計算)

画像の RGB 値を基準にしてアルファを計算します。黒 (0,0,0) は透明で、白 (1,1,1) は不透明です。画像テクスチャをマスクとして使う場合に、このオプションを有効にしてください。なおマスク画像で灰色の濃淡を使うと、幽霊や炎、煙/霧のような半透明にできます。

Invert (反転)

アルファ値を逆向きにします。マスク画像が、透明にしたい部分が白、したくない部分が黒になっていれば、オンにします



左: 前景画像のアルファを使用 (Use (使用) オプション)。各ピクセルのアルファ値を評価
右: 前景画像のアルファを計算 (Calculate (計算) オプション)

Flip X/Y Axis (X/Y軸を反転)

レンダリング時に反時計回りに画像を90度回転させます。

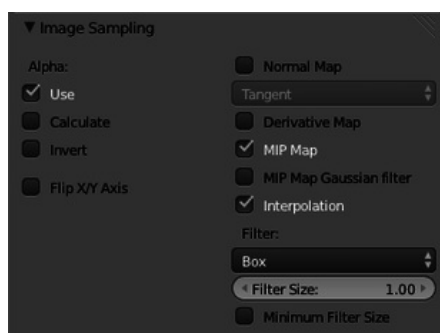


Image Sampling (画像サンプリング) パネル

Normal Map (法線マップ)

表面をでこぼこにする魔法をかけるのに使います。3つの RGBチャンネルそれぞれを使って、でこぼこの表面から偽の影を作る方法を決めます。入力画像を専用に用意する必要があります。[バンプマップと法線マップ](#)をご覧ください。

Normal Map Space (法線マップ空間)

Tangent (接線)
Object (オブジェクト)
World (ワールド)
Camera (カメラ)

Derivative Map (微分マップ)

赤と緑を微分値として使います。

MIP Map (ミップマップ)

ミップマップ は特定サイズ用に事前計算されたより小さなフィルターのかかったテクスチャです。前の絵の半分のサイズの絵を順に連ねた画像が生成されます。これはフィルター処理を最適化します。デフォルトではこのオプションは有効で、レンダリング速度を向上させます (ゲームエンジンで特に役立ちます)。このオプションをオフにすると、通常は画像がより鮮明になりますが、フィルター寸法が大きくなると (下記参照)、計算時間が著しく増えます。テクスチャが非常に小さくなると、MIP Map を使わないと、カメラ角度のわずかな変化で絵が変化するかもしれません。これはアニメーションで問題になるでしょう。

MIP Map Gaussian filter (ガウシアンフィルター)

MIP Map に繋げて使い、類似色に基いて MIP Map のサイズを小さくします。ゲームエンジンで使うテクスチャ、特に MIP Map テクスチャは、レンダリング速度とフレームレートを上げるため、なるべく小さくしたいはずで

Interpolation (補間)

画像のピクセルを補間するオプションです。これは画像を拡大するとわかります。デフォルトではオンです。オフにすると、正しくアンチエイリアスがされても、個々のピクセルの見分けがつくようになります。最後の機能は線やタイルなどの規則正しいパターンに使うと役立ちます。相当拡大しても形がくっきり残ります。10x10 ピクセルの画像 (左) の拡大前と拡大後を比べると、補間ありなしの差がはっきりわかります (図「補間なし/補間ありで拡大した画像テクスチャ」)。デジタル画像の crispness (クリスプ/エッジの明瞭さ) を保ちたいときはオフにしてください。



補間なし/補間ありで拡大した画像テクスチャ

Filter (フィルター)

レンダリングや、MipMap と Interpolation のオプションで使われるフィルターの大きさです。テクスチャを貼ったオブジェクトの、特に画像の透明部分の周囲に灰色の線や輪郭線が見えたら、この値を 1.0 から 0.1 付近まで下げてください。

Texture Filter Type (フィルター種類)

画像サンプリング用のテクスチャフィルターです。ちょうど *pixel* (ピクセル) が *picture element* (ピクチャーの要素) を表すように、*texel* (テクセル) は *texture element* (テクスチャの要素) を表します。テクスチャ (2次元テクスチャ空間) を 3Dモデル (3Dモデル空間) に対応付ける

ときは、複数 texel からのサンプリングに基づいてピクセル値を計算するのに、様々なアルゴリズムを使うことができます。

Box

モンテカルロ積分として知られる速くてわかりやすい最近隣内挿

EWA

Elliptical Weighted Average (楕円形加重平均) — 1980年台に Paul Heckbert と Ned Greene が開発した、最も効率のよい直接畳み込みアルゴリズムの一つです。EWA は楕円状の範囲内にある各テクセルを調査し、重みつけて、集計結果を重みの合計で割ります。

Eccentricity (偏心率)

偏心率の最大値です。値が高いほど、離れた/斜めの角度でのぼかしを少なくしますが、遅くなります。

FELINE

FELINE (Fast Elliptical Lines)は、テクスチャ空間内にあるライン上の数カ所で等方性の探索を行なって、レンダリング時間をそれほど増やさずにアンチエイリアスによるアーティファクト(画像のざざつき)を抑える異方性フィルターを作ります。

Probes (プローブ)

探索の数。1 から 256 までの整数です。

さらに学ぶためには

McCormack, J; Farkas, KI; Perry, R; Jouppe, NP (1999) [Simple and Table Feline: Fast Elliptical Lines for Anisotropic Texture Mapping](#), WRL

Area (エリア)

画像サンプリング用途のエリアフィルター

Filter Size (フィルターサイズ)

MIP Map と補間で使われるフィルターの大きさです

Minimum Filter Size (最小フィルターサイズ)

Filter Size の設定をピクセル単位の最小フィルター値として使います

Image Mapping

(イメージマッピング)

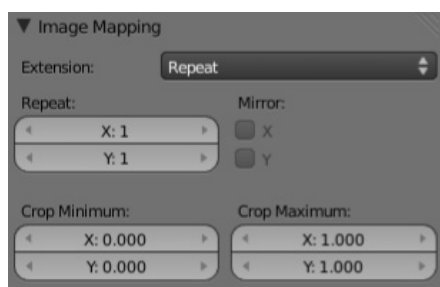


Image Mapping (イメージマッピング) パネル

Image Mapping パネルでは、3Dモデルへの画像のマップや投影の仕方を制御できます。

Extension (延長)

Extend (延長)

画像の端の色を、画像の外側に延長します

Clip (クリップ)

画像サイズにクリップし、外にあるピクセルを透明にします。画像の外では、0.0 のアルファ値が返されます。小さなロゴを大きなオブジェクトに「貼り付ける」ような使い方ができます。

Clip Cube (立方体クリップ)

画像を囲む立方体にクリップし、外にあるピクセルを透明にします。Clip と同じですが、Z座標も計算されるようになります。画像を囲む立方体の外側では 0.0 のアルファ値が返されます。

Repeat (繰り返し)

画像を縦横両方向に繰り返します。

Repeat (繰り返し)

X/Y 方向の繰り返し回数

Mirror (ミラー)

X/Y軸を中心にミラー反転します。テクスチャを自動的に反転させて使えます。

Checker (チェッカー)

即席のチェッカー模様です。Mapping パネルの Size も使うと、任意の数の格子を作れます。

Even (偶数)/Odd (奇数)

偶数/奇数のタイルを設定

Distance (距離)

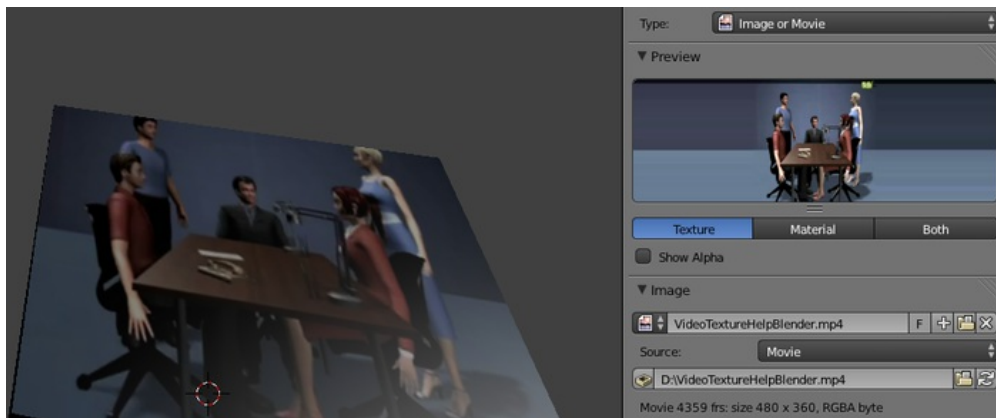
テクスチャサイズ内でチェッカータイル間の距離を決めます。

Crop Minimum (切り抜き最小)/Crop Maximum (切り抜き最大)

テクスチャ空間に相対的なテクスチャのオフセットとサイズ。この空間の外にあるピクセルは無視されます。Crop (切り抜き)に使ったり、大きな画像の一部をテクスチャに使ったりできます。

Video Textures

(ビデオテクスチャ)



Video texture

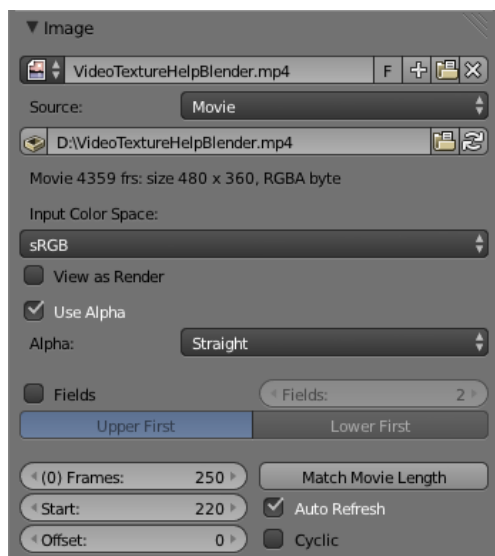
ビデオテクスチャは **画像** テクスチャのようなものですが、動画ファイルか、連番の振られた一連の画像をもとに作られます。これは、画像テクスチャと同じ方法で追加します。

Options

(オプション)

Image

(画像)



ビデオテクスチャ用の 画像 パネル

Source (ソース)

ビデオテクスチャ用に使用する入力ファイルの種類

Movie (動画)

[対応動画形式](#) をご覧ください。

Image Sequence (連番画像)

[対応画像形式](#) をご覧ください。

対応形式の連番画像ファイルを読み込むには、まず最初のフレームを選んで開きます。次にソースを Image Sequence (連番画像) に変え、このシーケンスの最後のフレーム番号を入力します。

ビデオテクスチャ用の入力ファイルの読み込みについての詳細は [こちらをご覧ください](#)。

Fields (フィールド)

フィールド画像を扱います。ふたつの別の画像 (フィールド) からなるビデオフレームがマージされます。Fields がレンダリングされたとき、画像のフィールドとレンダリングのフィールドが一致することを保証します。

Upper First (上優先 (偶数))

ビデオフィールドの順序 – 上位フィールドが先になります。

Lower First (下優先 (奇数))

ビデオフィールドの順序 – 下位フィールドが先になります。

Fields (フィールド)

レンダリングされるフレームごとのフィールド数。フィールドとインターレースされたビデオとともに使われ、各画像が奇数と偶数の両方を持つのか、あるいはただひとつだけを持つのかを示しています。

Frames (フレーム)

動画かシーケンス内のフレーム/画像のうち使用する数

Start (開始)

動画/シーケンスのグローバルな開始フレーム

Offset (オフセット)

アニメーションで使うフレーム番号のオフセット。動画/シーケンス内の、取り込みを開始するフレームの番号。

Match Movie Length (動画の長さと同じ)

画像の使用する長さを選択した動画/シーケンスの長さに合わせます。

Auto Refresh (自動更新)

フレームの変更時画像を自動更新する。

Cyclic (繰り返し)

ビデオが終わるとはじめに戻って再生をやりなおします。

ソースに動画を選んだ場合:

Use Alpha (アルファを使用)

画像からアルファチャンネルの情報を使うか、完全に不透明にするのか選びます

Straight (ストレート)

透けた RGB とアルファピクセルは変更されません。

Premultiplied (プリマルチプライ)

画像にある透けた RGB ピクセルは画像のアルファ値で乗算されます。

ビデオテキスト用の入力色空間についての詳細は [こちらをご覧ください。](#)

ビデオテキスト用のビデオサンプリングについての詳細は [こちらをご覧ください。](#)

ビデオテキスト用のビデオマッピングについての詳細は [こちらをご覧ください。](#)

Texture Nodes (テクスチャノード)

[テクスチャを積み重ねて](#)使う方法の替わりとして、Blenderにはノードによるテクスチャ生成システムがあります。これを使って色、パターン、あるいは他のテクスチャを [Material Nodes \(マテリアルノード\)](#) とほぼ同じ方法で組み合わせ、テクスチャを作ることができます。

このテクスチャは、普通のテクスチャを使える場所ならどこにでも使えます。テクスチャチャンネル、マテリアルノード、パーティクルシステム、あるいは他のテクスチャの中、などです。

ノードの回路網("noodles")には3種類の汎用的なノードが含まれています。**input**(入力)ノード、**filter**(フィルター)ノード(または**transformation**(変換)ノード)、**output**(出力)ノードです。こうしたノードをいくつでも回路網に組み込むことができ、任意の方法で繋ぐことができます。際限のない創造的で技巧的調整を行えます。

Note

ノードベースのテクスチャはリアルタイム表示では動作しません。レンダリングされた画像でのみ見ることができます。

テクスチャノードの利用

現在のテクスチャにテクスチャノードを使うには、[Node Editor ウィンドウ \(eng\)](#)を開き、ヘッダーにある "Texture" アイコンをクリックして Texture モードにします。

ノードを追加するには、最初にマテリアルを選択し、ノードエディタにある New ボタン(+)かテクスチャパネルにある New ボタンを使います。選択中のテクスチャは、普通のテクスチャとして機能させるかノードテクスチャとして機能させるかをノードエディタにある Use Nodes(ノードを使用)オプションで切り替え可能です。

ノードを使用するとデフォルトのノード構成が表示されます。"Default" という名前の Output ノードに繋がった赤と白のチェッカーボードのノードです。ノードテクスチャノード用の Output は、好きなだけ作れます(思い出されたかもしれませんが、他の種類のノード回路では1つの Output ノードに制限されています)。詳細は次章をご覧ください。

ノードツリーの追加、削除、操作方法の導入については [Node Editor のマニュアル \(eng\)](#) をご覧ください。

出力を増やす

テクスチャノードで定義したテクスチャに複数の出力を持たせて、それぞれを別の用途に使うことができます。例えば、テクスチャで diffuse (color) マップと normal マップの両方を定義する場合は:

1. テクスチャー一覧にテクスチャスロットを2つ作って、同じテクスチャデータブロックに設定します。
2. ノードツリーに Output ノードを追加して、Name テキストボックスに新しい名前を入力します。例えばひとつを "Diffuse"、もうひとつを "Normal" にします。
3. テクスチャパネルのテクスチャピッカーの下に、出力名の並ぶドロップダウンリストがあります。テクスチャリストの各項目ごとに、希望の出力をメニューから選んでください(例えば "Diffuse"、他を "Normal" にする、など)。

名前つき出力は、マテリアルをマテリアルノードで定義するときに使えます。この場合テクスチャチャンネルはおそらく使わないでしょう。代わりに、Texture ノードを Add » Input » Texture でマテリアルノードツリーに追加します。次に、追加したテクスチャノード内で、使いたい出力を選ぶことができます(例: "Diffuse" や "Normal")。

関連する情報

- [Blender 2.49 開発ページ\(英語\)](#)

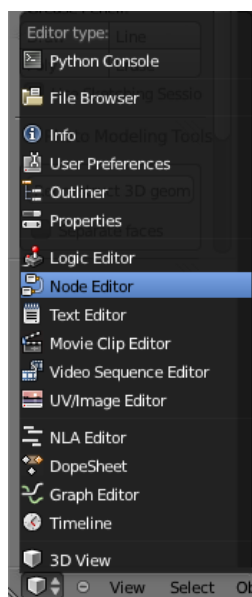
The Node Editor

(ノードエディタ)

このページでは通常のウィンドウと、ヘッダーメニューにあるオプション機能について説明します。Blender 内でノードを有効化する方法も説明します。

Accessing The Node Editor

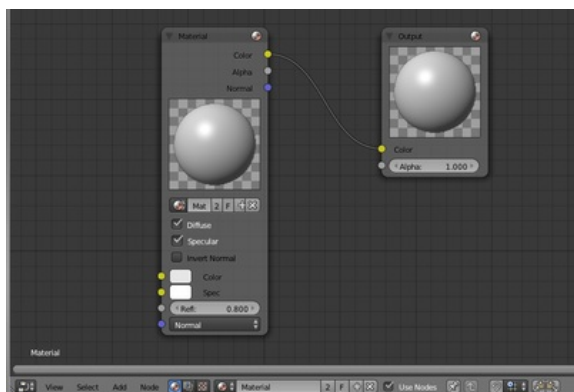
(ノードエディタを開く)



Node Editor ウィンドウの
選択

スクリーンレイアウトから「Compositing」を選ぶと、ノードエディタを含むウィンドウ配置になります。もしくは、任意のウィンドウのタイプを Node Editor に変えて、ノードエディタに入りましょう。図「Node Editor ウィンドウの選択」にあるように、ウィンドウタイプアイコンをクリックし、ポップアップリストから Node Editor を選びます。ノード図は非常に大きくなることもあるので、広いウィンドウを使うか、大きめのウィンドウを作ってください。ウィンドウには方眼紙様式の背景と、ヘッダーがあります。

blend ファイルにある各シーンは、複数のマテリアルノード図と「一つの」コンポジティングノード図を持つことができます。ノードエディタウィンドウにはノード種類セレクターで選択されている、いずれかのノード図が表示されます。



ノードエディタ

デフォルトではヘッダーの最初の表示は次のようになります:



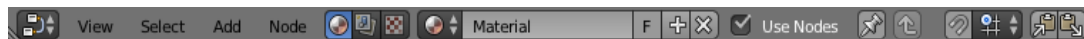
デフォルトのノードエディタのヘッダー

Activating Nodes

(ノードの有効化)

- どのノードを使うべきか？
 - マテリアル(シェーダー)ノード図で作業したいときは、ノード種類セレクターの球体のアイコンを選択します(図「マテリアルノードが有効なノードエディタのヘッダー」)
 - テクスチャノード図で作業したいときは、ノード種類セレクターのチェッカー模様のアイコンを選択します(図「テクスチャノードが有効なノードエディタのヘッダー」)
 - コンポジティングノード図で作業したいときは、ノード種類セレクターの写真の重なったアイコンを選択します(図「コンポジティングノードが有効なノードエディタのヘッダー」)
- ノードを実際に有効化するには Use Nodes ボタンを押します

- 最初にマテリアル(シェーダー)、コンポジティングノード、テクスチャノード図のいずれかを選んだ時は、ノードエディタウィンドウにはすでに結合された入力ノードと出力ノードが置かれます。



マテリアルノードが有効なノードエディタのヘッダー



コンポジティングノードが有効なノードエディタのヘッダー



テクスチャノードが有効なノードエディタのヘッダー

Node Editor Window Actions

(ノードエディタウィンドウでの操作)

カーソルがウィンドウ内であれば、次のような Blender の標準的なホットキーやマウス操作を使えます:

ポップアップメニュー

- ⇧ ShiftA - 追加するノードを選ぶポップアップメニューを表示します。

削除

- X か Del - 選択ノードを削除します。

ボックス選択

- B - 範囲選択を開始します。カーソルを移動して LMB クリックし、ドラッグして複数ノードを選択します

接続の切断

- Ctrl LMB クリックしてドラッグ - アイコンが小さなナイフに変わり、ノード間を接続するスレッドをまたぐようにドラッグして切断します

取り消し (Undo)

- CtrlZ を押し忘れてボックス選択をしようとしたとき、便利です。

やり直し (Redo)

- CtrlY か Ctrl⇧ ShiftZ - ちょっと「Undo」しすぎたときに使えます :)

複数選択

- ⇧ Shift LMB か ⇧ Shift RMB - 複数ノードを選択します。

つかむ/移動

- G - 選択中のものを動かします

通常のウィンドウ操作

ノード図は非常にひげもじゃく(広くて入り組んでいるという意味です)になる可能性があります。ウィンドウの中身(ノード図)は他の Blender のウィンドウと同じように、MMB クリックしてドラッグすると平行移動することができます。Wheel を上下させるかテンキーの + NumPad/- NumPad を使うと、ズームイン/ズームアウトができます。通常のウィンドウと同じようにウィンドウのサイズを変えたり、結合させたりできます。

Node Editor Header

(ノードエディタのヘッダー)

At a glance

(項目一覧)

ウィンドウのヘッダーには次のオプションがあります:

- View (ビュー) - はっきり見たいものを選びます
- Select (選択) - 条件をつけてノードを選択します
- Add (追加) - ノードを追加します。種類別にまとめられています
- Node (ノード) - 選択したノードに対する操作です。頂点のメニューに似ています
- マテリアル/テクスチャ/コンポジティングノードセレクタ
- Use Nodes (ノードを使用) ボタン
- ピン止め ボタン
- 親ノードに移動 ボタン
- スナップ ボタン
- スナップノードセレクタ
- コピーノード ボタン
- ペーストノード ボタン
- Free Unused (未使用ノードを解放) ボタン
- Backdrop (背景) ボタン
- Auto Render (自動レンダリング) ボタン

Menus

(メニュー)

View, Select and Add

(ビュー、選択、追加)

基本機能を持つメニューです:

View(ビュー)

ウィンドウの視点を変えるメニューです。+ NumPad (ズームイン)、- NumPad (ズームアウト)、 \wedge Home (すべてを見渡せるまでズーム)といった標準的なキーボードショートカット、または同等のマウス操作の代わりになります。

Select(選択)

ノードやノードグループを選ぶメニューです。全選択 A やボックス選択の開始 B といったホットキーの入力と同じ操作ができます。

Add(追加)

ノードを追加するためのメニューです。追加できるノードの種類やその役割については、次の節をご覧ください。このメニューにある項目は、ウィンドウ内にカーソルがあるときに \diamond ShiftA を使うのと同じものです。

Node

(ノード)

Hide(隠す)

H - 選択ノードを隠します。メッシュ操作と同じ要領です。

Grouping(グループ作成)

最重要項目で、ユーザー定義のノードグループを作ることができます。作成したグループは編集したり、ノード図に追加したりできます。グループを作るには、対象ノードを選び、Node(ノード) » Make Group(グループ作成) を選ぶか、キーボードショートカット CtrlG を使います。名前はグループ内にある小さな入力欄で変更できます。緑色のヘッダーや、あなたのつけたクールな名前によってグループが簡単に見分けられます。

Delete(削除)

X - 選択ノードを削除します。

Duplicate(複製)

\diamond ShiftD - 元のものと同じ設定を持ち、元のものとはつながっていない複製を作ります。

Grab(つかむ/移動)

G - ノードをマウスで動かします。メッシュ操作と同じです

複製されたノード

ノードを複製すると、新しいノードは **ぴったり元のノードの上に重なります**。下にあるノードにスレッドがつながっているように見えたとしても、複製はつながっていないので、コントロールをいじっても画像には何も起きません。複製を動かして下にあるつながったノードを見えるようにする必要があります。

Buttons

(ボタン)

Material/Composite/Texture Selector

(Material/Composite/Texture セレクター)

ノードは操作対象によってグループ分けされています:

- [Material Nodes \(マテリアルノード\)](#) / [Blender 2.67 では Shader Nodes \(シェーダーノード\)](#) で作業するには球体のアイコンをクリックします。
- [Compositing nodes \(コンポジットングノード\)](#) で作業するには写真の重なったアイコンをクリックします。
- [Texture nodes \(テクスチャノード\)](#) で作業するには、チェッカー模様のアイコンをクリックします。

Use Nodes Button

(ノードを使用)

有効にするとマテリアル色の計算や最終的な画像のレンダリングにノード図を使うよう、レンダリングエンジンに指示します。チェックを外すとノード図は無視され、マテリアルタブやシーンの基本のレンダリングが行われます。

Use Pinned Button

(ピン止めを使用)

レンダーエンジンにピン止めされたノードツリーを使うよう指示します。

Go to Parent Button

(親に移動)

親ノードツリーに移動できます。

Snap Button

(スナップ)

ノードエディタウィンドウ内でのノードのスナップモードを切り替えます。

スナップノード要素 セレクタ

スナップ対象要素を次から選べます:

Grid (グリッド、デフォルト)

ノードエディタウィンドウのグリッドにスナップ

Node X (ノードX)

ノードの左右の縁にスナップ

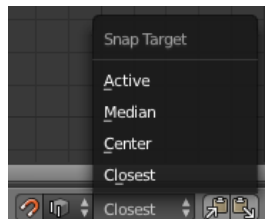
Node Y (ノードY)

ノードの上下の縁にスナップ

Node X/Y (ノードX/Y)

ノードの縁にスナップ

ノード縁へのスナップは次のスナップターゲットを考慮します:



Snap Target

Snap Target

どの部分をスナップさせるか選べます。

Closest (近接): 一番近い位置をスナップ

Center (中心): 中心をスナップ

Median (中点): 中点をスナップ

Active (アクティブ): アクティブ要素をスナップ

Copy Nodes Button

(コピーノードボタン)

選択ノードをクリップボードにコピーします。

Paste Nodes Button

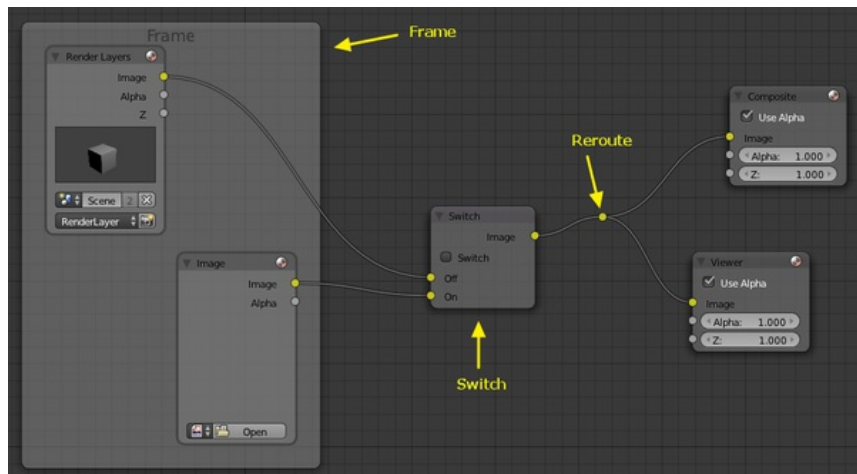
(ペーストノードボタン)

アクティブなノードツリーに、クリップボードからノードをペーストします。

Layout Nodes

(レイアウトノード)

Layout ノードはノードエディタウィンドウのノード配置の改善用に設計されています。Add (追加) » Layout (レイアウト) から利用できます。



レイアウトノードの例

Blender は次のレイアウトノードを備えます:

Frame (フレーム)

このノードは単に1箇所に複数ノードをまとめるために使います。

Reroute(リルート)

複数スレッドを枝状にしてノードエディタウィンドウのスペースを節約するためのものです。

Switch(スイッチ)

カラー(色)値の切り替えに使われるノードです。コンポジットノードでのみ使えます。

Free Unused Button

(未使用ノードを解放)

(コンポジットノード専用)非常に複雑なノード図で、未使用ノードのメモリ空間を解放します。お勧めです。

Backdrop

(背景)

(コンポジットノード専用)アクティブな Viewer ノードの出力を背景に使います。有効にすると、ヘッダーとプロパティパネルにさらに設定が現れます:


Backdrop Channels(背景に使うチャンネル)

画像の Color、Color and Alpha または Alpha だけを表示します。

Zoom

背景画像の大きさを指定します

Offset

背景の画面に対する位置を変更します。Move ボタンを押すかショートカット Alt MMB  を使って実際に動かしながら変えることもできます。

Auto Render

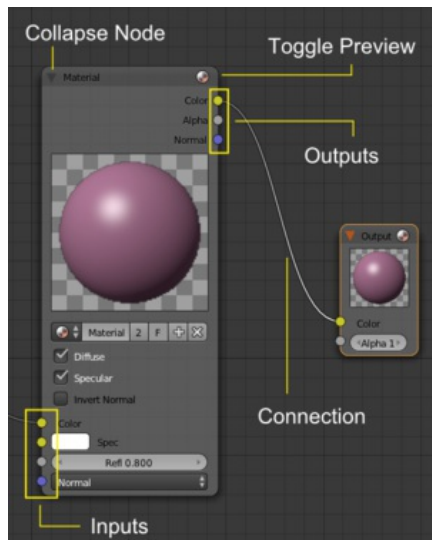
(自動レンダリング)

(コンポジットノード専用)3Dシーンで編集が行われると、変更のあったレイヤーの再レンダリングと合成を行います。

Node Controls

(ノードのコントロール)

このページではノードを操作するウィジェットを説明します。



ノードの主なコントロール

上から Collapse Node: 折り畳み Toggle

Preview: プレビュー切り替え Outputs: 出力

Connection: 接続 Inputs: 入力

タイトルバー

ノード名が表示され、折り畳み用のボタンがあります。

入力 (Input) ソケット

ノードの左側には入力ソケットがあります:

- 青色ソケットはベクトルを受け入れます
- 黄色ソケットは色を受け入れます
- 灰色ソケットは単独の値を受け入れます (例: アルファ)

出力 (Output) ソケット

ノードの右側には出力ソケットがあります:

- 青色ソケットはベクトルを作ります
- 黄色ソケットは色を作ります
- 灰色ソケットは単独の値を作ります (例: アルファ)

画像プレビュー

ノード内にはこのノードが出力する画像のプレビューやノードの動作を制御する曲線 (例えば RGB ノードにあるもの) を表示する領域があります。

ボタンとメニュー

画像プレビューの下には、ノードの動作を制御するボタンとメニューがあります。

スレッド

曲がった線は出力ソケットから入力ソケットへの接続を示しています。ソケットの種類は一致しなければなりません。

アクティブノードに関連した接続は見やすくなるようにハイライトされます。

[edit] Collapsing toggles

(折りたたみ表示の切り替え)

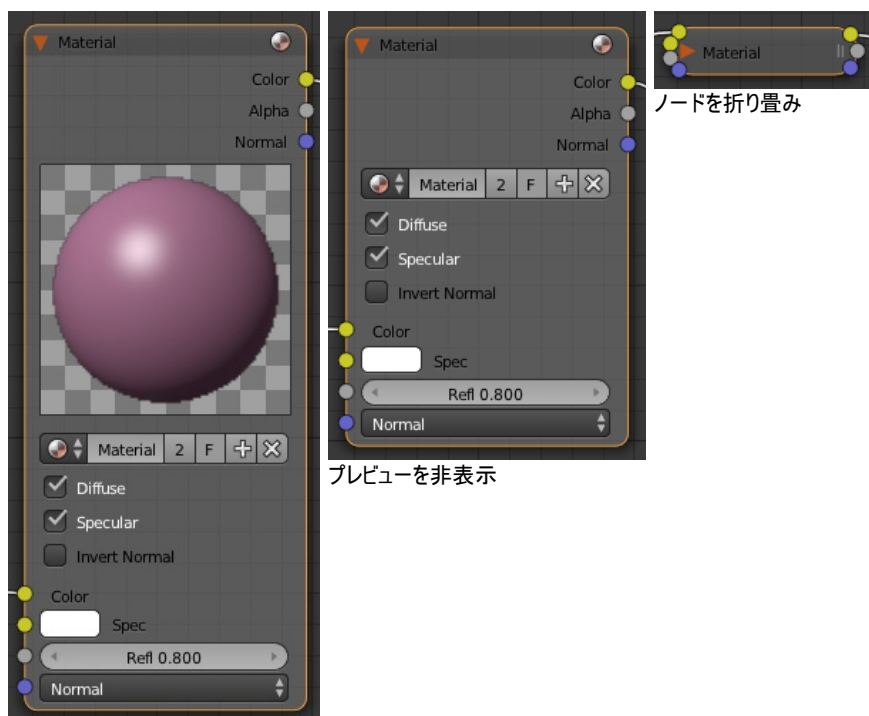
ノードの最上部には最大4つの外見に関わるコントロールがあります。クリックするとノードが表示する情報量に作用します。

ノード切り替え (▼▶)

左にある三角形はノードを折り畳み/展開します。

プレビュー画像の切り替え

タイトルバーの一番右にある球のボタンはプレビュー画像の表示/非表示を切り替えます。



すべて表示

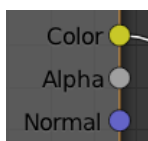
[edit] Sizing the node

(ノードのサイズ調整)

ノードの左右の端を LMB クリックしてドラッグするとノード個別にサイズの調整を行えます。

[edit] Sockets

(ソケット)



ノードソケット

ノードウィンドウにある各ノードは色のついた小円形の「ソケット」(「コネクタ」ともよく呼ばれます)を持ち、入力データや出力データをつなげられます(図「ノードソケット」)。

ノード左側にあるソケットは *入力*を、右側にあるソケットは *出力*を表します。

便利のようにノードは送信/受信することを期待される情報の種類によって *色分け*されています。三つの色があります:

- **黄色ソケット**
色 情報が入力に必要、またはノードから出力されることを示します。
- **灰色ソケット**
値(数) 情報を示します。単独の数値かいわゆる「値マップ」を使えます(値マップは各点の値が明暗に反映されるようなグレースケールのマップだと考えることができます)。「値マップ」ソケットの入力として単独の値が使われると、マップのすべての点がこの値と同じ値になります。
一般的な用途: アルファマップやノード用の値オプション。
- **青/紫色ソケット**
ベクトル/座標/法線 情報を示します。

ノード間は、コンバータ(*converter*)を使わない限りは、黄は黄に、灰は灰に、青は青に繋げなければなりません。コンバータについては後ほど説明します。

ノードの色名の隣にはソケット名が表示されています。常にとは言えませんが、ソケットが何の情報を表すものか名前から想像できます。しかし必ずしも「従わねばならない」ものではありません。例えば Alpha と書かれた灰色ソケットから Material ノードの灰色の Reflection ソケットにつなぎ、結果を得ることができます。重要なのはこれが「灰色から灰色」への接続であるということです。

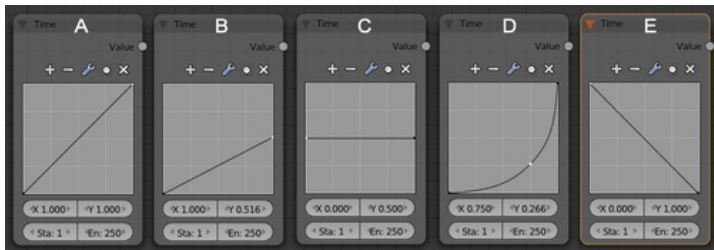
黄色(例: カラー画像)と灰色(例: グレースケール)をコンバータなしで組み合わせられる、例外があります。必要なら普通は Blender がコンバータを置くので、気軽に試してみてください。この後で説明する "Viewer" 出力ノードを使うと、動作するかどうかや、動作の様子を見ることができます。

[edit] Curves

(曲線)

ノードによっては入力値を出力値に変換するための曲線領域を持つものがあります。曲線の形状は制御点をクリックして動かすか、制御点を追加し

て編集できます。以下は例です:



Curve ノードの変形

どの曲線も傾き 1 の直線が始まります。曲線は始め両端に小さな黒い制御点を持っています。制御点を LMB クリックすれば選ぶことができ、制御点は白くなります。

曲線の変更は出力の生成のされ方に影響します。入力 X は普通 底の軸に交差して線形に(一定の間隔を持って)進みます。曲線にぶつかるまで上に進み、ぶつかれば右に進んで、X に対応する Y 出力を決めます。二つ目の例では、X が底に交差して 0 から 1.0 に進むに連れ、Y は 0.0 から 0.5 まで変化しています。3つ目の例では X は 0.0 から 1.0 に進みますが、Y は 0.5 に留まり続けています。したがって、上の図の曲線は次のような時間的効果を持ちます: **A** は影響なし、**B** は遅延させ、**C** は停止、**D** は加速、そして **E** は時間を逆向きにします。

「曲線」ウィジェットはどこでも使える Blender UI の組み込み機能で、与えた曲線データはこのウィジェットに渡されています。現在のところノードエディタと UV ウィンドウ内で使われています。

ウィジェットは水平方向に入力値をマップし、曲線の高さで示される新たな値を返します。

註釈: スクリーンショットでは曲線上の点が一「白く」なっていますが、深い意味はありません。単に、著者がチュートリアルを準備しているときに、たまたま最後に選択した点だったというだけです。ここで重要なのは曲線の形状であって、それを定義するための制御点の位置(または色)ではありません。

[\[edit\]](#) RGB Curves

(RGB 曲線)

ひとつのウィジェット内で複数の曲線を編集できます。その代表的な使用法が RGB Curves ノードで、このノードは「すべてを組み合わせた」結果である「色 (C)」を最初の曲線として持ち、R、G、B 要素それぞれの曲線も持ちます。4つの曲線すべてが有効で、「C」の曲線が最初に評価されます。

[\[edit\]](#) Selecting curve points

(曲線の制御点の選択)

- LMB は常にひとつの点を選び、残りの選択を解除します。
- Shift を押しながらかlickすると、選択範囲を増やすか少なくすることができます。

[\[edit\]](#) Editing curves

(曲線の編集)

- 点を LMB クリックしてドラッグすると点を動かせます
- 曲線を LMB クリックすると新たな点を追加します
- 他の点にちょうど重なるように動かすと一つにできます
- Shift を押しながらかlickするとグリッド単位にスナップします
- Ctrl LMB は点を追加します
- 選択した点を削除するには X アイコンをクリックします

[\[edit\]](#) Editing the view

(ビューの編集)

デフォルトではビューは 0.0 から 1.0 の範囲に固定されています。デフォルトの設定である clipping が有効なら、ビューの拡大やドラッグは行えません。丸いアイコンを押して clipping を無効にできます。

- 曲線の外側を LMB クリックしてドラッグするとビューを動かせます
- + と - のアイコンを使ってズームイン/アウトを行います

[\[edit\]](#) Special tools

(特別なツール)

レンチアイコンはビューのリセット、各点の補間の定義、曲線のリセットを行うメニューを表示します。

Using nodes

(ノードの利用)

Adding Nodes

(ノードの追加)

ノードは次のいずれかの方法でノードエディタウィンドウに追加します。

- ノードエディタのツールバーにある Add メニューをクリックし、目的のノードタイプを選びます
- カーソルをノードエディタウィンドウに置いた状態で、⇧ ShiftA を押し、ポップアップの Add メニューからノードを選びます


Arranging Nodes


(ノードの並べ替え)

ふつうはウィンドウ内でノードを並べ替えて、画像が左から右、上から下に流れるようにしてください。コントロールなどに影響しない部分をドラッグしてノードを動かします。ノードはほとんどどこでもクリックし、ドラッグすることができます。すべての接続は可能な限りベジエ曲線として再整形されます。

Connecting nodes

(ノードの接続)



ソケットを LMB -クリックしてドラッグすると、枝が伸び出てきます。これは「スレッド」と呼ばれます。

ドラッグしたまま他のノードの入カソケットにスレッドをつなげ、それから LMB  を離します。

この場合各出力のコピーがスレッドに沿って送信されます。一方、ひとつの入カソケットにはひとつのスレッドだけをつなげることができます。

Disconnecting nodes

(ノード接続の解除)

ソケット間のつながりを切るには、切りたいスレッドに近い何も無い空間を Ctrl LMB -クリックしてドラッグします。マウスポインタは小さなカッターナイフのアイコンになります。スレッドを横切るようにマウスを動かして、LMB  を離します。

Duplicating a node

(ノードの複製)

目的のノードをクリックして ⇧ ShiftD を押し、元のノードが見えるようにマウスを動かします。

わかった！

ノードを複製すると、新しいノードは *正確*にもとのノードの上に重なります。そのままにしておく(そうするのは本当に簡単です)、ふたつノードがあることが簡単にはわからなくなります！ 疑わしいときは、ノードをつかんで少し移動し、何か隠れているものがないか確かめてみてください。

Node Groups

(ノードグループ)

マテリアルとコンポジットのどちらのノードもグループ化できます。ノードをグループ化すれば、ノードエディタ上でノード回路網の配置を単純化でき、ヌードル(ノードの回路網)が扱いやすくなります。またノードグループ(.blend ファイル内)またはノードツリー(アペンド時)と呼ばれるものが作られます。

概念的には「グループ化」として「単体のノード」であるかのように扱えるノードの *セット* を定義することができます。ノードグループは同じ .blend ファイル内でも別の .blend ファイルからも、何度でも再利用できます。

例えばノードを使って作ったマテリアルを、別の .blend ファイルに単に追加することができます。しかし新たに作ったマテリアルから、既存のマテリアルノードの回路網から一部の分岐を使いたいときはどうでしょう？ 分岐を再作成することもできるでしょう。もしくはそのマテリアルを新たな .blend ファイルに追加して、使いたい分岐をカット&ペーストして新たなマテリアルに足すこともできるでしょう。どちらの方法もうまくいきますが、.blend ファイル間で作業するときはそれほど効率が良くありません。「被写界深度」のコンポジットノードの回路網を作り、別の .blend ファイルで使いたいときにはどうでしょう？ 一連の操作を正確に何十回も繰り返したいときは？ この場合も回路網の再作成はできますがそれほど効率が良くありません。マテリアルノードの分岐とコンポジットノード回路網のどちらにおいても、ノードグループを作って再利用するのが上手なやり方と言えるでしょう。

一旦定義したグループは、再利用可能なソフトウェア部品になり、ブラックボックス化します。定義を厳密に知らなくても、働きを知れば(何度でも好きなだけ)利用できます。グループは [Blender のライブラリと標準的な追加\(アペンド\)方法](#) を通じて利用可能な状態にできます。

Grouping Nodes

(ノードのグループ化)

パネル: [Node Editor](#)

メニュー: ◊ ShiftA » Group (グループ) » Make Group (グループ作成)

ノードグループを作るには、ノードエディタでグループに含めたいノードを選択し、CtrlG か ◊ ShiftA » Group » NodeGroup を使います。ノードグループは緑色のタイトルバーを持ちます。選択したノードはすべて最小化され、ノードグループ内に入ります。ノードグループのデフォルトの名前は *NodeGroup*、*NodeGroup.001* といったものです。ノードグループには名前を入力欄があり、クリックして変更できます。名前を意味のあるものに変えてください。別の .blend ファイルにノードグループをアペンドしたときに、Blender はマテリアルノードグループとコンポジットノードグループの区別をさせません。したがってこの種類を見分けやすくする命名規則を作るのがよいでしょう。例えばマテリアルノードの分岐には *Mat_XXX*、コンポジットノードの回路網には *Cmp_XXX* と命名します。



グループに含むべきではないもの(すべてのタイプのノードエディタ)

覚えておきたいのは基本思想として、グループは簡単に再利用できる、自己完結したソフトウェア部品であるべきだということです。マテリアルノードグループは以下のノードを **含むべきではありません**：

Source nodes (ソースノード)

グループ内にソースノードを含めると、最終的にソースノードがグループ内とグループ外の新たなマテリアルノード回路内の二箇所にてきます。

ソースノードの例としてマテリアルノードの *Material* ノードや、コンポジットノードの *Render Layers* ノードがあります。

Output node (出力ノード)

グループ内に出力ノードを含めると、このグループからの出力ソケットが使えなくなります！

出力ノードの例としてマテリアルノードの *Output* ノードや、コンポジットノードの *Viewer* ノードがあります。

Editing Node Groups

(ノードグループの編集)

ノードグループを選択して ⇄ Tab を押すか、ノードグループ右肩の「ノード」アイコンをクリックすると、ノードがウィンドウ枠いっぱいになり、グループ内にある各ノードが表示されます。各ノードに対して移動や、コントロールの調整、スレッドのつながりなど、通常のエディタウィンドウの一部であるかのような操作ができます。外部にあるノードと直接スレッドをつなぐことはできず、グループノードの端にある外部ソケットを使う必要があります。ノードをグループに追加したりグループから削除するには、一旦グループを解除する必要があります。

Ungrouping Nodes

(ノードグループの解除)

AltG コマンドはグループを壊して、各ノードをエディタの作業領域に置きます。内部の接続は保持され、内部にあったノードを作業領域内にある外側のノードと接続できるようになります。

Appending Node Groups

(ノードグループの追加)

ノードツリーをグループとして .blend ファイルに一旦追加すれば、ノードエディタ内で ◊ ShiftA » Add (追加) » Group (グループ) を選び、追加したグループを選ぶとそのグループを利用できます。グループの「コントロールパネル」はノードグループ個別にあります。他のノードと同じように操作できます。

Volume Textures

(ボリュームテクスチャ)

ボリュームに使えるテクスチャは 2種類あります:

[Voxel Data\(ボクセルデータ\)](#)

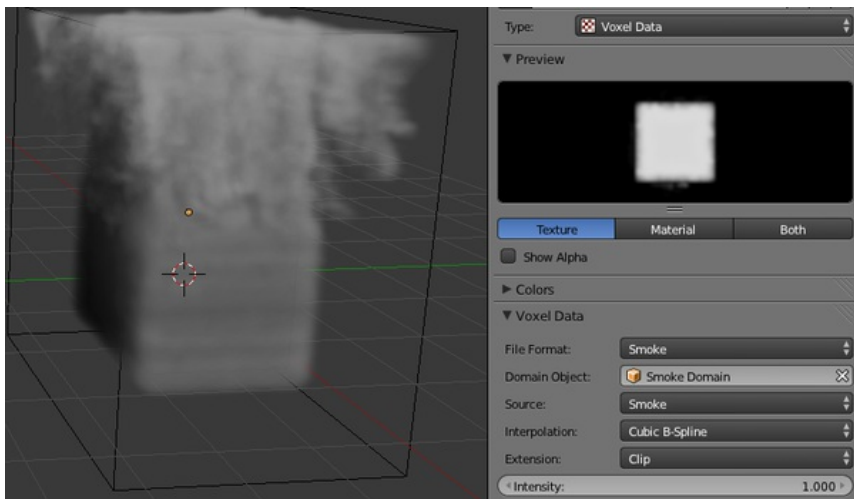
Voxel(3D上のピクセルに相当)をレンダリングするものです。Blender 内蔵の煙のシミュレータで使うことができます。それ以外の入力源 Binary Raw 形式、Image Sequence(連続画像)などは連続画像から 3D 表現を作ります。

[Point Density\(点密度\)](#)

Point Cloud(点群。オブジェクトの頂点かパーティクルシステム)を 3D ボリュームとしてレンダリングするものです。

Voxel Data

(ボクセルデータ)



Voxel Data texture

Voxel データは、入力された Voxel をレンダリングします。Image Texture (画像テクスチャ) と非常に似ていますが、3D です。入力に使えるデータ種類は多様で(たとえば煙の Voxel データや外部ファイル) 補間方式もたくさんあります。

Voxel は平らな浮動小数点数の z/y/x グリッドに記憶されます。次の Blender ソースコードで定義されているサンプリング関数が、(0,1) の領域内の位置に基いてサンプリングを行っています:

- `source/blender/blenlib/intern/voxel.c`

voxel データのデフォルトの入力源である煙は、Blender 内蔵の煙のシミュレーションのレンダリングに使われます。他の入力源として Binary Raw 形式や連続画像があります。これは 3D 表現用の連続画像の作成に使われるもので、CT スキャンなどの医療用ボリュームデータとして一般的な形式です。

Settings

(設定項目)

File Format (ファイル形式)

Blender Voxel (Blender ボクセル)

デフォルトのバイナリ voxel ファイル形式です。

8 bit RAW

8ビットのグレースケールのバイナリデータです。

Image Sequence (連続画像)

連続する画像スライスから Voxel を生成します。

Smoke (煙)

Blender の煙シミュレーションから Voxel をレンダリングします。

Source Path (パス)

入力に使う外部のデータファイルです。8ビットの Raw データと Blender Voxel 形式で使います。

Domain Object (領域/ドメインオブジェクト)

煙シミュレーションで領域定義に使うオブジェクトを指定します。

Source (入力)

Smoke (煙)

煙の密度と色をテクスチャデータとして使います。

Flame (炎)

炎の温度をテクスチャデータとして使います

Heat (温度)

煙の温度をテクスチャデータとして使います。使われる値は -2.0 から 2.0 です。

Velocity (速度)

煙の速度をテクスチャデータとして使います。

Resolution(解像度)

Voxel グリッドの解像度。8 ビットの Raw データ形式で使います。

Interpolation(補間)**Nearest Neighbor**

補間しません。高速ですが、むらができ低品質です。

Linear

滑らかさや速さに優れています

Quadratic

中程度の品質と速さです

Cubic Catmull-Rom

滑らかで高品質な補間ですが、低速です。

Extension(延長)**Extend(延長)**

画像の端にあるピクセルを繰り返して延長します。

Clip(切り抜き)

画像の大きさにクリップし、外側のピクセルを透明にします。

Repeat

画像を縦横両方に繰り返します。

Intensity(強度)

intensity 値用の乗数

Point Density Texture

(点密度テクスチャ)

Point density(点密度)は、点群(オブジェクトの頂点やパーティクルシステム)を、点に対してユーザーが定義した半径を使い、3Dボリュームとしてレンダリングします。内部的には、システムは 高速な範囲参照のために BVH データ構造を使います。

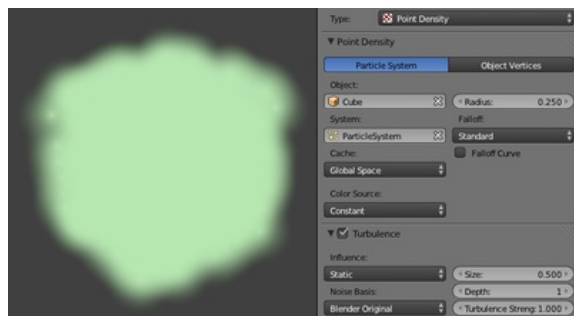
レンダリングされた点はデフォルトでは球になり、いろいろな滑らかな減衰オプションだけでなく、ノイズ値で結果をずらす簡単な乱流オプションを使って、細部を付け加えます。点密度をパーティクルシステムといっしょに使うと、パーティクルの velocity、age、speed などの追加情報を色/アルファ傾斜のグラデーションで視覚化できます。

Options

(オプション)

Point Density

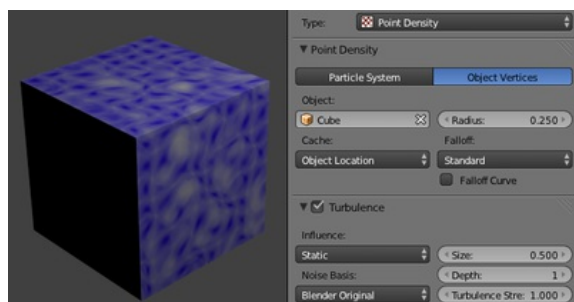
(点密度)



パーティクルシステムを使った 点密度 テクスチャ

Particle System(パーティクルシステム)

パーティクルシステムから点密度を生成します。



cloud オブジェクトの頂点を使った 点密度 テクスチャ

Object Vertices(オブジェクトの頂点)

オブジェクトの頂点から点密度を生成します。

Object(オブジェクト)

データの取得元オブジェクト

Radius(半径)

点の探索半径。shaded sample からの距離です

System(システム)

点としてレンダリングするパーティクルシステム

Falloff(減衰)

点からの距離に応じて密度を減衰させるための方式

Standard

Smooth

Soft

Softness

Constant(一定)

探索半径内で密度を一定にします

Root

Particle Age

Particle Velocity

Velocity Scale

Falloff Curve(減衰曲線)

カスタムの減衰曲線を使います。

Cache (キャッシュ)

パーティクルのキャッシュに使う座標系です。

Global Space (グローバル空間)
Emit Object Space (放射オブジェクト空間)
Emit Object Location (放射オブジェクトの位置)

Color Source (色ソース)

色の抽出元となるデータです。
Constant (一定)

一定の色を使います

Particle Age (パーティクル寿命)

寿命値は 0.0 - 1.0 の強度に変換されます

Particle Speed (パーティクルスピード)

パーティクルスピード (velocity の絶対的な大きさ) は、0.0 - 1.0 の強度に変換されます
Scale (拡大縮小)

パーティクルスピードを許容範囲内にする乗数です

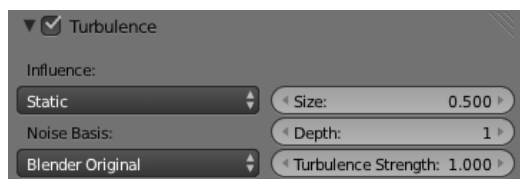
Particle Velocity (パーティクル速度)

XYZ velocity です。RGB カラーに変換されます。
Scale (拡大縮小)

パーティクル速度を許容範囲内にするための乗数です

Turbulence

(乱流)



Turbulence panel

レンダリング時に、向きを持ったノイズを密度に加えます。

Influence (影響)

加えた乱流ノイズの制御方式です。
Static (固定)

ノイズパターンは固定、高速で、スチール写真に向いています

Particle Velocity (パーティクル速度)

乱流ノイズをパーティクル速度で制御します。

Particle Age (パーティクル寿命)

乱流ノイズをパーティクルの誕生から死までの年数で制御します。

Global Time (グローバルな時間)

乱流ノイズをグローバルの現在のフレームで制御します。

Noise Basis (基本ノイズ)

[こちら](#) をご覧ください。

Size (サイズ)

乱流ノイズの大きさ

Depth (深度)

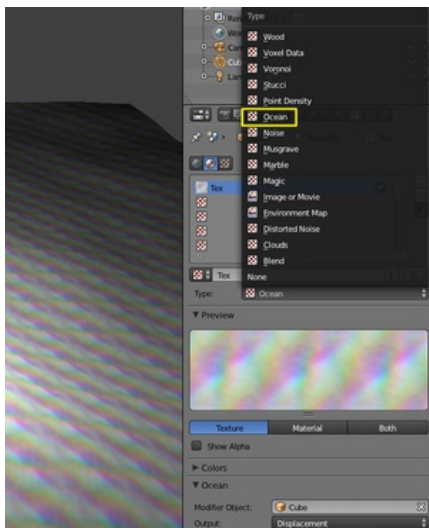
乱流ノイズの詳細度

Turbulence Strength (渦の強さ)

乱流ノイズの強さ

Ocean Texture

(海洋テクスチャ)



Ocean テクスチャ

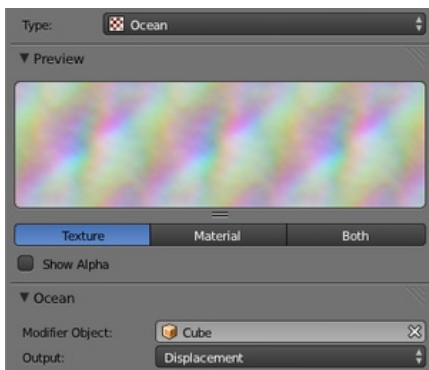
[Ocean\(海洋\)](#) モディファイアによって生成されるテクスチャです。

Options

(オプション)

Ocean panel

(海洋パネル)



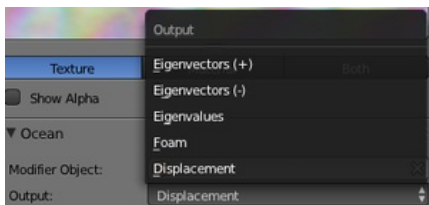
Ocean(海洋) テクスチャの オプション

Modifier Object(モディファイアオブジェクト)

ocean(海洋)モディファイアを入れたオブジェクト。

Output(出力)

テクスチャの出力するデータ



Ocean(海洋)テクスチャの Output(出力)設定

Displacement(ディスプレイスメント)

RGBチャンネルに XYZ ディスプレイスメントを出力します

Foam(泡)

泡(波頭の交差)の量を単一チャンネルに出力します。

Eigenvalues

正の固有値

Eigenvector (-)

負の固有ベクトル

Eigenvector (+)

正の固有ベクトル

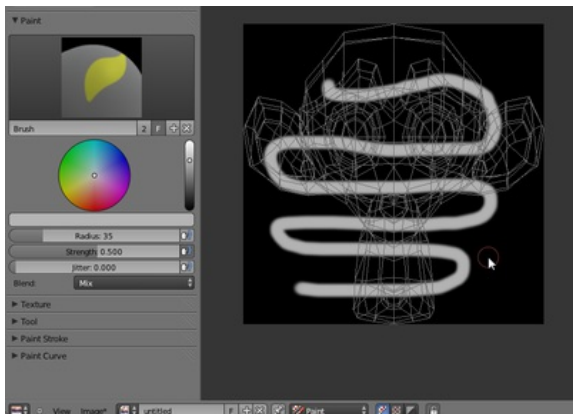
Texture Painting

(テクスチャペイント)

UVテクスチャとは、メッシュ表面の着色に使われる絵(画像、連番画像、動画)です。ひとつ以上の UVマップを通じてメッシュに対応付けられます。UVテクスチャに使う画像を作る方法は3つあります。

- UV/画像エディタで選択中の UVテクスチャの上に平面の画像を塗り、UVマップを使って色をメッシュの面に転送します。
- 3D ビューでメッシュに色を塗ります。選択中の UVマップを使って UVテクスチャが更新されます ("[Projection Painting \(投影ペイント\)](#)") をご覧ください。
- 任意の画像編集(ペイント)プログラムで画像を作ります。UV/画像エディタで UVテクスチャを選び、その画像を読み込みます。色はテクスチャの UVマップを使ってメッシュの面に転送されます。

Blender には、特に UVテクスチャや画像を手軽に編集できるように設計された、組み込みの Texture Paint(テクスチャペイント)と呼ばれるペイントモードがあり、UV/画像エディタウィンドウと、3Dビューウィンドウのどちらでも使うことができます。UVテクスチャは単に専門的な用途を持つ「画像」なので、任意の外部のペイントプログラムでも編集できます。例えば Gimp は画像操作用のフル機能を備えたプログラムで、オープンソースです。



Blender のテクスチャペイント

メッシュには UVテクスチャを重ねることができるので、メッシュ着色用の画像がたくさんあるかもしれません。ただし各 UVテクスチャはひとつしか画像を持ってません。

テクスチャペイントは 3Dウィンドウと UV/画像エディタウィンドウのどちらでも動作します。3Dウィンドウでテクスチャペイントモードに入ると、[UVに投影を行って](#)、メッシュを直接塗ることができます。

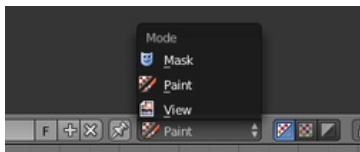
Getting Started

(はじめましょう)

モデルを展開(Unwrap)して UVマップを作ったら(これについては以前のページで説明しました)、テクスチャを貼る作業を開始できます。テクスチャペイントモードでの描画は、「まず」メッシュを展開し、「さらに」次のいずれかの手順を踏まないと行えません:

- UV/画像エディタに [画像を読み込みます](#) (Image(画像) » Open(開く) » ファイルを選択)。もしくは
- [新規画像を作ります](#) (Image(画像) » New(新規) » サイズを指定)。

次のいずれかを行えば、テクスチャペイントモードを使った画像編集ができます:



ペイントモードの有効化

- 3Dビューウィンドウで、ウィンドウヘッダーにあるモード選択メニューからテクスチャペイントモードを選ぶと、メッシュ上に直接描画できます。
- UV/画像エディタウィンドウで編集コンテキストを View(ビュー)から Paint(ペイント)に切り替えます(右図)。

2の乗数の正方形

画像サイズを2の乗数サイズの正方形にすると、テクスチャペイントはとても高速で反応がよくなります。256x256、512x512、1024x1024 などで

テクスチャペイントを始めると、マウスがブラシ(Brush)に変わります。座標の移動など UVレイアウトに働きかけるときは、View モードに戻る必要があります。

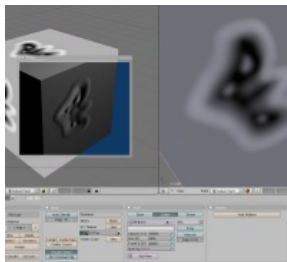
テクスチャペイント有効化するか、テクスチャペイントモードに切り替えるとすぐに、ツールシェルフ(T)でブラシ設定を行えるようになります。

UV/画像エディタウィンドウでは、UV座標を使ってメッシュを包んでいる平らなキャンバスに描画します。UV/画像エディタウィンドウで行った変更は、すぐに 3Dウィンドウに反映されます。逆も同様です。

UV/画像エディタのツールシェルフで、すべてのブラシと塗り色を選べます。どのパネルでブラシを変更しても、すぐに他のパネルに反映されます。ただしテクスチャの編集は自動的に保存 **されません**。UV/画像エディタウィンドウの画像(Image)メニューから、自分で保存する必要があります。

Texture Preview

(テクスチャのプレビュー)



パンプのペイント(UIは Blender 2.4のもので)

テクスチャがシーン内にあるモデル表面用のカラー、パンプマップ、ディスプレイス、アルファ透過等にすでに使われていれば(具体的には UV を入力にし、テクスチャチャンネルを通じてテクスチャの各特性に対応付けられていれば)、ブラシで塗ったときにシーンのコンテキストでその効果を見ることができません。

このためには、横並びのウィンドウを作り、片方を3Dビューにして描画モードを Textured (テクスチャ) にし、もう片方を UV/画像エディタウィンドウにして画像を読み込みます。読み込んだ画像にUVマップしているオブジェクトが見えるように、3Dビューを調整します。プレビューウィンドウを開いて(詳しくは3Dビューのオプションをご覧ください)、オブジェクト上に動かします。右側の画像でペイントされているテクスチャは「法線」にマップされています。これは「パンプマッピング」と呼ばれるもので、グレイスケールの濃淡により平らな表面に凹凸ができます。パンプマッピングの詳細は、テクスチャマッピングの出力をご覧ください。

Brushes Settings

(ブラシの設定)

UV/画像エディタで T を押すとツールシェルフが開きます。このパネルでは、色や幅など設定の異なるたくさんのブラシを作ることができます。ブラシ選択メニューで別のブラシに切り替えるか、新たにブラシを追加します。ブラシを追加すると元のブラシの複製ができます。そこから設定を調整して新たなブラシを作っていきます。テクスチャペイントで管理できるブラシやユーザー定義のコントロール(ペイントツールパネルで設定できます)の数に上限はありません。

ブラシを使うには、ブラシ名をクリックします。一度に表示できる数より多いブラシがあるときは、上下矢印のボタンを使って選びます。名前欄をクリックすればブラシ名を変えることができます。例えば赤色のエアブラシを "Red Air" にするといった具合です。ブラシを捨てるには、ブラシ名の隣にある削除ボタン X をクリックします。Blender の次回起動時にもブラシを持ち越したいときは、削除ボタンの隣の Fake User ボタンを押します。

筆圧感知機能のあるペンタブレットがあれば、透明度 (Opacity)、サイズ (Size)、減衰 (Falloff)、間隔 (Spacing) の各パラメータの隣にある、指で壁を押し込むアイコンのボタンを押すと筆圧が使われます。ペンの消しゴム側を使うと、Erase Alpha (アルファ消去) モードに切り替わります。

画像の任意の場所で S を押すと、そこで使われている色をブラシの色に設定できます。

Brush

(ブラシ)



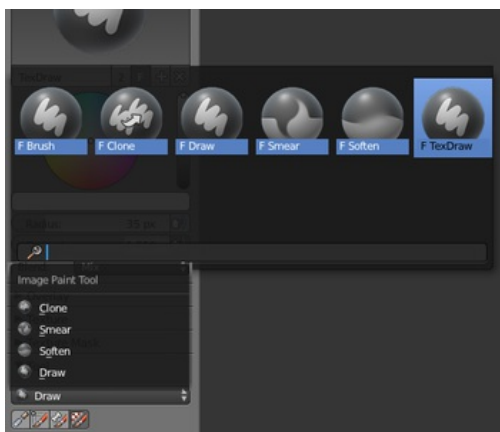
ブラシの設定

Brush presets (ブラシのプリセット)

Blender に付属のブラシを選びます。ほとんどのブラシに共通の設定があります。

Types of brushes

(ブラシの種類)



ブラシの種類

ブラシには4種類あります。

Draw(塗り)

普通のブラシ。帯状に色を塗ります。

Soften(ぼかし)

ブラシのふちの内側と外側の色を調和させます

Smear(こすりにじみ)

クリックしたときカーソルの下にある色を取得し、マウスを動かした向きに混ぜ合わせます。Gimp の "smudge" ツールに似ています。

Clone(クローン)

指定した画像(例では Tex.Dirt)から現在の画像へ色を複製します。このボタンが選ばれると背景画像が表示されます。Blend スライダーを使って背景画像の目立たせ方を調整します。

Enable Pressure Sensitivity(筆圧感知を有効化)

筆圧感知で効果の強度を変える機能です。次の3つの設定の隣にあるアイコンで切り替えます。

Color(色)

ブラシの色

Radius(半径)

ブラシの半径。ピクセル単位です

Strength(強度)

ブラシの適用される強さ

Blend(ブレンド)

他のテクスチャの上に塗られた色の適用方法を決めます

- Mix(混合): ブラシ色は既存色とミックスされます。
- Add(加算): ブラシの色は既存色に加算されます。緑色に赤色を加算すると黄色になります。
- Subtract(減算): ブラシの色は減算されます。紫色の上に青色を塗ると赤色になります。
- Multiply(乗算): 元の色 of RGB値がブラシ色で乗算されます
- Lighten(明るく): 元の色 of RGB値がブラシ色で増やされます
- Darken(暗く): 色調を下げます
- Erase Alpha(アルファ消去): 画像のペイント箇所を透明にし、背景色や下にあるテクスチャを透かします。このモードで「塗る」と、透明色であることを示すチェッカー模様の背景が見えるようになります。
- Add Alpha(アルファ加算): 塗った箇所の不透明度を増します

UV/画像エディタで Erase と Add Alpha モードの効果を見るためには、Display Alpha または Alpha-Only ボタンを押してアルファチャンネルを有効化しておく必要があります。透明(アルファがゼロ)の領域がチェッカー模様の背景で表示されるようになります。

Image(画像)

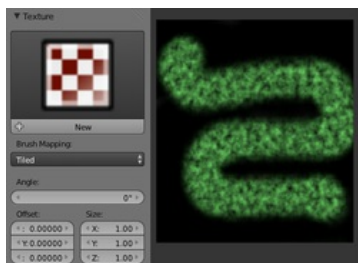
クローンブラシを使う場合、クローンの入力に使う画像を選びます

Alpha(アルファ)

画像の透明度を表示に反映します

Texture

(テクスチャ)



テクスチャオプションと例

ペイントパネルの最下部にあるテクスチャセレクターを使って、事前に読み込んだ画像か、プロシージャルテクスチャをブラシのパターンとして使うことができます。なお利用のためには、プレースホルダーになるマテリアルと、そのテクスチャを定義しておく必要があります。マテリアルやテクスチャがメッシュに用いられている必要はなく、定義されていることだけです。右の例は平面の(縞模様の)木のテクスチャで塗りを施した結果を示しています。テクスチャを輪に変えると花模様タイプのパターンになります。

註釈

クローンペイントモードでは、この欄はクローン元の画像やテクスチャを示すものになります。

Brush Mapping (ブラシの対応)

ブラシへのテクスチャの適用方法を決めます

View Plane (ビュー平面)

2D のペイントで、テクスチャがブラシとともに動きます

Tiled (タイル状)

ブラシ位置によってテクスチャがずらされます

3D (3D)

Tiled モードと同じです

Stencil (マスク)

テクスチャは型紙 (stencil) の境界線より内側にだけ適用されます。

Random (ランダム)

テクスチャのランダム適用。

Angle (角度)

テクスチャブラシの回転角度です。3D ビューで CtrlF を押すとインタラクティブに変更できます。インタラクティブな回転中はキーボードから数値での入力もできます。次のいずれかにセットできます:

User (ユーザー)

角度値を直接入力します

Rake (レーキ)

角度はブラシストロークの向きに従います。3D テクスチャでは使えません

Random (ランダム)

角度はランダムに決まります

Offset (オフセット)

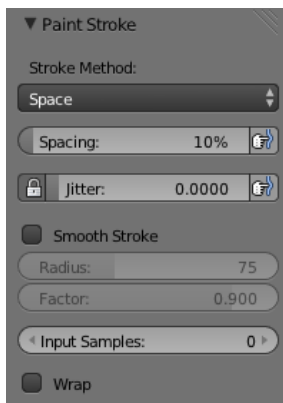
x、y、z で示したテクスチャのオフセット

Size (サイズ)

テクスチャの各軸方向への拡大縮小率

Stroke

(ストローク/筆致)



Stroke (ストローク)パネル

Stroke Method (ストローク方法)

ストロークの適用方法を設定できます。

Airbrush (エアブラシ)

マウスを押し続ける間、効果を適用し続けます

Rate (レート)

エアブラシの塗りの間隔です

Space (空間)

点の連なりとしてブラシストロークを作ります。間隔は Spacing 設定で決まります。

Spacing (間隔)

ブラシの直径に対する比率を表します。この距離にブラシの適用が限定されます。

Dots (ドット)

マウスが動くたびに塗りを適用します

Jitter (ジッター)

塗っている間、ブラシ位置を小刻みに揺らします

Smooth stroke

ブラシがマウスに遅れてついていき、滑らかな経路に補正されます。有効にすると次のオプションが有効になります:

Radius (半径)

前の位置からストロークが継続していると判定する最小距離を設定します。

Factor (強さ)

滑らかにする度合いを設定します

Input Samples (入力サンプル数)

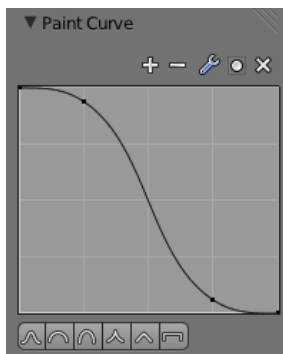
ブラシストロークを滑らかにするために複数の入力サンプルの平均を取ります。

Wrap

キャンパスの外にブラシが動いた時に、画像の反対側からブラシを出現させて塗りを続けます(上下、左右ともに働きます)。継ぎ目のないテクスチャを作るときに非常に便利です

Curve

(カーブ)



Curve panel

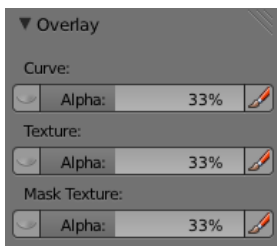
曲線を使ってブラシの減衰を調節できます。曲線の形を変えてブラシを柔らかくしたり硬くしたりします。

Paint options

(塗りのオプション)

Overlay

(オーバーレイ)

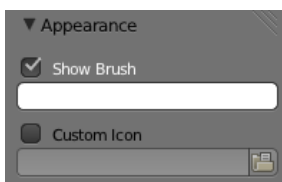


Overlay パネル

曲線やブラシに適用するテクスチャの表示がカスタマイズできます。

Appearance

(外見)



Appearance パネル

ブラシ領域の輪郭の色のカスタマイズやカスタムアイコンの指定ができます。

Saving

(保存)

ヘッダーにある Image (画像) メニューの項目名にアスタリスクがついていれば、画像の変更がまだ保存されていないことを意味しています。Image (画像) » Save Image (画像を保存) オプションで、作業内容を別名か同じ名前の上書きして保存します。

UV テクスチャ

UV テクスチャとして使われる画像は他の画像と機能的に違うものなので、他の画像と別のディレクトリに置くのがよいでしょう。

保存画像のファイル形式はレンダリング用の形式とは別のものです。UV 画像の保存に使う形式は、画像保存時のウィンドウのヘッダーで選ぶことができます。デフォルトは PNG (.png) です。

ウィンドウヘッダーで Pack (パック) が有効になっているか、手動で Image (画像) » Pack Image (画像をパック) を選んだときは、別のファイルに画像を保存する必要はありません。

Using an External Image Editor

(外部の画像エディタの利用)

UV テクスチャの編集に外部のプログラムを使うときは、次の手順が必要です:

1. そのペイントプログラムを実行 (Gimp、Photoshop など)
2. 画像を読み込むか新規に作ります
3. 画像を変更します
4. そのプログラムで再保存します
5. Blender に戻り、UV/画像エディタウィンドウでその画像を読み込みます

UV テクスチャを描くために別のプログラムを使っているか、テクスチャペイント機能で行えない特別な効果を使いたい場合、あるいは使い慣れたお気に入りのペイントプログラムがある場合には、そのプログラムを使うのがよいでしょう。

Projection Texture Painting

(投影テクスチャペイント)

投影テクスチャペイントを使うと、3Dモデルに対応づけたテクスチャに色を塗ることができます。Image(画像)エディタでのペイントとは違って、投影テクスチャペイントは Blender の 3D 描画領域で行います。

Getting Started

(はじめに)

テクスチャペイントモードに入るには、メッシュオブジェクトを選択し、(Object(オブジェクト)や Edit(編集)などのモードを選ぶ)モード選択メニューから Texture Paint(テクスチャペイント)を選びます。

3D モデルに色を塗るためには、少し準備が必要になります。Blender が画像と 3D モデルを対応づける方法が必要です。これは UV マップを使って行うので(詳しくは [UVマッピング](#) をご覧ください)、モデルがまだ展開(unwrap)されていないのなら、テクスチャペイントモードに入る前に展開しておく必要があります。UV レイヤーに割り当てられた画像も、テクスチャペイントに使われます。つまり、次のいずれかを行う必要があります:

- 目的の画像が Image(画像)エディタウィンドウで表示されているときにモデルを展開する

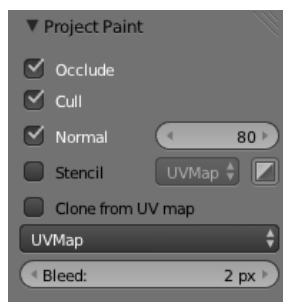
または

- 展開したら、編集モードのまま UV エディタウィンドウの画像を目的の画像に変更する

目的の画像が正方形でないなら、画像のアスペクト比に基いて展開できる最初の方法がよいでしょう。

Hints

(ヒント)



Project Paint パネル

ペイント時に既知の制限があります。

- 重なった UV はサポートされていません(テクスチャのバイクと同様です)
- 透視投影(Perspective)のビューでは、面が部分的にビューから隠れているとき、この面に色を塗ることができない制限があります。この場合ズームアウトするか、平行投影(Ortho)のビューを使ってください。
- 透視投影モードでビューからそれた法線を持ったローポリのオブジェクトの面にペイントするとき、ペイントはおそらく失敗します。回避策としてペイントパネルの **Normal(ノーマル)** オプションを無効にしてください。
主に立方体の側面を塗るときに起きます [T34665](#)

Texture Mapping

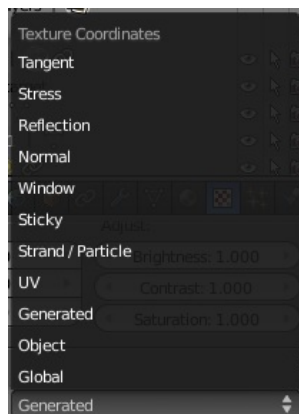
(テクスチャマッピング)

テクスチャには座標を対応付けて (map)、オブジェクトにどのように用いるのか決めなければなりません。マッピングは、テクスチャが最終的にどのようにオブジェクトを包み込むのか決定します。

例えば 2D 画像のテクスチャを円筒状のオブジェクトに巻きつけるような設定ができるでしょう。

Coordinates

(座標)



マッピングの座標メニュー

座標のマッピングは、座標の集合をマッピング処理のガイドに使って動作します。任意の座標を使えますが、通常はテクスチャを用いるオブジェクトのものを使います。

Global(グローバル)

シーンのグローバルな 3D 座標です。アニメーションにも役立ちます。オブジェクトを動かすと、テクスチャはそれを横切るように動きます。オブジェクトを空間のある場所で見せたり消したりするのに役立ちます。

Object(オブジェクト)

オブジェクトを座標の情報源として使います。Empty でよく使われ、オブジェクト位置に小さな画像を置く簡単な方法です。このオブジェクトもアニメーション可能で、オブジェクト表面をあちこち動かしたり、その中に移動できます。

Object

オブジェクト名を選びます。

Generated(生成)

オブジェクト固有の変形されていない座標です。デフォルトです。

UV(UV)

UV マッピングは 2D テクスチャを 3D の表面に非常に正確に対応付けます。メッシュの頂点それぞれが、展開して皮のように平らに広げることのできる UV 座標を持っています。UV 座標とは、独自のローカル座標系を持つ 2D 平面上で、対象の平面に対して行う写像(mapping)とほぼ同じものです。[UV マッピング](#)に見られるように、このマッピングは 2D 画像をテクスチャとして使う際に特に役立ちます。一組の UV 座標に複数のテクスチャを使うことができます。

Layer(レイヤー)

マッピングに使う UV レイヤーを選びます。

Strand/Particle(ストランド/パーティクル)

正規化した 1D のストランドテクスチャ座標か particle age(X)と trail position (Y)を使います。髪やパーティクルにテクスチャを用いるとき使ってください。

Sticky(?)

(訳注: この機能は少なくとも2012年には取り除かれました) Uses a mesh's sticky coordinates, which are a form of per-vertex UV coordinates. If you have made sticky coordinates first (in (usually) Camera View → Space → type Sticky → choose Add Sticky/Remove Sticky)、このテクスチャは camera view でレンダリング可能です (このため "[Camera Mapping \(eng\)](#)" と呼ばれていました)。

Window(ウィンドウ)

ウィンドウ座標でレンダリングします。2つのオブジェクトを混ぜ合わせる際に最適です。

Normal(法線/ノーマル)

表面の法線ベクトルの向きを座標として使います。見る角度によって変わる特殊な効果を作るときでも役立ちます。

Reflection(反射)

反射ベクトルの向きを座標として使います。反射マップを追加する際に役立ちます — Environment Mapping (環境マッピング)の際にこの入力が必要になるでしょう。

Stress(ストレス)

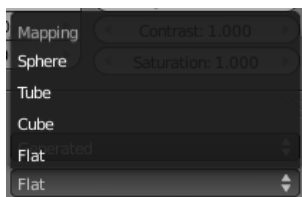
メッシュの元の座標と辺の長さの差を使います。メッシュがモディファイアで変形されているときなどに役立ちます。

Tangent(接線/タンジェント)

任意の接線ベクトルをテクスチャ座標として使います。

Projection

(投影)



投影メニュー

Flat(フラット)

単一の平面でもっともよい結果になります。球に用いると面白い効果を生み、球状マッピングした球に比べて結果は平らに見えます。マッピング平面にない面ではテクスチャの最後のピクセルが広がり、球や円柱では縞模様を作ります。

Cube(立方体)

オブジェクトの曲線部分が多すぎず有機的でもない場合にもっとも有益な結果が出ます(球の継ぎ目に注目してください)。

Tube(チューブ)

瓶に貼ったラベルのように、オブジェクトの周りにテクスチャをマップします。テクスチャは円柱表面に間延びします。もちろん瓶にラベルを貼ったり、丸みのあるオブジェクトにステッカーを貼ったりするのに役立ちます。ただしこれは円柱マッピングではないので、円柱の両端は定義されません。

Sphere(球)

球のマッピングに最適で、惑星やそれに似たオブジェクトにうってつけです。有機的なオブジェクトの作成に非常に役立つことがよくあります。円柱にも面白い効果を生みます。

Inheriting coordinates from the parent object

(親オブジェクトから継承される座標)

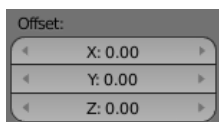
From Dupli(複製の親)

頂点、面、パーティクルから複製されたインスタンスが親のテクスチャ座標を継承します。

Todo: explanation

Coordinate Offset, Scaling and Transformation

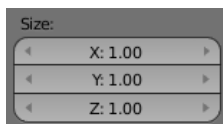
(座標のオフセット、拡大縮小とトランスフォーム)



Offset パネル

Offset(オフセット)

テクスチャ座標をオフセット分平行移動できます。各値を大きくするほどテクスチャが左上に向かって動きます。



Size パネル

Size(サイズ)

テクスチャ座標の拡大、縮小ができます。



マッピング 軸

X、Yおよび Z Mapping(マッピング)

このボタンを使って、テクスチャ自身の座標系と選択したマッピングシステム(Generated、UV、など)との間で各軸の対応を変えることができま

す。具体的には、ドロップダウンリストからテクスチャの軸(X/Y/Z/None)を選ぶと、その軸とマッピングシステム内のどの軸が対応するのか選ぶことができます。ここから次のことが言えます:

- 2D テクスチャ(例: 画像)については、Z のデータを持たないので最初の2つのドロップダウンリストだけが関係します。
- 最初のドロップダウンリスト(=テクスチャのX軸)を Y にし、2番目(テクスチャのY軸)を X にすると、2D 画像を90度回転できます。
- テクスチャ軸を一つも対応させない場合(ドロップダウンリストをすべて「None(なし)」にした場合)は、0次元(すなわち点やピクセル)のテクスチャが使われ、均一に塗りつぶされたテクスチャになります(Blender は この点の色を各軸上に繰り返します)。
- 1つのテクスチャ軸にだけ対応させると(ドロップダウンリストのいずれか2つが None であるなら)テクスチャの1次元(すなわちピクセルの線)だけが使われて「縞模様」のテクスチャになり、Blender は他の2軸に沿ってこの線を引き延ばします。
- 3D テクスチャ(=procedural なもの)についても同様で、1つの軸が None なら、関連する3つ目の軸に沿って図面が広げられ(「スライス」され)ます。

これらすべてを理解し、習得するのは少し難しいことです。幸いなことに、一部の特殊効果を除いては、頻繁に設定を変える必要はありません。ともかく、慣れるにはやってみるしかありません!

UV Mapping

(UV マッピング)

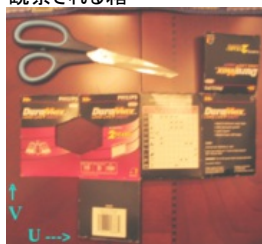
3次元のオブジェクトに2次元のテクスチャをマッピングする一番柔軟な方法は「UVマッピング」と呼ばれています。この処理では3次元(X、Y、Z)のメッシュを与えて平らな2次元(XとY...またはすぐにお目見えする「UとV」)の画像に展開します。要するに画像の各色がメッシュに配置され、メッシュの面の色として見えるようになります。UV テクスチャをオブジェクトに張って、プロシージャルなマテリアルやテクスチャにはできない現実感と、頂点ペイントを越えるディテールを与えてください。

UVs Explained

(UV の説明)



観察される箱



平らにマップされた箱

UVマッピングを理解するのに一番よい例えは、ボール紙の箱の裁断です。箱はシーンに追加した立方体と同じように3次元(3D)のオブジェクトです。

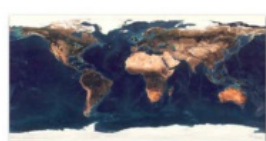
ハサミで箱を切ったり折ったりしたら、机の上に平らに広げることができるでしょう。机の上にある箱を見下ろしているとき、Uは左右方向を、Vは上下方向を指していると考えます。この画像はしたがって2次元(2D)です。この"テクスチャ空間の座標"を指すのに通常のXとYの代わりに(というのはこれはZと一緒に"3D空間"の参照にいつも使われているので)UとVを使います。箱を再び組み立てると、紙のUV位置は箱の(X,Y,Z)位置に変換されます。これが、コンピュータが2次元画像で3次元のオブジェクトを包むために行なっていることです。

UVの展開処理では、物体の面(この例では箱)をUV/画像エディタウインドウにある平らな画像に対応付ける(map)方法を、Blenderに正確に伝えます。方法は完全にあなたに任されています。(前の例の続きで、最初に机の上に箱を平らにして置いているものと考えてください。それを小さく切り、何とかして伸ばしたり縮めたりして、机の上にある他の写真の上になんらかの方法で並べます...)

Cartography Example

(地図製作の例)

地図製作者は数千年の間、この問題に取り掛かってきました。地図製作の例(下図)では世界全体の投影図を作っています。地図製作ではスペースシャトルに乗って地球の表面(球体)の写真を撮り、折り畳んで車の小物入れに入るような平らな地図を作ります。どのやり方も両極に向かって穴を'埋める'か、地図の輪郭を変えます:



メルカトル図法



モルワイデ図法



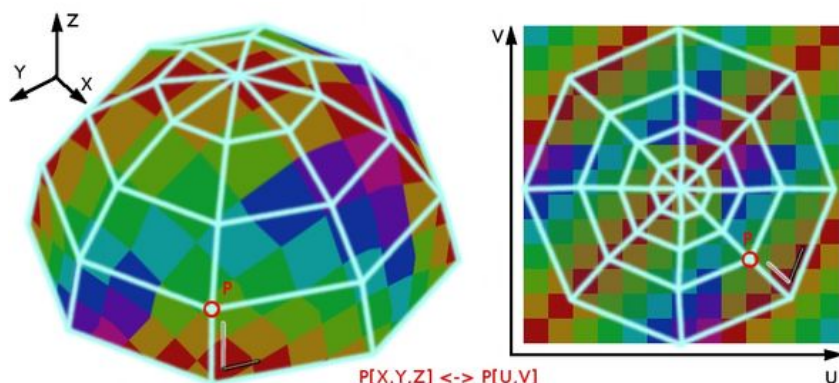
アルバース図法

これは球をUVマップするやり方の一例です。百ほどある一般的な投影手法にはそれぞれ長所と短所があります。どの方法を使うにしても、Blenderを使えば同じことをコンピュータでできます。

もっと複雑なモデル(上の世界地図で見たようなもの)では、面が「切れない」部分で問題が起きますが、代わりにこの部分は平らになるように引き延ばされます。これはUVマップを作りやすくしますが、最終的にマップされたテクスチャに歪みを生じることがあります(北極、南極に近い国や州は赤道上にあるものより平らな地図では広く見えます)。

Half-Sphere Example

(半球の例)



3D空間 (XYZ)と UV空間の比較 (クリックして拡大)

この画像では 3D空間で印をつけられた面の形状や大きさが、UV空間と違うことがよくわかります。

この違いは 2D平面上 (例えば UVマップ)に合わせて 3D空間の一部 (XYZ)を「引き延ばすこと」(専門的に言えばマッピング)によるものです。

3D オブジェクトが UVマップを持っていれば、3D 座標の X、Y、Zに加えて、オブジェクト上の各点是对応する Uと V 座標も持ちます (上の画像の P は 3Dオブジェクト上の点がどのように 2D画像上にマップされるのかを示す例です)。

The UV Editor

(UVエディタ)

UVマッピング用の機能について詳しくは [UV/画像エディタ](#) をご覧ください。

Advantages of UVs

(UV の利点)

以前の章で説明された procedural (生成式) テクスチャは、繰り返すことなしに 3Dオブジェクトに常に「フィット」するので便利です。一方でもっと複雑なものや自然なオブジェクトには不十分です。例えば人間の頭部の肌が procedural で生成されれば、必ず不自然に見えます。顔のしわや車のこすれ傷の位置はランダムではなく、モデルの形や使われ方に左右されます。手描きの絵や実写写真は、もっと制御が効いて現実味を与えます。具体的には本の表紙、タペストリー、じゅうたん、しみ、細かな装飾のあるものなどでは、UVテクスチャを使った面のすべてのピクセルをアーティストが制御できます。

UVマップは、モデルの各ポリゴンにテクスチャのどの部分を割り当てるかを表します。面の頂点はそれぞれ、マップされる画像の一部分を示す 2次元座標に割り当てられます。この 2次元座標は UV と呼ばれます (3次元の XYZ 座標と対照的です)。UVマップの生成操作は 2次元の平面にメッシュを広げるように見えるので「unwrap (包装を解く)」とも言います。

Blender は単純な 3D モデルのほとんどに対して、簡単に使える自動展開アルゴリズムを備えています。複雑な 3D モデルは、通常の立方体、円筒状、球状マッピングでは普通は間に合いません。一様で正確な投影のために、シーム (seam、縫合線) を使って UV マッピングを誘導してください。シームは人の頭や動物のような不規則で複雑な形にテクスチャを張るのに使えます。こうしたテクスチャは手描きの画像であることが多く、Gimp や Photoshop、あるいは好きなペイントアプリケーションで作ります。

ゲーム

UVマッピングは Blender ゲームエンジンや他のゲームにとっても欠かせないものです。モデルに使うテクスチャ方式のデファクトスタンダードで、ゲームで目にする大半のモデルは UVマッピングが施されています。

UV Mapping a Mesh

(メッシュの UV マッピング)

最初のステップはメッシュの展開 (unwrap) です。メッシュが形状を表すのに必要な面をすべてそろえていると思えたら、展開を始めます。展開済みのモデルに対して面の追加や分割を行うと、blender は新たにできた面を追加してくれますが、マッピングや編集の追加作業が必要になるかもしれません。この方法で、ジオメトリの追加変更をガイドするために UV テクスチャ画像を使えます。

この章では UV のマッピング技法を扱います。次の章では [UV の編集](#) を、さらに次の章は [UV レイアウトの管理](#) 手法を、続いて [画像の適用](#) をそれぞれ扱います。

About UVs

(UV について)

UV マップにある点はすべて、メッシュの頂点に対応しています。UV マップの点を繋ぐ線はメッシュの辺に、UV マップの面はメッシュの面に対応しています。

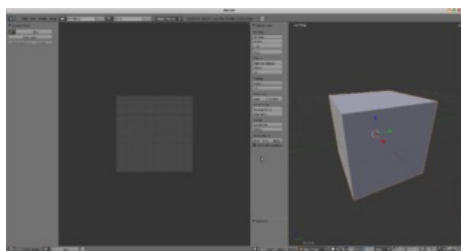
メッシュの各面はたくさんの UV テクスチャを持てます。UV テクスチャにはそれぞれ、別の画像が割り当てられます。UV/画像エディタで面を UV テクスチャに展開すると、メッシュの各面が自動的に 4つの UV 座標に割り当てられます。この座標は画像やテクスチャを面にマップする方法を定義します。これは2次元座標で、3次元の XYZ 座標と見分けやすいように U と V の名前で座標を呼んだのが名前の由来です。この座標はレンダリングやリアルタイムの OpenGL 表示にも使われます。

blender の面はそれぞれ別の画像にリンクすることができます。UV 座標はこの画像が面にどのようにマップされるのかを定義します。こうしてこの画像は、レンダリングやリアルタイム表示できるようになります。3D ウィンドウは、アクティブなメッシュオブジェクトの UV 座標を変更したり、画像を割り当てたりできるように、「面選択モード」にします。これで、面が複数の UV テクスチャに加わることができます。キャラクターの髪の毛の生え際にあたる面は、顔の UV テクスチャと頭皮/髪の毛の UV テクスチャの両方にあるかもしれません。

これについては次の章でもっと詳しく述べます。

Getting Started

(始めましょう)



UV Editing スクリーンレイアウト

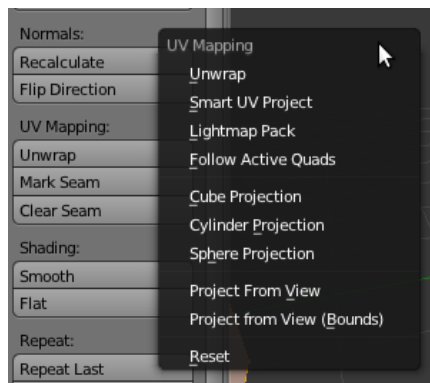
デフォルトでは、メッシュには UV がありません。始めに面をマップする必要があり、そうすれば [それを編集できます](#)。モデルの展開作業は 3D ビューウィンドウの編集モードで行います。この作業で、UV/画像エディタウィンドウに1つ以上の UV Island (島状のまとまり)ができます。

始めるには、画面最上部にある Info (情報) ウィンドウヘッダーの選択リストから UV Editing (UV編集) の [スクリーンレイアウト](#) を選んでください。これで UV/画像エディタウィンドウ (Shift+F10) の表示パネルと、3D ウィンドウ (Shift+F5) の描画域が並びます。

編集モードに入ってください。展開作業はすべて編集モードで行います。頂点、面、辺選択モードを使えます。

Workflow

(作業手順)



展開方式の選択

展開の手順はわかりやすいのですが、展開の結果に大きく影響するような選択肢が山ほどあります。選択肢の裏にある意味を理解すると、より効率的な展開ができるようになるでしょう。手順は次の通りです:

1. 必要なら seam (シーム、縫合線) をつけます
2. メッシュの構成要素をすべて選択

- UV Unwrap メニューからUV マッピング方式を選びます
- 展開の設定を調整します
- テスト画像を追加して歪みが生じないか確かめます。[画像の適用 \(en\)](#) をご覧ください。
- UV エディタでUVを調整します。[UV の編集](#) をご覧ください。

Mapping Types

(マッピングの種類)

blender には UV のマッピング方法が複数用意されています。シンプルな投影方式は、メッシュ表面のすべての点の位置を点/軸/平面方向に補間して、3次元空間を2次元空間に配置する換算式を使います。高度な方式ほど複雑なモデルに使うことができ、専門的な用途を持ちます。

基本:

[Cube\(キューブ投影\)](#)

メッシュを立方体の各面にマップしてから、立方体を展開図にします

[Sphere\(球状投影\)](#)

UV を球体に投影します。眼球や天体など、球か球に近い形にだけ役立ちます

[Cylinder\(円筒状投影\)](#)

UV を円筒状の面に投影します

[Project from View\(視点から投影\)](#)

3D ビューポートの現在のビューに見えているまま平らにします

高度:

[Unwrap\(展開\)](#)

生物の形状に役立ちます。seam に沿って切り分け、メッシュを平らにします

[Smart UV Project\(スマートUV投影\)](#)

メッシュを角度を基準にして island に分割します

[Lightmap Pack\(ライトマップパック\)](#)

面ごとに切り離して UV グリッドに詰めます

[Follow Active Quads\(アクティブ四角形面に追従\)](#)

アクティブな四角形から始めて、そこに環状に繋がる面に沿って uv をたどりませ

UV はリセットすることもできます。各面が UVグリッドを埋めるように配置され、どの面も同じマッピングになります。

タイル状に敷き詰めることのできる画像を使ったとすれば、メッシュ表面は、面の形にあわせて歪まされたこの画像の、滑らかな繰り返しで覆われるでしょう。この展開オプションは、マップをリセットし、すべての展開を取り消すときに使ってください(開始時の状態に戻ります)。

Basic Mapping

(基本のマッピング)

オブジェクトの基本となるジオメトリとその見え方に基いて、Mesh » UV Unwrap » Cube(キューブ)、Cylinder(円筒)、Sphere(球) の方式は最初うまくフィットしたとおりに面を広げようとします。ここで、3D ウィンドウでの見え方は特に重要です。また、オブジェクトを包むように立方体のサイズや円筒の半径の設定(ツールシェルフ内)も調整しておくべきです(blender単位(BU)で)。

次の設定は キューブ、円筒状、球状の投影マッピングでよく使われます。

Correct Aspect(アスペクト比の補正)

画像のアスペクト比を踏まえて UV を配置します。画像がすでに正方形でないテクスチャ空間にマップされていれば、これを前提に正しく表示されるように配置をひずませ、投影します。

Clip to Bounds(境界にクリッピング)

0 から 1 の範囲外にある UV を一番近い UV 空間の境界に移動して、範囲内におさめます。

Scale to Bounds(境界に拡大縮小)

0 から 1 の範囲より UV マップが大きければ、マップ全体が入るように縮小します。

Cube

(キューブ投影)

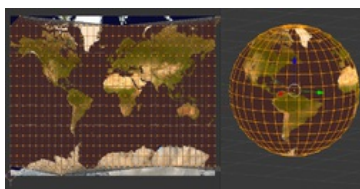
キューブマッピングはメッシュを6つのわかれた面に投影し、6つの UV Island を作ります。この島は UV エディタで重なって表示されますが、動かせませす。[UV の編集](#) をご覧ください。

Cube Size(立方体サイズ)

投影に使う立方体のサイズを設定します。

Cylinder and Sphere

(円筒状/球状投影)



球状投影でメルカトルの画像を使う

円筒状投影と球状投影は同じ設定を持ちます。相違点は円筒状投影が UV を円筒に対して投影するのに対し、球状投影は球の曲率を考慮し、緯度線が等間隔になります。

通常は円筒の展開を、縦に切込みを入れて平らに広げたかのように行うため、blender はその筒が直立していて、ビューが縦方向になることを求めます。ビューが変われば UV マップへの筒の投影も変わり、もし使っていれば画像を歪ませます。ただ、計算の行われる方向は手動で設定できます。球状投影でも同じ考え方ができます:

最初の地図製作者が世界地図を作った方法を思い出してください。blender でも、別の視点から球を展開すれば同じことを行えます。通常は球の展開のために、上下にある極から球を見ます。展開すると、blender が再現しているのはメルカトル投影図法です。あなたの側にある赤道上の点が、画像の中央に置かれるでしょう。この極視点 (polar view) は、非常に変わってはいるがよくある投影図を作ります。UV 画像としてメルカトル透視図法の世界地図を使うと、球上にマップされた素敵な惑星が作れます。

Direction(方向)

View on Poles(ビューを極に)

一番上(極)に視点を置き、視点からまっすぐ降りる軸を使います。

View on Equator(ビューを赤道に)

赤道を見る視点で、縦方向の軸を使います

Align to Object(オブジェクトで揃える)

軸の計算にオブジェクトのトランスフォーム(位置、回転、拡大縮小)を使います

Align(整列)

上になる軸を選びます

Polar ZX(極のZX)

極 (polar) 0 は x 軸上にあります

Polar ZY(極のZY)

極 (polar) 0 は y 軸上にあります

Radius(半径)

使用する円筒の半径です

Project From View

(視点から投影)

3D ウィンドウの Face->Unwrap UVs->Project from View(視点から投影/プロジェクション) オプションは、選択中の 3D ウィンドウに見えている通りに面をマップします。あなたに透視能力があれば、あるいはメッシュをつぶしてパンケーキのように平らにし、UV マップ上に置いたとすれば、こんな感じに見えるでしょう。モデリングしたオブジェクトの UV テクスチャに実物の写真を使うときは、このオプションを選んでください。モデルのあなたから遠い部分では、多少画像が引き伸ばされるでしょう。

Project from View (Bounds)(視点から投影(バウンド)) も同じ動作ですが、UV 空間の境界線に UV のサイズを合わせます。

Resetting UVs

(UV のリセット)

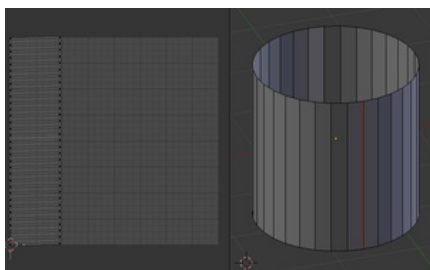
前に述べたように、3D ウィンドウの Face->Unwrap->Reset(リセット) は、選択面すべてを画像の同じ領域にマップします。オブジェクト(たとえば立方体)のすべての面を同じ画像にマップするには、立方体のすべての面を選んでからリセットメニューを使って展開します。

Advanced Mapping

(高度なマッピング)

Unwrapping Using Seams

(Seam を使った展開)



円柱の単純な seam

多くの場合、立方体、円筒形、球状、または best fit (訳注: おそらく「展開(Unwrap)」を指しています)の展開方式を使えば、良い UV レイアウトができます。しかしもっと複雑なメッシュ、とりわけ凹凸の激しいものには、上述のいずれかの展開処理の範囲制限と誘導用に、**seam** (シーム、縫合線) の定義が必要になるかもしれません。

縫い物と同じで、seam は画像/布地が縫い合わされる境界線です。展開処理を行ったメッシュは seam で切り開かれます。オレンジの皮をむいたり動物の皮を剥ぐようなものだと考えてみてください。皮に連続した切れ目を入れて、皮を剥ぎます。それから皮を多少引き延ばしながら平らにできます。この切れ目は seam と同じです。

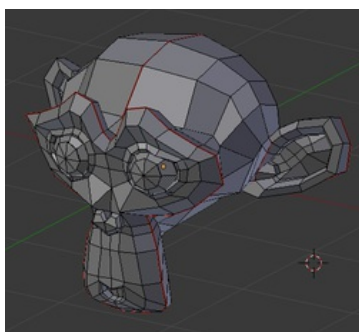
この方法を使うとき、引き伸ばしほどの程度あるか気を配る必要があります。seam が多いほど引き延ばしは少なくなりますが、これは大抵テクスチャ製作で問題になります。引き延ばしを最少にしつつ seam をできるだけ少なくするのが良い考えです。見えない部分に seam を隠すようにしましょう。3D ペイントを使う製作では、投影ペイントは seam を扱いやすいのでさしたる問題になりませんが、対照的に 2D テクスチャリングでは、別の UV island の境界線同士を合わせるのが難しくなります。

処理手順は次の通りです:

1. seam を作ります。編集モードで seam になる辺を選択し「シームをつける (Mark Seam)」を使います。
2. 展開 (unwrap) します。
3. seam を調整します。これを繰り返します。
4. 手動で UV を調整します。次章の UV 編集をご覧ください。

Marking Seams

(シームをつける)



seam のつけられた Suzanne

辺を seam に追加するには、単に辺を選択して Ctrl+E Mark Seam (シームをつける) を実行します。辺を seam から外すには、選択して Ctrl+E Clear Seam (シームのクリア) を実行します。

右の例では、seam を隠すために円柱の一番背面にある辺を seam に選び、デフォルトの展開方式を使いました。UV/Image Editor ウィンドウで、まるでさきで布地の縫い目を切って広げたように、すべての面がうまく展開されているのがわかります。

seam をつけるとき、面選択モードで Select (選択) -> Linked Faces (リンク) や Ctrl+L を使って作業状況を確認できます。このメニューは選択中の面に接続している面を seam を越えない範囲ですべて選択します。もし作りたい seam より外側の面が選択されていれば、seam がどこかで途絶えているのがわかります。ただし、seam が続くであろう部分を blender が解決できる限りは、seam が途絶えていても構いません。

動物の皮を剥ぐ方法が何通りもあるように、seam をつけるべき場所を決める方法もたくさんあります。普通は次のように考えてみてください。片手に (動物タイプの) オブジェクトを、片手にはさみを持っていて、オブジェクトをなるべく少ない回数切り開いてテーブルに広げようと考えています。体の前面と背面をわけたいので、耳の外側を通る seam をつけます。目は他とは別のメッシュなので、自動的にそれだけで展開されます。後頭部には縦に seam を入れて、頭の左右両側をそれぞれ平らに広げるようにします。

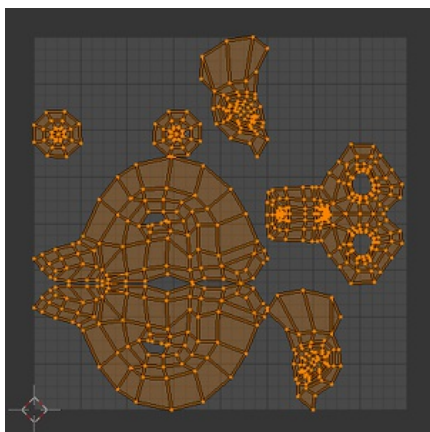
seam は、展開される面を制限する使い方もできます。例えば、頭のテクスチャリングをしているとき、頭頂と後頭部の頭皮は髪に覆われるため、実際にはテクスチャを必要としません。ですから、髪の生え際に沿って seam を定義します。こうすると、顔の正面を選択して Linked Faces (選択 -> リンク) を選択すれば、髪の生え際より先は選択されず、その部分の頭皮は展開されません。

頭や体のような左右対称のものを展開するときは、対称軸に沿って seam をつけます。例えば、正面図の真ん中で頭や体全体がわかるようにします。展開すると両半文を同じテクスチャ空間に重ね合わせることができ、右手の画像は左手の画像と共有され、顔の右側は左側と共有され、といった具合になります。

最後に、「どこでも何にでも完璧に動作する展開方法」を考えだす必要は「ない」ことを覚えておいてください。後述しますが、メッシュの部分部分に応じた別の方法を使うことで、簡単に複数の UV 展開を持つことができます。

Unwrap

(展開)



Suzanne の展開結果

展開すべき面を 3Dビューですべて選択することから始めましょう。面が選択できたら、次は展開です。3D ビューで Mesh(メッシュ) » UV Unwrap (UV 展開) または U を選択し、さらに Unwrap (展開) を選びます。

UV/画像エディタウィンドウからも、UVs(UV) » Unwrap (展開) または E を使って同じことができます。この方法は面すべてを展開し、それまでの作業をリセットします。

Face (面) » Unwrap (展開) » Unwrap (展開) オプションは、選択された面にあるすべての seam を踏まえ、面の接続の仕方を見てオブジェクトの各面が画像内にちょうどおさまるように展開します。可能なら、選択された面にはそれぞれ画像の別の領域が与えられ、どの面も他の面の「下に入り込む」ことがないようにします。もしオブジェクトのすべての面が選択されていれば、どの面も画像のどこかにマップされます。

blender は 展開処理の計算方式を2つ持ちます。3D ビューの ツールシェルフにある設定パネルで選べます。

Angle Based (アングルベース)

この方式はメッシュをうまく2次元で表現します。

Conformal (等角)

LSCM (Least Squared Conformal Mapping、最小二乗等角写像) を使います。通常はアングルベースに比べて正確性に劣る UV マッピングになりますが、オブジェクトが単純なほどうまく働きます。

Fill Holes (穴を埋める)

Fill Holes (穴を埋める) を使うと UV 領域内で重なりが発生を防ぎ、穴をうまく表現します。

Correct Aspect (アスペクト比の補正)

画像のアスペクト比を考慮した UV マッピングです。

Use Subsurf Modifier (細分割曲面モディファイアを使用)

subsurf モディファイア適用後の頂点位置を考慮した UV マッピングです。

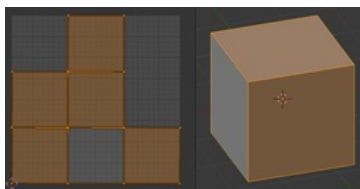
Margin (余白)

UV island の間隔

後々、マッピングを理解するために極めて重要になること: 面が使う UV 画像テクスチャは画像全体である必要はなく、一部で構いません。また、同じ画像の一部を複数の面で共有することができます。面は画像全体のごく一部にマップできます。

Smart UV Project

(スマートUV投影)



立方体にスマートUV投影を使う

スマートUV投影 (以前は Archimapper と呼ばれていました) を使うと、メッシュの角度変化に基づいた seam の自動生成の方法を、細かく制御できます。機械的なオブジェクトや建築物のような単純かつ入り組んだジオメトリ形状に適しています。

この機能はオブジェクトの形状、つまり選択されている面と面同士の関係を調べ、この情報と指定した条件に基づいて UV マップを作ります。

右の例ではスマートマップパーは、6面どれもが立方体の面と同じ四角形になるように、3面を上3面を下にした、きれいな並びで立方体をマップしています。

複雑度の高い機械的なオブジェクトに対して、このツールは非常に論理的でわかりやすい UV レイアウトを手早く簡単に作れます。

ツールシェルフの設定パネルでメッシュの展開方法を細かく調整できます:

Angle Limit (角度制限)

面のグループ分けを調節します: 制限値を上げるほどたくさんの小さなグループができ、歪みが減ります。制限値を下げるほど歪みの増加と引き換えにグループの数が減ります。

Island Margin(島の余白)

UV island の密度を決めます。数値が高いほど island の間隔が広がります。

Area Weight(エリアウェイト)

面ごとの投影ベクトルに、広いものほど重みづけします。

Lightmap

(ライトマップ)

ライトマップパック(Lightmap Pack)はメッシュの面や選択面を、1つずつ UV 境界線内に詰め込みます。ライトマップは初めゲーム開発で使われ、UV 空間を可能な限り多く利用するために、ライティング情報をテクスチャマップに焼きこむために使われていました。複数のメッシュに同時に作用させることもできます。ツールシェルフには複数のオプションがあります:

1つのメッシュで作業していれば、ツールのマップ対象を Selected Faces(選択面) だけにすることも All Faces(全ての面) にすることもできます。

Selected Mesh Object(選択したメッシュオブジェクト) オプションは複数のメッシュに対して使えます。使うには、オブジェクトモードで複数のメッシュオブジェクトを選択してから編集モードに入り、ツールを実行してください。

Share Tex Space(テクスチャ空間の共有)

複数のメッシュをマッピングするときに便利です。オブジェクトのすべての面を、重ならないように UV 領域内にフィットさせようとしています。

New UV Layer(新規UVレイヤー)

複数メッシュをマッピングするとき、このオプションはメッシュごとに新しい UV レイヤーを作ります。[レイアウトの管理](#) をご覧ください。

New Image(新規画像)

メッシュごとに新規画像を割り当てます。テクスチャ空間の共有 が有効なら全てのメッシュに1つだけ画像を割り当てます。

Image Size(画像サイズ)

新規画像のサイズを指定します。

Pack Quality(パック品質)

複雑な箱詰めの前に、事前詰め込みを行います。

Margin(余白)

UV island の密度を設定します。数値が高いほど island の間隔が広がります。

Follow Active Quads

(アクティブな四角形面に追従)

Face->Unwrap->Follow Active Quads(アクティブな四角形面に追従) は選択面から連続する面ループを(面の形が不揃いであっても)たどって、それらの面を配置します。なお、画像サイズは考慮されないで、画像領域にあわせて多少縮小する必要があるかもしれません。

Edge Length Mode(辺の長さモード):

Even(均一)

UV をすべて均等に置きます

Length(長さ)

各ループのUV辺の長さの平均スペース

注意してほしいのは、出発点となるアクティブな四角形は UV 空間のもので、3D 空間のものではないということです。きれいな 90度に展開するには、「アクティブな四角形面に追従」を使う前にアクティブな四角形が UV 空間にあることを確認してください。


UVレイアウトの管理

UVマップの編集を終えた後、同じオブジェクトにさらにマップを作ったり、他のメッシュに UVマップを転送する必要があるかもしれません。

UVの転送

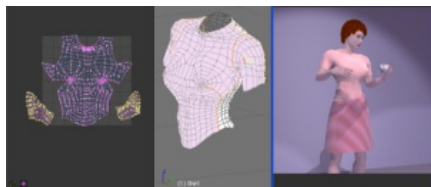
二つのメッシュが同じジオメトリ/頂点順序を持っていれば、一方からもう一方へ、UVマップをコピーできます。これは、たとえば UVマップを、以前に保存したモデルの手を触れる前のものから作り直すようなときに役立ちます。

作業手順

- RMB  で対象メッシュ (UVマップのコピー先) を選びます
- ⇧ Shift でコピーしたい UVマップを持つメッシュを選びます
- Object (オブジェクト) » Make Links... (リンク作成) » Transfer UV Maps (UVマップのコピー) (Ctrl)L

これで、対象メッシュの UVマップは元のメッシュと一致します。

複数の UVマップ



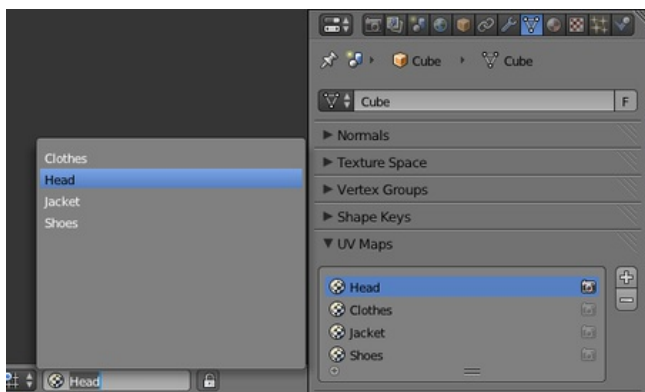
複数の UVテクスチャを持つメッシュ

メッシュごとの UVマップの数はひとつに制限されません。新たな UVテクスチャを作れば、メッシュの一部に対して複数の UVマップを持つことができます。最初の UVテクスチャは、UV 面選択モードで面を選択したときに自動的に作られます。自分でさらに UVテクスチャを作るには、プロパティウィンドウのテクスチャコンテキストを開き、テクスチャスタックパネル内のテクスチャデータブロックで、「UVテクスチャ」の隣の「+」(New/新規) ボタンをクリックして、メッシュの別の部分を展開します。この操作より前に展開した面は以前の UVテクスチャに付属したままですが、これ以降に展開した面は新たな UVテクスチャに付属します。なお、同じ面を2度以上(それぞれ別の UVテクスチャに)展開した場合には、面の色はそれらの UVテクスチャをアルファ合成で重ね合わせたものになります。

右の例はブラウスのメッシュです。中央に示すように UV 面選択モードで、メッシュは通常のブラウスと同じように縫い合わされています。cut pattern を作りたかったので、ブラウスの前面を展開して、基本的な回転とスケールを行なって UV/画像エディタの中央に配置しました。それを一旦脇に移動させ、左右の袖を展開して、それぞれに回転とスケールを施しました。試しに衣服の断片ごとに面を1つ選択して、3D ビューで Select (選択) » Linked Face (リンク) を選ぶと、UV/画像エディタはすべての断片を表示します。こうしてすべての断片をこの UVマップ用に手がけることができます。この例では3つの断片すべてがペイント用の画像領域に移動されています。1平方ヤードの生地がうまく収まっていることがわかります。

テクスチャコンテキストのテクスチャデータブロックにある New (新規) ボタンをクリックして、別の UVマップを作りました。それから袖の背面とブラウスの背面用にこの工程を繰り返しました。生地に色をつけるため、前面用に1枚、背面用に1枚の、合計2枚の画像を使っています。この例では、1つの面が1つ目のテクスチャに、もう1つの面が2つ目のテクスチャにマップされています。

UVテクスチャの一覧



UV Maps パネル(右図)は、このメッシュ用に作られた UVテクスチャマップの一覧を表示します。今後の展開操作のために、プレースホルダーとして新しいものを作れます。

新たな UVテクスチャを追加するには + ボタンを、既存のものを削除するには - ボタンをクリックします。メッシュの UVマップを削除すると、メッシュに結びつけられているすべての展開作業が失われます。気をつけてクリックしてください。ご注意ください。

マップごとに選択ボタンがあります。カメラアイコンをクリックして、UV テクスチャのレンダリングを有効化します。UVテクスチャを選択して Name ボックス内のテキストを変えると、名前を変更できます。選択されたマップは UV/画像エディタウィンドウに表示されます。図はあるキャラクター用に作られた数枚の UVマップを示しており、頭用のマップが選択されています。

各テクスチャは指定した UV テクスチャにマップできることに注意してください。テクスチャパネルの [マッピング](#) セクションをご覧ください。

Editing UVs

(UV の編集)

展開し終わったら、UVマップをテクスチャ付けや描き込みがしやすい配置に変える必要があるでしょう。編集作業では次を目指します:

- 断片 (UVマップ) をいくつかつなぎなおす
- 画像の無駄な空きを最小限にする
- 詳細を描き込みたい箇所の '面' を拡大する
- 引き延ばされている '面' をリサイズ/拡大する
- 粒子が粗すぎ、詳細度を持ちすぎている '面' を縮小する

無駄な空きを最小限に抑えようと、ピクセル (画像を構成する小さな点) の大半を、細部の描き込みを最大限反映させて UV テクスチャにおさめることができます。UV 面はピクセルと同程度に小さくすることも、画像全体と同程度に大きくすることもできます。普通は最初に大きな調整を行ってから、レイアウトを微調整することになるでしょう。

Selecting UVs

(UV の選択)

選択ツールは 選択メニュー (Select Menu)、ヘッダーバー、もしくは以下のショートカットで使えます:

矩形選択 (Border Select) ; B

矩形の範囲選択で UV 座標を選びます。

全てを選択/選択解除 (Select/Deselect All) ; A

すべての UV 座標を選択または非選択します。最初に展開処理を行ったときは、すべての UV を選択して回転、スケール、移動を行うことになるでしょう。

繋がった UV (Linked UVs) ; CtrlL

同じ UV マップの一部であるすべての UV を選択します。思い出してほしいのは、メッシュはサブメッシュと、メッシュの一部を縫い合わせたものでできていて、1枚の布地に例えられることです。繋がった UV (Linked UVs) は、3Dビューのコマンドと似た動作をします。現在選択中の UV に '接続した' UV をすべて選択します。

ピン留めされた UV (Pinned UVs) ; ⇧ ShiftP

UV をピン留めして、複数回の展開操作をしても動かないようにできます。このメニュー項目はピン留めした UV 全てを選択します。[ピン留め](#)をご覧ください。

ピン留めの矩形選択 (Border Select Pinned) ; ⇧ ShiftB

矩形の範囲選択を使って、ピン留めされた UV 座標だけを選択します。

選択の切り離し (Unlink Selection) ; AltL

選択した UV をマップから切り離します。完全に選択された面に属する UV だけが、このコマンドのあとに選択されたまま残ります。名前が暗示するように、面を切り離して別の場所に移動するときに特に役立ちます。

Selection Modes

(選択モード)

ヘッダーにある 選択の同期 (Sync Selection) ボタンをオンにすると、3Dビューの選択成分を UV エディタ上の対応する要素と同期させます。デフォルトではオフです。この2つのモードは、UV エディタで要素を動かしたときの結果がかなり異なります。

選択の同期がオフ のとき: 選択面だけが UV エディタに表示され、次の選択モードが利用可能になります:

- 頂点 (Vertex)
 - 個々の頂点を選択
- 辺 (Edge)
 - 辺を選択
- 面 (Face)
 - 面を選択
- 島 (Island)
 - 隣接した面のグループを選択

このモードでは 吸着選択モード (Sticky Selection Mode) のメニューが使えます。これは UV の選択方法を決めます:

頂点の共有 (Shared Vertex)

メッシュの頂点を共有する UV を、UV マップ上の位置が離れているものもすべて選択します。

位置の共有 (Shared Location)

同じ UV 位置にありメッシュの頂点を共有する UV を選択します。このモードのデフォルトで、大抵の場合最もよく働きます。

無効化 (Disabled)

吸着選択を無効化します。このモードで UV を動かすと各面がそれぞれ UV を持ち、切り離すことができます。

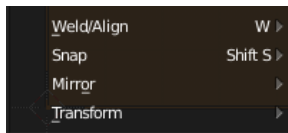
選択の同期 (Sync Selection) が オン のとき次を選べます:

- 頂点 (Vertex)
- 辺 (Edge)
- 面 (Face)

このモードでは選択操作の挙動が異なります。UV や 辺を選択するとき、上述の 頂点の共有 (Shared Vertex) モードのように動作します。面を選択するとき、上述の 吸着選択の無効化 (Disabled Stick Selection) のように動作します。

Transforming UVs

(UV のトランスフォーム)

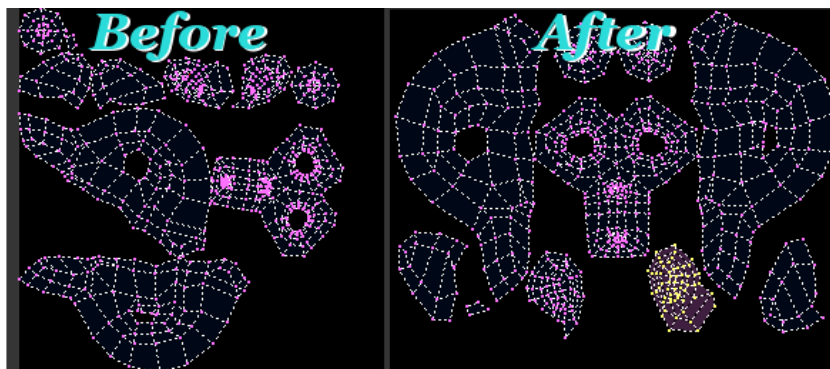


UV トランスフォームメニュー

UV には次のトランスフォーム操作を行えます:

- 移動 (Translated) G
- 回転 (Rotated) R
- 拡大縮小 (Scaled) S


また、編集モードと同じように H で隠し、AltH で再表示できます。



繋がったものの選択、移動、回転を使った配置変更


Axis Locking

(軸のロック)

トランスフォームツールのあとに X や Y を押すと、トランスフォームがこの軸にロックされます。また、MMB  を押し続けると、移動を X か Y 軸に制限します。

Pivot Points

(ピボット)

UVエディタには 2D カーソルがあります。カーソル位置は UVエディタ内で LMB  クリックすると変えることができます。プロパティパネルから手動で位置を調整することもできます。デフォルトでは左下隅から始まる 0 から 256 の範囲を持ちます。座標 (Coordinates) の下の 正規化 (Normalized) を有効にすると、範囲は 0 から 1 に変わります。

スナップ (Snap) の下にある UVメニュー (UVs Menu) を選ぶと、2Dカーソルは最も近いピクセルか選択中の要素にスナップします。

ピボットは次のうちいずれかに変更できます:

- バウンディングボックスの中心 (Bounding Box Center)
- 中点 (Median Point)
- 2Dカーソル (2D Cursor Location)

Proportional Editing

(プロポーショナル編集)

UV編集でプロポーショナル編集を使えます。コントロールは 3Dビューのものと同じです。完全な説明は [3D のプロポーショナル編集](#) をご覧ください。

Snapping

(スナップ)

UV のスナップは [3D のスナップ](#) と似ていますが、UV に対するスナップだけが働きます。一方 UVメニュー (UVs Menu) にある (ピクセルにスナップ

(Snap to Pixels) オプションは、もし画像がロードされていれば、UV を画像のピクセルにスナップさせます。

さらなるツールが スナップ(Snap) サブメニューの下にある UV のメニュー(UVs Menu) にあります:

ピクセルにスナップ(Snap Pixels)

選択要素を近くのピクセルに移動します

カーソルにスナップ(Snap to Cursor)

選択要素を 2D カーソルの場所に移動します

隣接した非選択物(Snap to Adjacent Unselected)

選択要素を隣接した非選択の要素まで動かします

Weld and Align

(溶接と整列)

溶接(Weld) ツール W1

選択した UV をその平均位置に移動します。

整列(Align) ツール W2、W3、W4

選択した UV を X軸、Y軸、または自動的に選ばれた軸に沿って一列に並べます。

Mirror

(ミラー)

要素が Y軸か X軸を中心に対象位置に置かれます。UV メニュー内の スナップ(Snap) サブメニューから X軸(Mirror X) と Y軸(Mirror Y) を選べます。

ホットキー CtrlM に続けて X か Y を押すか、MMB  を押し続けてミラーする方向にドラッグしても同じ操作ができます。

Stitch

(スティッチ/縫い合わせ)

スティッチ(Stitch)、V は選択している UV のうち頂点を共有する UV を繋げます。ツール設定の 制限する(Use Limit) を有効化して 制限距離(Limit Distance) を調整すると距離によってスティッチを制限することができます。

Blender 2.66 以降では、辺を基準にしたスティッチを使えます。これは UV 頂点をスティッチしようとする時に現れるあいまいさを解消してくれます。一つの頂点に属する UV 頂点を、多くの UV 島(Island)の一部にすることができますが、UV 辺は通常二つの UV 島でのみ共有されます。

スティッチの初期モードはツール開始時の選択モードに依存し、頂点選択時は頂点ベース、それ以外は辺ベースになります。二つのモードを自分で切り替えるには、スティッチを開始してから⇐ Tabを押します。

Minimize Stretch

(最小のストレッチ)

最小のストレッチ(Minimize Stretch) ツール CtrlV は 角度を最小限にすることで UV の伸縮を抑えます。これは本質的に UV をほぐします。

Reverse and Rotate UVs

(面のミラーと UV の回転)

UV テクスチャの方向がそれぞれの面とどう関係していたか覚えていませんか? 例えば画像が上下さかさまであったり、横に寝ていたりします。そういうときは Mesh(絵メッシュ) » Face(面) » Rotate UVs(UVを回転) (3D ウィンドウで面選択モードのとき)メニューを使って UV を面ごとに 90度ずつ回転させてください。

Uvs(UV) » Mirror(ミラー) は画像をフライパンに乗せたパンケーキのように裏返します。面ごとに UV を鏡映させて、'反転させた' 画像を表示します。

Pinning

(ピン留め)

モデルの展開処理を行うとき、UV レアウトの一部を同じ形のまま残したり、同じ場所に置いたままにしたりするために、特定の UV をロックできると便利ことがあります。

ピン留めは UV を選択して Uvs メニューからピン留め(Pin) を選ぶか、ショートカット P で行います。UV のピン留め解除はショートカット AltP でできます。

ピン留めが最も効果的なのは、「展開(unwrap)」の UVマッピング方式で生物のオブジェクトを展開する場合です。ひとつの例として左右対称のオブジェクトを [Mirror モディファイア](#) を使ってモデリングしているところを考えます。ミラーの軸上にある UV は、鏡映された反対側でも共有されるかもしれませんが。中心線に一致する UV をピン留めできれば、これらを X 軸上に並べることができ、その場に留まらせることができます。

ピン留めはまた、ライブ展開(Live Unwrap) ツールを使うとき大変役立ちます。ライブ展開をオンにして2つ以上の UV をピン留めすると、ピン留めた UV をドラッグすることでインタラクティブにモデルを展開できます。UV 島のある形や領域にあわせるときにも役立ちます。


Optimizing the UV Layout

(UV レアウトの最適化)

展開をおそらくは seam を使って行ったとき、UV レイアウトはとても散らかった雑然としたものになるかもしれません。次のタスクを進める必要があるでしょう: UVマッピングの方向、UVマップの整頓、いくつかのマップの縫い合わせ。

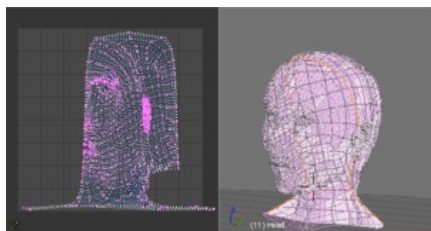
次の手順は、展開の工程で作った UVレイアウトを扱う作業です。展開済みのモデルに面の追加や面の分割を行うと、Blender はこうした新しい面を UVマップに追加してくれます。このようにして UV テクスチャ画像を、ジオメトリのさらなる変更のガイドとして使えます。

整頓作業にはウィンドウ全体を使えますが、グリッド内にある UV座標だけが画像にマップされることは忘れないでください。したがって、整頓の間一部を脇に移動しておくこともできます。また UV 展開はそれぞれが、座標の独自のつながりのセットになります。

他のものに重ねあわせることができ、そうするとオニオンスキンになります(下にあるものが透けて見えます)。そのうちの1つだけをつかむには、UV座標の1つを RMB  選択して、選択 (Select) » リンク (Linked UVs) (CtrlL) を使ってつながっている UV を選択してください。この場合範囲選択は 両方の UV が選択されてしまうため使えません。

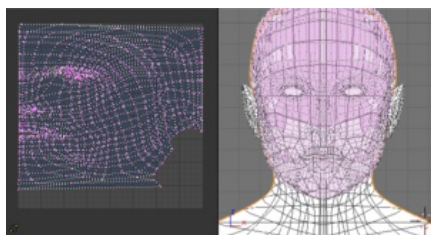
Combining UV Maps

(UV マップの結合)



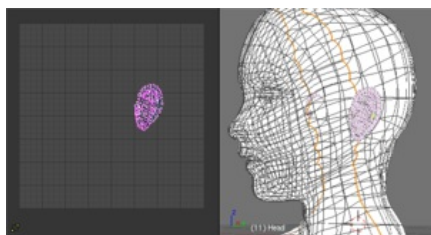
悪い展開-耳と首に注目してください

ここで使った例のように、顔のようなオブジェクトを展開することはよくあるでしょう。展開結果は 'だいたい正しい' ですが、一部が正しく展開されていないか、ひどく入り組んでいます。右の画像は、球状マッピングをした顔の最初の展開結果を示しています。耳に問題があり、UV がぐちゃぐちゃになっています。首については、引き延ばされ下に折り畳まれています。これをきれいにするのは大変です。



耳や首を除いて顔だけを展開

耳は側面図からの直線的な投影で、首は円筒状の展開でうまくいきそうです。ですから一般的なアプローチは、オブジェクトの部品(顔、耳など)別に、何であれその部品に一番うまく働く計算方式を選んで展開することです。そんなわけで始めましょう。顔の面だけを選び 球状投影方式で展開し、スケールと回転で UV/画像エディタの画像領域におさまるようにします。

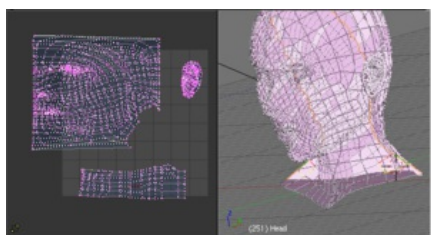


投影による展開: 耳

顔に満足したら、今度は耳に興味を向けます。初めに、作業していた顔の頂点をすべて非選択にします。UV が UV/画像エディタから消えますが、なくなったわけではなく、単に見えなくなっただけです(これを確認するには、3Dビューで面をいくつか選びます。UV が UV/画像エディタに現れます)。

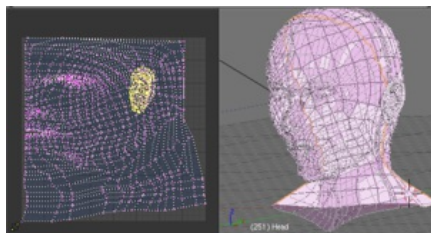
耳にとりかかるため、3Dビューで今度は "耳" の面だけを選択します。頂点グループを使って耳の面を選択することができます。サブメッシュの選択も、残りのメッシュと繋がっていないため簡単です。単にリンクした頂点を選択すればサブメッシュ全体を選ぶことができます。要するに、いまは編集モードにいますので、選択/非選択の機能をすべて利用できます。

それでは再度、側面図からの 投影方式で耳を展開し、スケールと回転を少し施して(次の章で扱います)、片側に寄せましょう。これを、UV の方式を変えながら繰り返し行うことができます。再計算すると、選択中の面の UV がどこか別の場所に置き直されるだけです。部品ごとに、ベストフィットし、この後の描画作業に一番都合のよい配列になるような展開方式を選んでください。



UVマップして合わせる

様々な方式を使ってメッシュのすべての部品が展開されたら、最終的に右の例のような状態になるでしょう。メッシュの各部はすべて割り付けられ、この割り付けはすべて、同じ UV テクスチャマップ上に並べられています。おめでとうございます！ここから、UV マップ全体を1つのマップとして組み立てるのに縫い合わせる(次章で扱います)のは簡単なことです。



整頓し縫い合わせた UV マップ

整頓と縫い合わせが終わると、最終的に右の例のような結合された UV マップができます。一枚の画像で UV マップ全体を覆えるように、そして細部の描き込みが必要なメッシュをすべてペイントできるように、並べられています。この方法についてはすべて、次章で詳しく扱います。この段落の要点は最終目標をお見せすることです。注意してほしいのは、右の図で示されたメッシュは Z 軸に対してミラーコピーされているので、顔の右側は仮想的なものだということです。反対側からの正確な複製なので、実際に存在する UV は 1 セットだけです(もっと具体的な話をすると、*Mirror* モディファイアが適用されれば鏡像は物理的に複製され、結果として完全な頭ができます。こうすると片側だけを編集して左右を物理的に違う形にすることもできます。UV 展開するとフルセットの UV が(両側に)でき、両側で別の絵を割り当てられるため、現実感が増します)。

Average Island Scale

(島の大きさを平均化)

島の大きさを平均化(Average Island Scale) ツールまたはショートカット CtrlA を使うと、UV の島の大きさをスケールして、すべてがほぼ同じサイズになるようにします。

Packing Islands

(島の詰め込み)

島を梱包(Pack Islands) ツールまたはショートカット CtrlP を使うと、UV 空間をできる限り埋めるように島を変形します。テクスチャ空間を有効に使うために重要なツールです。

Constraining to Image Bounds

(画像範囲への拘束)

画像範囲に拘束(Constrain to Image Bounds) をオンにすると、UV が 0 から 1 の範囲外に動かせないようにします。

Iteration and Refinement

(繰り返しと精練)

少なくとも凡人は "1 回で正しく行う" ことはありません。"完了" という魔法のマイルストーンに到達するまでは、発想を得て造形作業を行うことの繰り返しです。ソフトウェア開発ではこれはスパイラルモデルと呼ばれています。

コンピュータグラフィックスに当てはめると、以下の様な作業の繰り返しです: モデリング、テクスチャリング、アニメーティングのあとでまた戻ってメッシュに変更を加え、再び UV マッピング、アニメーションの微調整、ボーンを1つか2つ追加し、面がもっと必要なことに気づき、モデリングに戻って、など。時間やお金、忍耐が尽きるか、または結果に本当に満足する稀なケースもありますが、それまではこうしたことを何度も何度も続けるのです。

Refining the Layout

(レイアウトの精練)

最終的に自分の作ったキャラクターを見て、特定箇所にもっと細かな作業が必要だとわかったとき、精練のための時間がやってきます。例えば、目の周りにカラスの足あとが必要だったり、ベストにロゴを追加しなければならないような場合です。画像の編集を始めると、詳細を描き込むにはピクセルが足りなかったことに気づきます。

唯一の選択肢は UV 面のサイズを拡張(スケールアウト)することです。最小限の引き延ばしか拡大縮小のコマンドを使って目や胸周りの UV 面を広げ、割り当てるピクセルを増やせたら、同時に他の何か、例えば後頭部からピクセル(ディテール)を奪います。UV マップの精練が終われば次は画像を編集して、もっともらしく見えるように、必要な細部を描き入れます。

Reusing Textures

(テクスチャの再使用)

もう一つの考慮すべき点は、リソースの節約です。画像ファイルはそれぞれメモリにロードされます。別のメッシュに同じ画像を再利用できれば、メモリ節約になります。ですから例えば、汎用的な '顔' の絵を用意して別々のキャラクターに使い、区別がつくように UV マップ、形状や小道具(サングラス)を変えるとよいでしょう。

"色褪せたブルー jeans" のテクスチャがほしい場合には、その画像を使ってキャラクターの足だけを展開するのがよいでしょう。汎用的な肌の画像を用意してキャラクターの手、足、腕、脚、首に使うと便利です。ファンタジーの剣をモデリングするとき、剣の刃の一部用の小さな画像があれば十分です。その画像を剣の長さ分再利用できるように、剣の面の展開をリセットします。

Applying Textures

(テクスチャの適用)

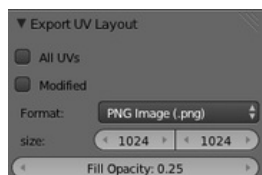
遅かれ早かれ、自作のモデルに画像テクスチャを使いたくなるでしょう。外部アプリケーションでテクスチャを描くなら、テクスチャのどの部分がメッシュのどの部分に使われるのか、知っておかないといけません。作成した UVマップにはテスト画像を使ったテストも必要かもしれません。この章では UVマップの輪郭線のエクスポート方法と、画像を UVエディタに読み込む方法を扱います。

Exporting UV Layout Image

(UV レイアウトのエクスポート)

Export UV Layout (UV 配置をエクスポート) ツール (UV/画像エディタウィンドウの UVメニューにあります) を実行すると、選択中のオブジェクトの UV マップの輪郭線を、Targa (.tga)、EPS、または SVG 形式の画像として保存します。この画像は UVテクスチャを描くアーティストとのやりとりに使えます。

ツールを実行するとファイルブラウザが開き、以下の保存オプションを選べます:



エクスポートのオプション

All UVs (すべてのUV)

無効なら、選択された UV面だけの輪郭線が描かれます

Modified (モディファイア)

モディファイア適用後のメッシュから UV をエクスポートします

Format (フォーマット)

保存に使う画像の形式を選びます (.png、.eps、.svg)

Size (サイズ)

画像サイズをピクセル単位で選びます。画像は正方形になります。

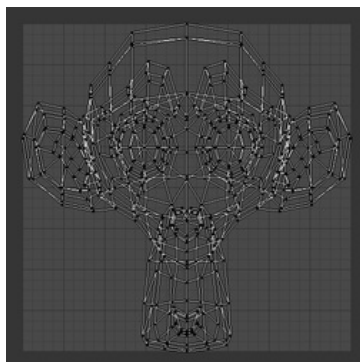
Fill Opacity (塗りつぶしの不透明度/透過部分を埋める)

塗りつぶし色の不透明度を設定します。

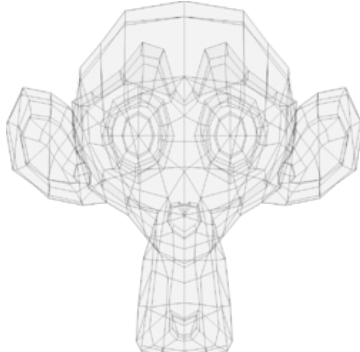
エクスポートされる画像は、UVマッピング領域の画像領域内にある、UV の辺に相当する線の集まりになります。領域より外にある辺は、もし選択されていたとしても、保存される画像には描かれません。

アーティストはペイントプログラムでこの画像を透過レイヤーにして、テクスチャ描画位置のガイドに使えます。下の例の左は blender の UVエディタ、右はテクスチャ制作中の Gimp で、保存したレイアウトをガイドにしています。なお targa 形式はアルファチャンネルをサポートしているので、透明度を持って塗ることができ、メッシュにも反映されます。

テクスチャとして画像を使う方法については、[画像テクスチャ](#) のページをご覧ください。



UVエディタ上の UVレイアウト



画像エディタで使用中の UVレイアウト

Applying Textures to UVs

(テクスチャの UV への適用)

UV/画像エディタを使うと、テクスチャを直接メッシュの面にマップできます。3Dビューウィンドウにはテクスチャの張られたオブジェクトが表示されます。3Dビューウィンドウの描画モードを Textured (テクスチャ) に切り替えると、UV/画像エディタウィンドウで行った変更をすぐに確認できます。逆も同様です。

マテリアルなしでも、キャラクターやオブジェクト用に UVテクスチャを使えば、画像の編集や読み込みを行って 3Dウィンドウで見ることができ、Blender Game Engine でゲームを遊ぶことができます。これができるのは '本当の' レンダリングが一切行われていないからです (ただの viewport shading です)。試しにレンダリングしてみると、デフォルトではこのテクスチャは現れません。

画像をレンダリングするためには以下の作業が必要です。

1. オブジェクト用にマテリアルを作る
2. Blender に、レンダリング時に各面に UVテクスチャを使うよう指示する

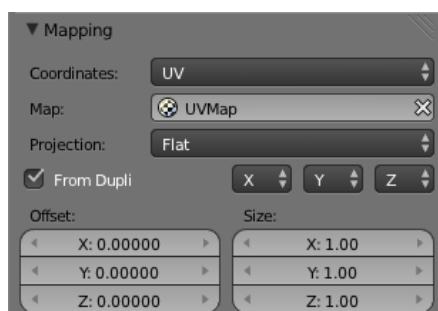
マテリアルを作るには、プロパティエディタのマテリアルタブで Add New (新規) をクリックする必要があります。

レンダリングに UV テクスチャを使う方法は二つあります: 正式な方法と素早い方法です:

- UV座標の使用
- 面テクスチャ

Use UV Coordinates

(UV座標の使用)



UV座標を使ってマップするテクスチャ設定

プロパティエディタのテクスチャコンテキストでテクスチャを追加したら、"Image or Movie" (画像/動画) を指定し、使いたい画像を読み込みます。Mapping セクションで Coordinates (座標) を UV にします。

デフォルトで、Influence (影響) パネルの Color (色) にチェックが入ります (色として結果に反映します)。アルファチャンネルを持つ画像の透明度を反映させるなら、同セクションの "Alpha (アルファ)" にチェックを入れます (さらにマテリアル側で Transparency (透過) の設定が必要です)。

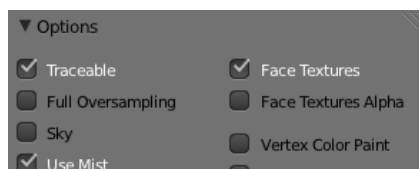
画像テクスチャの使い方の詳細は [画像テクスチャ](#) のページをご覧ください。

レンダリングにはマテリアルが必要です

Blender ではマテリアルを割り当てることなくメッシュに UVテクスチャを関連付けることができます。textured 描画モードの 3D ビューで、その結果を見ることができでしょう。ところがレンダリングしてみると、デフォルトのただの灰色で表示されます。画像を読み込んでいなければ黒色になるでしょう。画像を使ったテクスチャを作らないのであれば Face Texture (面テクスチャ) を有効にして、オブジェクトが生成式の (procedural) マテリアル設定でレンダリングされるようにします。

Face Textures

(面テクスチャ)



Material パネルで Face Textures (面テクスチャ) を有効にしたところ

もう一つの方法は面テクスチャ (Face Texture) のマテリアルを見せるものです。

追加したマテリアルの Options パネルで Face Textures (面テクスチャ) を有効化します。この方法はお手軽ですが、速く結果を得るために通常のレンダリングシステムを回避するので、透明度や陰影処理の正確性に欠けます。

Loading and Saving Images

(画像の読み込みと保存)

UVエディタでは、面ごとにテクスチャを割り当てることができます。そのために、はじめに画像を用意します。Image (画像) メニューからファイルブラウザを使って画像ファイルを開くことができます。使いたい画像が .blend ファイルにすでに読み込まれているなら、ヘッダーにある Browse (閲覧) ボタンをクリック

クします。そうでなければ、New(新規)ボタンをクリックします。グループで作業しているときや、Blenderと同時に外部のペイントプログラムを起動して画像を編集しているときは、画像ファイルが更新され保存されたら、UV/画像エディタのImage(画像) » Reload(再読み込み)を実行し、最新で最良の状態をBlenderに反映させます。Reload(再読み込み)は、1枚の画像に複数の面をマップしているときにも使ってください。3Dビューに各面の最新のマッピングを反映します。

画像ファイルの場所を移動した場合は、Blenderが見つけれられないかもしれないので、Image(画像) » Replace(置き換え)を行います。このオプションはまったく別の画像をUVレイアウトにマップするときに使います。

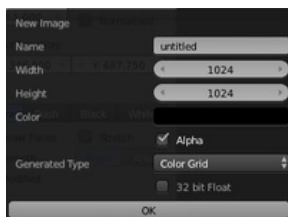
Replacing the active Image

(アクティブな画像の置き換え)

面はそれぞれ座標を持ち、画像に関連付けられていたことを思い出してください。面を別の画像にマップするには、まず面を選択し、UV/画像エディタにあるメニューImage(画像) » Replace Image(画像の置き換え)を選び、既存のファイル(例えばJPGやPNGファイル)で現在の画像を置き換えます。

New Images

(新規画像)



新規画像のダイアログ

New Image(新規画像)を選ぶと、いくつかのオプションが提示されます。ここで生成する画像は、プロパティシェルフ(N)であとから編集することもできます:

Image Name(画像名)

生成画像の名前を設定します

Width(幅) / Height(高さ)

ピクセル単位で画像のサイズを設定します

Color(色)

無地画像を作る場合の塗りつぶし色を設定します

Alpha(アルファ)

画像にアルファチャンネルを作ります(透明度を反映します)

Generated Type(生成する種類)

生成する画像の種類です。無地、およびUVマップのテスト用グリッドの画像2種類から選びます。テスト用グリッドはUVとメッシュとの対応を見たり、間延びを抑えたりするのに役立ちます。種類は生成後も変更可能です:

UV Grid(UVグリッド)

各マスに色つき十字の入った市松模様(チェッカーパターン)を作ります

Color Grid(カラーグリッド)

各マスに行列番号(英字と数字の組み合わせ)の入った格子模様を作ります。

Blank(無地、ブランク)

指定色で無地の画像を生成します。

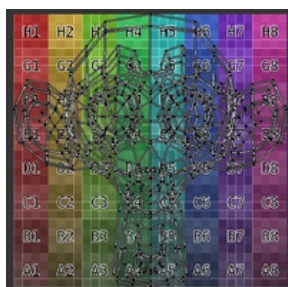
32 bit(32ビットFloat)

画像を32bitで作ります。ファイルサイズが大きくなりますが、標準的な8bit画像より多くの情報を保持できます。接写用や広い階調を持つ画像では、32bit画像を使うほうがよいかもしれません。

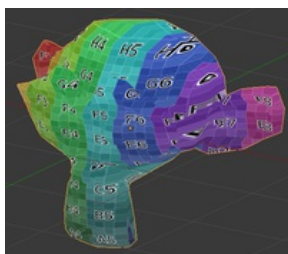
Using the Test Grid

(テストグリッドの使い方)

UVテストグリッドは、間延びしたり歪んだりしている面がないか、調べるのに使います。モデルに着せる服として均一なパターンの画像を使う場合、パターンの間延びを避けて布地のように見せたいと思うはずですが(材質をスパンデックスに見せたいのでない限り)。



UVに使われているテストグリッド



ジオメトリに貼られたテクスチャの
プレビュー

レンダリングすると、メッシュはテストグリッドの色で表示され、UV テクスチャは指定した大きさの画像になります。UV 画像は Image (画像) » Save (保存) メニューで保存できます。

Image Settings

(画像の設定)

UVエディタ内に画像が読み込まれるか生成されるかすると、プロパティシェルフ N)の項目が増えます。最初の行のボタンは次のような機能を持ちます:

- 画像の閲覧
- 画像名の変更
- Fake User のセット
- 新規画像の作成
- 画像を開く
- データブロックのリンク解除

Source (ソース) メニューから画像の種類を選べます。それぞれ違ったオプションを持ちます:

Generated (生成)

新たな画像を生成します。オプションは上述の「新規画像」と同じです。

File (ファイル)

画像ファイルの読み込みに使います:

Fields (フィールド)

画像がフィールドでできているときに使ってください。有効にすると Upper First (上優先 (偶数)) か Lower First (下優先 (奇数)) を選べます。

Premultiply (アルファ乗算済み/プリマルチプライ)

アルファ乗算済みの RGB を使います

Movie (動画) と Sequence (連番画像)

Frames (フレーム)

使用するフレーム番号

Start (開始)

動画/連番画像の開始フレーム

Offset (オフセット)

アニメーションに使うフレーム番号のオフセット

Fields (フィールド)

レンダリングされるフレームごとのフィールド数を設定します (2フィールドは1フレームです)

Auto Refresh (自動更新)

フレームが変わるたびに画像を更新します

Cyclic (繰り返し)

動画/連番画像にある画像を繰り返します。

Saving Images

(画像の保存)

Blender の Image (画像) メニューにあるツールで作成したり編集したりした画像は、外部のファイルに保存することができます。画像がすでにファイルである場合には、保存コマンド (AltS) を使ってください。画像が生成されたものであるか、別の名前でも保存したいときは、Save As (別名保存) (F3) を使うこともできます。Save as Copy (コピーを保存) (F3) も指定した名前でもファイルを保存しますが、画像エディタ内では元のものを開いたままにします。

Modifying your Image Texture

(画像テクスチャの修正)

テクスチャの修正方法には次のようなものがあります:

- メッシュの見た目をテクスチャにバイクする [Render Bake\(レンダーバイク\)](#)
 - Render Bake(レンダーバイク)機能は、現在の画像を以下のレンダリング結果に基づいて置き換えるためのツール群です: [頂点ペイント](#)、[法線/バンプ](#)、[プロシージャル\(生成式\)マテリアル](#)、[テクスチャ](#)、[照明効果](#)、[アンビエントオクルージョン](#)
- [Texture Paint](#) を使ったペイント
 - UV/画像エディタのメニュー Image(画像) » New(新規) を選び、メッシュにペイントを開始します
- 外部プログラムを使った画像作成
 - 好きなペイントプログラムを使えば、エクスポートした UV レイアウトからテクスチャを作れます。変更を保存したら Blender に戻り、メッシュの面選択モードで Image(画像) » Open(開く) メニューから UV 画像を希望の(アクティブな) UVテクスチャのレイヤーに読み込みます。Image(画像) » Edit Externally(外部で編集) メニューを選ぶと、User Preferences(ユーザー設定) で指定した画像エディタで画像が開かれます。
- 「projection painting(投影ペイント)」機能。Blender の最近のバージョンで使えます
- "Bake uV-Textures to Vertex Colors" アドオン。頂点カラーから画像を作ります
- 上記の方法の組み合わせ

UVペイント、テクスチャのバイクは、作成した画像で現行の画像を置き換えます。テクスチャペイントや外部ソフトウェアは、画像に描き足したり質を上げるのに使えます。どの方法を使うにしても、最終的には次のいずれか、または両方を行う必要があります。

- テクスチャを別の画像ファイルに保存します (例えばカラー画像なら JPG、アルファ値を持つなら PNG)
- blend ファイル内部に画像をパックします (UV/画像エディタの Image(画像) » Pack as PNG(PNGとしてパック))

別ファイルに保存すると、別のファイルで上書きすることで、簡単にテクスチャを切り替えることができます。また、編集作業に外部プログラムを使えます。一方、パックするとプロジェクト全体がひとつの .blend ファイルに保持されるので、管理が楽になります。

Image(画像) » Invert(反転) メニューを選ぶと、画像の色を反転できます。

Packing Images inside the Blend file

(画像を Blender ファイルにパック/梱包する)

.blend ファイルをパックすると、現行版の UVテクスチャ画像がすべて、ファイル内に取り込まれます。このファイルがあとで変更されても、自動的な再パックは行われず、古いバージョンの画像が使われ続けます。更新するためには、自分でパックし直すか、画像を読み直す必要があります。

画像をパックするには Image(画像) » Pack Image(画像をパック) メニューを選ぶか、プロパティシェルフ(N)の Image(画像) パネルで、該当画像の横にある、箱に紙が入ったアイコンをクリックします。アンパック/梱包解除には、このアイコンをもう一度クリックします。

File(ファイル) » Append(アペンド) 機能は自動的に .blend ファイルを調べ、その中にパックされているテクスチャを表示します。パブリックドメインの Blender Texture CD は優れたリソース集です。他にもパブリックドメインの(そしてライセンスされた)テクスチャがたくさんあります。Elephants Dream CD にあるすべてのテクスチャは教育用に [CC-BY 2.5](#) でライセンスされています。

Material Textures Influence

(マテリアルテクスチャの影響)

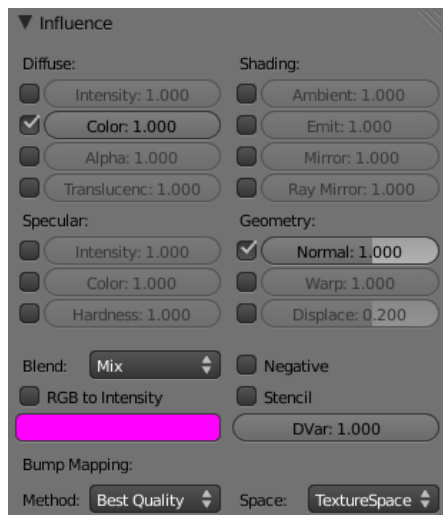
テクスチャはマテリアルの色だけでなく、マテリアルの他の多くのプロパティにも影響を及ぼすことができます。Influence (影響) パネルでは、テクスチャが作用するマテリアルのさまざまな特徴を調整できます。

備考

Surface と Wire マテリアル用のテクスチャオプションです。場合によっては Volume と Halo マテリアルでも使われます。

Surface and Wire materials

(サーフェスとワイヤーフレームマテリアル向け設定)



テクスチャの、サーフェス マテリアル用の Influence パネル

Diffuse

(拡散/ディフューズ)

Intensity (強度)

拡散反射率へのテクスチャの影響度

Color (色)

基本色またはRGB値へのテクスチャの影響度

Alpha (アルファ)

不透明度の影響度。[アルファ値を使ったオブジェクトの透明度表現 \(eng 2.4\)](#)をご覧ください。また、光源用には Z Transparency (Z値透過) と、混ぜ合わせる場合は複数チャンネルを使ってください。

Translucency (透光性)

透光性への影響度

Specular

(反射/スペキュラー)

Intensity (強度)

鏡面反射率へのテクスチャの影響度

Color (色)

Specular Color (反射色) への影響。反射色とは、つやのある材質に照明が作る反射の色です。

Hardness (硬さ)

反射の硬度への影響度。デフォルト値 1.0 は Hardness (硬さ) 130に、値 0.5 は Hardness 65 に相当します。

Shading

(陰影)

Ambient (周囲)

マテリアルが受ける Ambient Light (環境光) への影響度

Emit (放射)

マテリアルの放射する光への影響度

Mirror (ミラー)

鏡映色への影響度。Environment Map や レイトレースされた反射と連動します

Ray Mirror (レイミラー)

レイトレースされた鏡映反射の強度への影響度

Geometry

(ジオメトリ)

Normal(法線/ノーマル)

いわゆる bump mapping で、面の法線方向を変えます。bump mapping を通じて擬似的にキズや凸凹を作ったり、浮き彫りを作ったりするのに使われます。

Warp(歪み/ワープ)

隣のテクスチャチャンネルのテクスチャ座標へ作用したり、歪めたりします。新たな歪みが設定されない限り、あとのチャンネルにも効果が持続します。ゼロにすると効果をキャンセルします。

Displace(ディスプレイス)

頂点の displacement(ディスプレイスメント)への影響度。[ディスプレイスメントマップ](#) 用です。

Other Controls

(その他のコントロール)

Blend(ブレンド)

実行するブレンド操作。詳細は [Compositing Mix Nodes \(eng\)](#) をご覧ください。

RGB to intensity(RGBを強度に変換)

有効にすると、RGB テクスチャ(色に影響)を強度のテクスチャ(値に影響)として使います。

Blend Color(ブレンド色)

テクスチャが色に対応付けられているとき、テクスチャの強度に従って混ぜ合わせる色を、カラースウォッチか RGB値のスライダーで選びます。

Negative(負)

テクスチャの効果を逆にします。通常は白のピクセルがオン、黒のピクセルがオフですが、このオプションを使うと反対にできます。

Stencil(マスク)

アクティブなテクスチャを、後に続くすべてのテクスチャ用にマスクとして使います。半透明のテクスチャや "Dirt Maps" に役立ちます。黒のピクセルはテクスチャを反映しません。Stencil モードは 2D プログラムのレイヤーマスクに似た動作をします。stencil texture の効果を上書きすることはできず、場所を広げることだけができます。入力として intensity map が必要です。

DVar(デフォルト値)

RGB 以外で使われる Destination (目的) 値です。この値を使って Intensity テクスチャが現行の値とブレンドされます。2つ例をあげます:

- 通常、Emit(放射)値は 0 です。デフォルトでは DVar が 1 なので、Emit に対応付けられたテクスチャでは最大の効果を得られます。DVar を 0 にするとどのテクスチャも効果を持ちません。
- 透過するマテリアルを使うために Alpha(アルファ)に対応付けられたテクスチャを使っても、デフォルトの設定では何も起きません。これは マテリアル パネルの Alpha 値が 1 であるためです。透明のマテリアルを得るには DVar を 0 にする必要があります(もちろん Z Transparency(Z値透過)の設定も必要です)。これは入門者がよく迷うところです。もしくは裏表を逆にした方法を使います - Alpha を 0 にして Dvar を 1 のままにします。もちろんテクスチャは反転させて使います。

Bump Mapping

バンブマッピング用の設定です。

Method(方式)

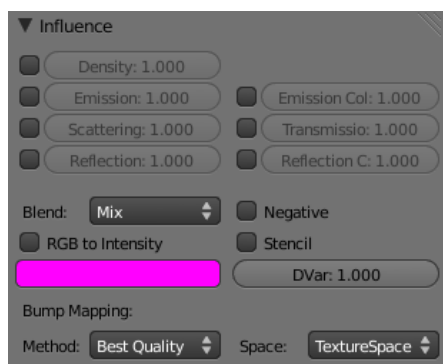
Best Quality(最高品質)、Default(デフォルト)、Compatible(互換)、Original(オリジナル)

Space(空間)

Texture Space, Object Space, View Space

Volume materials

(ボリュームマテリアル向けの設定)



テクスチャの、Volume マテリアル用 Influence パネル

Volume マテリアル用の特別なテクスチャオプションです。

Density(密度)

ボリュームの密度へのテクスチャの影響度です。

Emission(放射)

ボリュームの放射へのテクスチャの影響度です。

Scattering(散乱)

散乱値へのテクスチャの影響度です。

Reflection(反射)

out-scattered light の明るさへのテクスチャの影響度です。

Emission Color(放射色)

放射色への影響度です。

Transmission(透過色)

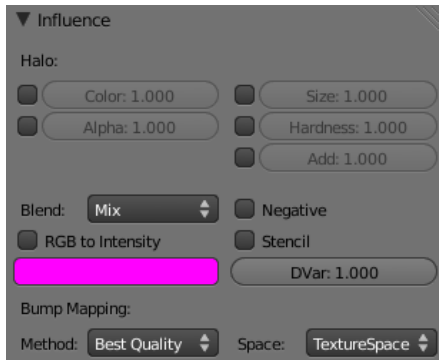
光の散乱/吸収後の結果色へのテクスチャの影響度です。

Reflection Color(反射色)

out-scattered light の色へのテクスチャの影響度です。

Halo materials

(光輪マテリアル向けの設定)



テクスチャの、Halo マテリアル用 Influence パネル

Halo マテリアル用テクスチャ専用のオプションです

Size(サイズ)

ray mirror へのテクスチャの影響度

Hardness(硬さ)

hardness へのテクスチャの影響度

Add(追加)

translucency へのテクスチャの影響度

Bump and Normal Maps

(バンブマップと法線マップ)

Description

(解説)

Normal Maps (法線マップ) と Bump Maps (バンブマップ) はどちらも同じ目的を持ちます。平らな表面の濃淡を変更して細かな凹凸があるように見せ、3D表面の細かな質感を真似ることです。各ピクセルの濃淡を変えるだけなので影は一切作らず、他のオブジェクトを遮ることもありません。表面に平行に近い浅い角度から見ると、表面の形状が本当のものではないことがわかるでしょう。

法線マップとバンブマップはどちらも、ピクセルの陰影処理に関わる法線角度(面の垂線の指す向き)を変えることで動作します。法線マップとバンブマップはよく同義語のように使われますが、若干の違いがあります。

Bump maps (バンブマップ)

カメラの観点 (viewpoint) からのピクセルの相対的な高さである **intensity** (強度) を保存するテクスチャです。ピクセルは面の法線方向に、必要な分だけ動かされたように見えます ("bump" (隆起) は、面の既存の法線ベクトルに沿って (法線ベクトルは変化させず) 位置をずらすことだけで作られます)。グレイスケール画像を使うか、(画像を含む) RGB テクスチャの intensity 値を使うことができます。

Normal maps (法線マップ)

direction (向き) を保存する画像です。法線の向きが画像の RGB 値に直接入っています。直線に沿う面から離れたピクセルだけをシミュレートするよりもはるかに正確で、任意の方向に動かされたピクセルを任意の方法でシミュレートできます。欠点は、手描きで簡単に作れるバンブマップと違って、普通は何とかして生成する必要があり、マップを適用しようとしているジオメトリより高解像度のジオメトリから生成することもよくあります。

Blender の法線マップは法線を次のように保存します:

- Red の (0-255) を X (-1.0 - 1.0) に対応づけ
- Green の (0-255) を Y (-1.0 - 1.0) に対応づけ
- Blue の (0-255) を Z (0.0 - 1.0) に対応づけ

法線はすべて観測者 (viewer) のほうを向くので、負の Z 値は保存されません (どちらにしても見えません)。Blender は全範囲の blue を保存しますが、他の実装には blue の (128-255) を (0.0 - 1.0) に対応づけるものがあります。後者の変換は例えば "Doom 3" で使われています。

Workflow

(作業の流れ)

バンブマップと法線マップの作成と利用の流れは以下の通りです:

1. 高精細 (ハイポリ) のモデルをモデリングします
2. バンブマップや法線マップを Bake (焼き込み) します
3. より低精細のモデル (ローポリモデル) を作ります
4. 通常の座標系でマップをローポリモデルにマッピングします

メッシュツールで高精細モデルを作る方法については、モデリングの章を参照してください。どこまで作りこむかはあなた次第です。たくさん隆起や細部 (こぶ、しわ、突起) を作るほど、マップも高精細になります。

マップの Bake (焼き込み) とは、簡単にいえば、高精細メッシュ (ハイポリ) の細部描写を取り出して、頂点数を減らした同じ形のオブジェクトに用いることです。焼き込みは Blender の [Render Bake](#) 機能を使って行えます。

ローポリモデリングは Blender のメッシュ編集ツールで行います。面の数は、通常は元の形状を示せる程度にすべきです。例えば、ハイポリモデルで細かく作られた耳が 1000 枚の面でできているとします。ローポリモデルではこれを、元の耳のメッシュと同じ方向を向いた 1 枚の平面で置き換えることがあります。(ヒント: ここでは Blender の [multi-resolution mesh](#) モデリング機能を効果的に使えます)。

マッピングはテクスチャをローポリメッシュに適用する作業です。メッシュのマテリアルにテクスチャを適用する方法について、詳しくは [テクスチャマッピングの章](#) を調べてみてください。バンブマップや法線マップでは特に次の点を確認してください:

- バンブマップを使うときは、テクスチャを Normal (法線) にマップして No RGB を有効化します。
- 法線マップを使うときは、テクスチャを Normal (法線) にマップします。

2つのオブジェクトの座標系は一致しなければなりません。ハイポリモデルの UV マップを使ってバイクするなら、ローポリモデルの UV マッピングを行なって、UV 座標をハイポリ画像の輪郭線にあうように並べる必要があります (ハイポリモデルの辺に揃える方法は [UV 展開](#) をご覧ください)。

Displacement Maps

(ディスプレイメントマップ)

Description

(解説)

ディスプレイメントマッピングを使うと、テクスチャを入力にして、レンダリングされたジオメトリの頂点位置を操作することができます。陰影を歪ませて凹凸の魔法をかける[法線マップ](#)や[バンプマップ](#)と違って、ディスプレイメントマップは実際のメッシュに、本当の凸凹やしわ、隆起などを作ります。このため変形されたメッシュは影を作り、他のオブジェクトを遮り、その他本当のジオメトリができるあらゆることを行えますが、一方で動作に必要な頂点はずっと多くなります。

Options

(オプション)

[Influence \(影響\) パネル](#) では、ディスプレイメントの強度を Displace (ディスプレイス) と Normal (ノーマル) のスライダーで調整できます。

- テクスチャが法線情報のみを提供する場合は (例: Stucci (凹凸))、頂点はテクスチャの法線データに従って移動します。法線の移動は Normal (ノーマル) スライダーで調整できます。
- テクスチャが強度情報のみを提供する場合は (例: Magic (マジック)、色に由来)、頂点は法線の向きに沿って移動します (頂点自身は法線を持ちませんが、隣接面から結果的に導かれるベクトルです)。白いピクセルは法線の向きに外側に移動し、黒いピクセルは反対向きに移動します。移動量は Displace (ディスプレイス) スライダーで調整できます。

2つの方式を同時に使うこともできます。テクスチャタイプの多くはどちらの情報も提供します (Clouds (クラウド)、Wood (木目)、Marble (マーブル)、Image (画像))。各タイプの量は、それぞれのスライダーを使って混ぜることができます。強度のディスプレイメントは頂点が外側にだけ移動するので、表面を滑らかに連続したものにします。法線のディスプレイメントは頂点が複数の方向に移動するので、もつと (aggregated) 表面を作ります。

ディスプレイメントの深度はオブジェクトの大きさに比例しますが、データにの大きさに比例しません。例えばオブジェクトモードでオブジェクトを2倍のサイズにすると、ディスプレイメントの深度も2倍になり、ディスプレイメントの見目は同じままです。編集モードで拡大すると、ディスプレイメントの深度は変化しません。したがって深度は相対的に小さく見えます。

Hints

(ヒント)

ディスプレイメントマップは物理的なメッシュの面ではなく、レンダリングされた面を動かします。したがって、3Dビューでは表面が滑らかに見えるかもしれませんが、レンダリングすると凸凹になります。表面の細部描写のためにはディスプレイスすべき面がなければならず、また非常に小さくなければいけません。これは、メモリや CPU 時間を使うか、レンダリングの質をとるかトレードオフの関係になります。

良し悪しは様々ですが、ディスプレイメントは次のようなオブジェクトとともに利用でき、ここに書かれた方法でレンダリングされる面の大きさを調整します。

[Subdivision Surface \(サブサーフ\) のメッシュ](#)

レンダリングされる面の大きさはサブサーフのレンダリング時のレベルで調整できます。ディスプレイメントは滑らかな法線に本当に合います。

編集モードで手動で [細分化した](#) メッシュ

レンダリングされる面を分割数で調整できます (上の方式と組み合わせることができます)。ディスプレイメントの結果は単純なサブサーフとまったく同じですが、増えた面の描画分 OpenGL に負荷がかかり、編集効率は落ちるでしょう (この方法では編集時の細分化レベルを下げることはできません)。

[メタオブジェクト](#)

レンダリングされる面をレンダーワイヤーサイズで調整できます。ワイヤー値が小さいほど面が増えます。

次の方法もありますが、現時点ではうまく動作していません。レンダリング前にメッシュに変換しておくのがおすすめです。

開いた [NURBS Surfaces \(サーフェス\)](#)

レンダリングされる面を U/V の Surface Resolution (サーフェス解像度) で調整できます。数値が高いほど面が増えます (法線のエラーに注意してください)。

閉じた [NURBS Surfaces \(サーフェス\)](#)

Surface Resolution (サーフェス解像度) で調整できます (法線のエラーや潜在的な seam の出現に注意してください)。

[Curves \(曲線\)](#) と [Text \(テキスト\)](#)

Surface Resolution (サーフェス解像度) で調整できます。値が高いほど面が増えます (大きな平らなサーフェスにはディスプレイスできるレンダリング面が少ししかないことに注意してください)。

Displace モディファイア

ディスプレイメントの調整項目を増やすために、[Displace \(ディスプレイス\) モディファイア](#) を使うとよいでしょう。この機能にはたくさんの別のオプションがあり、ディスプレイメントをお好みに合わせてカスタマイズできます。

ワールド



World panel

Blenderには、レンダリングする際に素敵な背景や奥行き効果を与えるセッティングがいろいろあります。これらはWorldコンテキストでアクセスできます。デフォルトでは、非常に均一で単調なワールドになっています。それを編集するか、または新しいワールドを追加できます。

使えるのは、

[Background](#)

ワールドの背景のカラーとテクスチャ。マッピング座標を設定できます。

[Mist](#)

奥行き感を強調するために切りの効果を付け加えることができます。

[Stars](#)

ランダムな光点が背景を覆います。

これらの設定は簡単にシーンに効果を追加できますが、しばしば[コンポジットノード](#)のほうが好まれます。これは複雑なのでマスターするのは容易ではありませんが、さらなるコントロールやオプションを提供してくれます。たとえば、Z値(カメラからの距離)や法線(面の向き)の情報をコンポジットノードでフィルタリングすれば、さらに深みや透明さを増すことができます。

Note

Worldパネルにあるいくつかの設定は照明に影響することなので、[ライティング](#)の章で解説されています([アンビエントライト](#)、[露出](#)、[アンビエントオクルージョン](#)を参照)。Sun Lampを使っている場合はLampメニューでSky & Atmosphereのオプションが利用できます。

背景

解説

worldプロパティではシーンの基本的なシェーディングの設定ができます。その中にはアンビエントカラーや、霧などの特殊な効果などがありますが、もっとも良く使われるのは背景カラーのシェーディングでしょう。

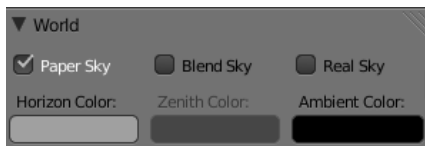
レンダリングの背景

レンダリングの背景に画像を使うには、[Outputパネルで指定する背景イメージ](#)を参照。

3Dビューの背景

モデリングの参考などのために3Dビューの背景に画像を使うには、[背景画像を使う](#)を参照。

オプション



World panel

Horizon Color

地平線のRGBカラー。

Zenith Color

天頂のRGBカラー。

これらの色がどう使われるかはSkyの種類によります。

None Enabled

三つのボタンがどれも無効になっている場合は、背景は単色(地平線)になります。

Paper Sky

このボタンが有効の場合、レンダリングされたイメージの範囲にグラデーションが収められます。カメラがどのようなアングルでも、地平線はイメージの中心を通ります

Blend Sky

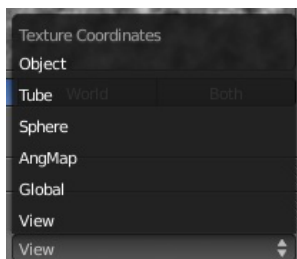
地平線から天頂にかけてカラーをブレンドします。このボタンのみが有効の場合、グラデーションはカメラアングルに関わらずレンダリングイメージの下から上へ移り変わります。

Real Sky

有効にすると、グラデーションは天低(天頂と同じ色)から地平線へ、地平線から天頂へと二回移り変わります。カメラアングルによっても結果は変わり、よりリアルな表現になります。地平線の設定色は正確に地平線方向(XY平面上)に位置し、天頂の設定色はカメラの真上の点に使われます。

テクスチャ

単色または二色のブレンドの代わりに、2Dイメージを使い、シーン全体を巨大な球状または箱状に包んだり、シーン内の下層的な空間に貼り付けたりすることができます。



Texture Coordinates popup menu

ワールドテクスチャはテクスチャメニュー(Worldを選択後、Texture)で設定します。これは二つの点以外はマテリアルテクスチャに扱いが似ています。テクスチャは以下の方向でマッピングできます：

View

デフォルトの設定です。最終的なレンダリングの視点に対してマッピングされます。

Global

グローバル座標を使います。

AngMap

標準的な半球のアンギュラマップを覆うのに使います。これは空の色にアンビエントオクルージョンを適用して、イメージに基づいて照明するのに使えます。基本的にHDRのアンギュラマップ(球形のような奇妙な画像)が必要です。

Sphere

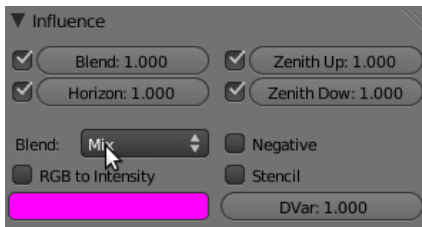
球状マッピング。マテリアルの場合と似ています。

Tube

長方形のテクスチャを円筒に巻きつけるようにマッピングします。マテリアルの場合と似ています。

Object

指定したオブジェクトのローカルテクスチャ座標に基づいてテクスチャを配置します。



Texture Influence panel

テクスチャが影響するのはカラーのみですが、それには4種類の方法があります。

Blend

テクスチャが非ゼロの位置では地平線のカラーが現れます。

Horizon

地平線のカラーに影響します。

Zenith Up

天頂のカラーに影響します。

Zenith Down

天底のカラーに影響します。

霧

解説

霧はレンダリングイメージの奥行き感を強調することができます。霧を作り出すために、Blenderはカメラから遠くにあるオブジェクトほど(アルファ値を下げて)透明になるようにして、背景のカラーと合成します。

オプション



Mist panel

Mist check box

霧の有効・無効を切り替えます。

Intensity

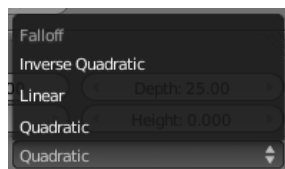
霧の強度の最小値。

Start

霧が現れ始める距離。

Depth

Startの距離から霧が完全に現れるまでの距離。カメラからStart+Depthの距離よりも遠くにあるオブジェクトは完全に見えなくなります。



Mist Falloff popup menu

Height

よりリアルにするために、高さに応じて霧を弱めます。数値を0よりも大きくすると、Z=0の高さを中心とした範囲で霧の強度が最高(下方)から0(上方)まで変化します。

Falloff

霧の減衰率(Quadratic/Linear/Inverse Quadratic)。これらの設定はカメラからの距離と霧の強さの関係をコントロールします。

霧の距離

3Dビューで霧に関する距離を視覚化するには、カメラを選択し、カメラメニューでShow Mistを有効にします。

カメラから伸びた直線がStartとDepthの距離を表すようになります。

視覚化された霧の範囲を評価するためには、カメラを選択して◊ ShiftNum1を押すと見やすいでしょう(Num5で遠近感の有効・無効を切り替えられます)。これで3Dビューはカメラを見下ろすような始点になります。

透明度

霧は透明度を調整することで作られるので、場合によってはオブジェクトが望ましくない形で部分的に透明になってしまうことがあります。一つの解決法は、霧の数値設定などはしておいて、有効にはしないことです。それでも霧のデータはコンポジットに使えます。[Do Composite](#)と[ノードエディタ](#)を使って、霧のパスをAlphaOverに与え、背景カラー(または空だけのレンダレイヤー)とレンダリングイメージを合成します。こうすると霧の効果は得られますが、オブジェクトの透明度(または不透明度)は保たれます。

例



Mist example

この例([blend](#))ではHeightオプションを使って床を覆う煙の高さを制限しています。

このシンプルなシーンは[Stefan Morellの Arc Sci-Fi Corridor](#)にインスパイアされています。

星

解説

星はランダムな光点を背景に配置します。

オプション



Star panel

Size

光点の実際の大きさ。これはデフォルトの値よりもずっと小さくしたほうが良いです。1ピクセルよりも小さくなるようにするとよりリアルです。

Colors

色相をランダムに変えます。または真っ白にします。

Min. Dist

星が配置されるカメラからの最小距離。これはカメラから最も遠いオブジェクトよりもさらに大きな距離にしたほうが良いです。そうでない場合はオブジェクトの手前に星が重なる可能性があります。

Separation

星同士の平均的な距離。星は根本的に3Dで計算されるので、これは空間の中での距離です。イメージ上の距離ではありません。

はじめに

キャラクターを作り終えたら、アニメーションあるいはポーズをとらせるための加工が必要です。リグづけ(rigging)とはキャラクターメッシュに骨格を取りつける工程で、これにより様々な方法でキャラクターを変形させ、ポーズをとらせることができます。

この作業はメッシュを根本から変えるものではなく、簡単に変更ややり直し、他のポーズとの合成ができます。

Wiki編集者向けの註釈

リグづけの全手順に関して、この下にもっと入門的な記述を足す必要があります！

以下はリグづけでよく行われる作業です:

1. アーマチュア (armature) を加えます。これは一本のボーンから始まります
2. ボーンを必要なだけ追加し、「手足 (limbs)」を形作るようにお互いをつなげます。
既存のボーンから押し出すか(この場合、既存のボーンは自動的に、押し出してできたボーンの親になります)、新たなボーンを複数追加してつなぎ合わせます
3. 骨格の調和がとれるようボーンを編集します
4. 接合部に拘束 (constraints) を与えます
(例: 人間のひじは一平面上の170度の範囲にだけ曲がります)
5. 骨格の「rest(休止)」位置(デフォルト位置)を決めます
6. メッシュをアーマチュアにとりつけます (skinning/スキニング)
7. アーマチュアの動きが、スキンにどのように作用するかを決めます
(曲げ、伸縮、膨張)
8. アーマチュアのポーズを決めます
(複数のやり方があります: 各ボーンを手動で配置する方法、アーマチュアのテンプレートを複製する方法、曲線に沿うようにボーンを配置する方法、blender 外部で取得したモーションキャプチャーデータにアーマチュアを沿わせる方法)
9. アーマチュアの動きがうまくスキンに作用しているか検証し、パラメータを調整します
(より自然に見えるように、スキンの構造 (topology) を調整します)

各手順の詳細を以下に説明します。

アーマチュア

アーマチュアは現実世界の骨格に似たもので、メッシュにポージングやアニメーション用の仕組みを作ります。

アーマチュアとボーンパネル

アーマチュアやボーンの設定に使う各パネルを紹介します

ボーン

アーマチュアの基本要素であるボーンのプロパティを説明します

視覚化

ボーンを視覚化する4つの方法を紹介します

組み立て

アーマチュア内のボーンの設定を説明します

要素の選択

アーマチュアの重要部分だけを選ぶ方法について説明します

編集

ボーン編集

Blender でボーン編集を実際に行うやり方を学び、その結果何が起きるのかを見ます

ボーンスケッチ

骨組みのスケッチツール (skeleton sketching tool) を使うとボーンのスケッチが簡単になり、また Blender に現実感をもたらします

テンプレート化 (eng)

テンプレートを使うと、既製のリグを自作のモデルに手早く再利用することができます

スキニング

この章では、アーマチュアからキャラクターに「肉付け」する方法を見ていきます。

標準的な英語では「スキニング (to skin)」とは「皮をはく」意味ですが、ここでは逆の意味で、アーマチュアに皮をかぶせる意味で使います。メッシュをアーマチュアの周囲に配置していきます。

オブジェクトをボーンへ連結 (eng 2.4)

ボーンをオブジェクトの親にする (parent) やり方です。ボーンでこのオブジェクトを制御できるようになります。機械の連結をシミュレートする場合や (例: [Newton's cradle](#))、昆虫やカニの体などのモデリングのように、アーマチュアの動きでメッシュが変形しない箇所に使います。

オブジェクト形状へのスキニング (eng)

各ボーンが「スキニング」オブジェクトの特定箇所を制御できるように、アーマチュアをとりつける方法です。アーマチュアが動くときオブジェクトの表面が伸縮するような場合、例えば腕を曲げたときの力こぶの膨らみに使われます。

再ターゲット (Retargeting) (eng)

モーションキャプチャーデータ (現実世界で取得したもの) をリグに与える方法です。元の動作をリアルにまねることができます。このやり方を使うと各動作の面倒なプログラミングをしなくて済みます。

ポージング

ポージングとは、興味深い構成になるように、各自のやり方でオブジェクトの形を決め、シーン内に配置することです。例えば [考える人](#) のボディラングー

ジを見て下さい。または、さそりが攻撃のため尾を持ち上げたところを想像してみてください。

ポーズはアニメーション作成にも使われます。例えば、テニスプレイヤーのサーブのアニメーションを作るとき、ストローク動作のいくつかの瞬間をとらえたポーズを作る必要があるでしょう。(a) ボールを手に持ちラケットが腰の高さにあるとき (b) ボールを投げ上げたとき (c) ボールを打ったとき (d) その後ラケットがもっとも低い位置にきたとき(フォロースルー)です。こうすると Blender が中間のポーズをすべて作り、アニメーションができます。

[視覚化 \(eng\)](#)

アーマチャのポーズづけに役立つ、特にアニメーション用の視覚的な補助機能について説明します。

[ポーズの編集 \(eng\)](#)

ポーズの作り方、および、ポーズを瞬間ごとに編集し、一連のアニメーションを作る方法です。

[ポーズライブラリ \(eng 2.4\)](#)

よく使うポーズや他のアーマチャにある既存ポーズを保存し、手早く探して使えるようにします。

[拘束の使用](#)

不自然なポーズをとれないように、ボーンに拘束 (constraints) を与える方法です。

[逆運動学 \(Inverse Kinematics\)](#)

連鎖内の最後のボーンを動かすと、Blender が自動的に連鎖全体を動かす機能です。誰かの指を持ち上げることを考えてみてください。指が持ち上がるにつれて、手全体が自然に動くでしょう。

[スプラインK](#)

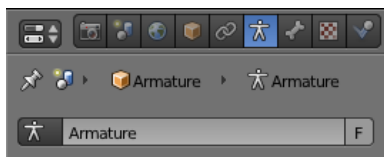
曲線上にボーンの連鎖を沿わせる機能です。

アーマチュアパネルの概要

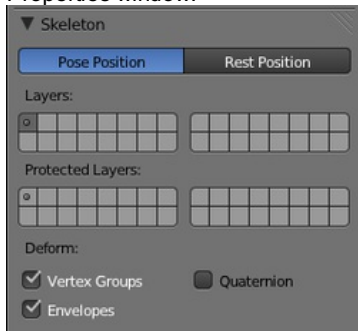
Mode: Object mode, Edit mode and Pose mode

Panel: All in Properties window, Object data property

まずはPropertiesウィンドウのObject dataコンテキストで、アーマチュアの設定を集めた様々なパネル一般を把握しましょう:



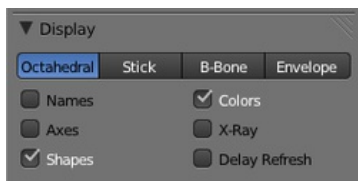
The Object data property in the Properties window.



The Skeleton panel.

スケルトンパネル (全モード)

このパネルでは、異なるレイヤーのボーンの設定をアレンジして、操作をしやすいことができます。

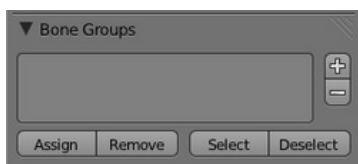


The Display panel.

表示パネル (全モード)

ここでは、3Dビューでのあなたのボーンを表示方法を制御し、4種類の設定を使用できます。

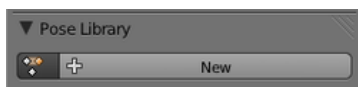
他にもいくつか利用可能なオプションがあり、この後説明します。



The Bone Groups panel.

ボーングループパネル (全モード)

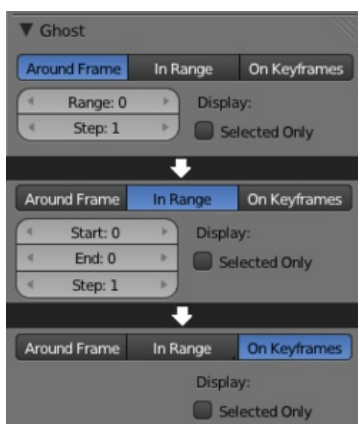
グループへボーンのセットを割り当てることができ、操作と管理をしやすいします。



The Pose Library panel.

ポーズライブラリパネル (ポーズモード)

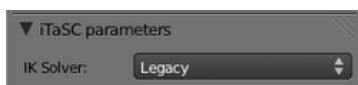
後で使用するために個別のポーズを保存することができます。



The Ghost panel.

ゴーストパネル (全モード)

異なるポーズを一連で連続した見ることができ、アニメーション化する際に非常に有用です。



The iTaSC parameters panel.

iTaSC パラメータパネル (全モード)

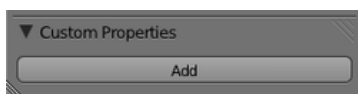
あなたのアニメーションで使用されるIKソルバの種類を定義します。



The Motion Paths panel.

モーションパスパネル (ポーズモード)

このパネルでは、アニメーション化のときに、スケルトンが残すモーションパスの可視化を有効にすることができます。



The Custom Properties panel.

カスタムプロパティパネル (全モード)

カスタムプロパティを定義するためのパネルで、スクリプトを作成するときこれが使用されます。

ボーンパネルの概要

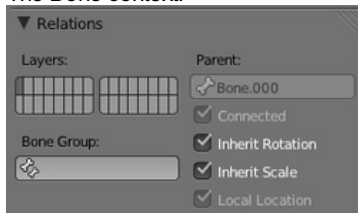
Mode: Object mode, Edit mode and Pose mode

Panel: All in Properties window, Bone property

まずはプロパティウィンドウ、ボーンコンテキストで、ボーンの設定を集めた様々なパネルの一般を把握していきましょう:



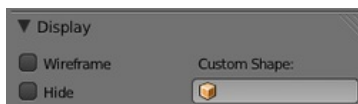
The Bone context.



The Relations panel.

関係パネル (エディットモード)

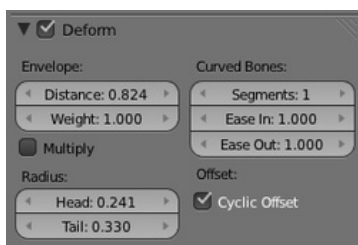
このパネルでは、異なるレイヤーのボーンのセットをアレンジして、操作しやすくなります。



The Display panel.

表示パネル (オブジェクトモード)

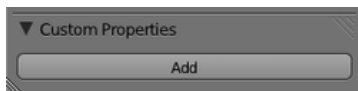
表示パネルは、あなたのボーンのLE外観をカスタマイズでき、別の既存のオブジェクトの形状にできます。



The Deform panel.

変形パネル (全モード)

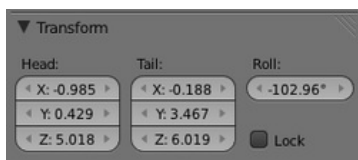
このパネルでは、ボーンの基本プロパティを設定することができます。



The Custom Properties panel.

カスタムプロパティパネル (全モード)

カスタムプロパティを定義するためのパネルで、スクリプトを作成するとき、これが使用されます。

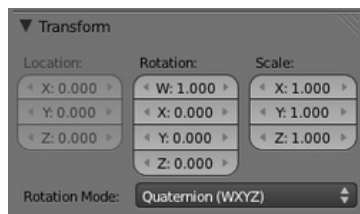


The Transform panel(edit mode).

変換パネル (エディットモードとポーズモード)

編集モードでは、個々のボーンの位置と回転の制御にこのパネルを使用できます。

ポーズモードで設定できるのはメインのボーンのみで、そしてあなたは今、回転とスケールで設定することができます。



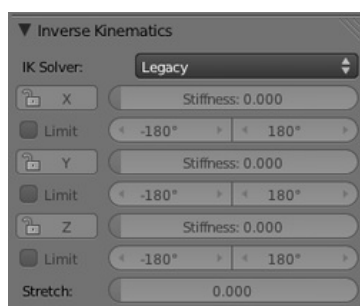
The Transform panel(pose mode).



The Transform Locks panel.

変換ロックパネル (ポーズモード)

このパネルは、ポーズモードでのみ表示され、アーマチュア内の各ボーンの軸での位置、回転およびスケールを制限することができます。



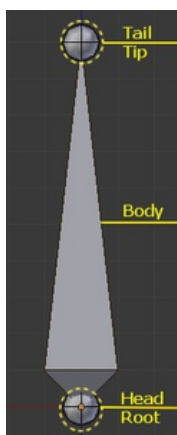
The Inverse Kinematics panel.

インバースキネマティクスパネル (ポーズモード)

このパネルでは、インバースキネマティックチェーンでリンクされた際のボーンやボーンセットの動作方法を制御します。

Bones

(ボーン)



ボーンの要素

ボーンは、アーマチュアの基本要素です。

ボーンには3つの要素があります:

- ルート や ヘッドと呼ばれる“始まりの位置”、
- “ボディ”
- チップ や テールと呼ばれる“終わりの位置”、

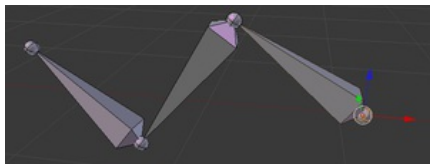
[デフォルトのアーマチュア](#) を選択して ⇨ Tab を押し 編集モードに入ります。ご覧のようにこのモードではルートとチップを選択することができ、メッシュの頂点と同じように動かせます(ただ、ここではさっと試す程度で構いません。このあとで選択および編集に特化したページがあります)。

ボーンの両端であるヘッドとテールの位置が、ボーンを定義します。ヘッドとテールには半径のプロパティもあり、エンベロープの変形メソッドには有用です(下記参照)。

Bones Visualizatio

(ボーンの可視化)

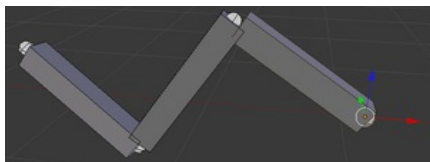
ボーンは様々な形式(八面体、スティック、B-ボーン、エンベロープ、ワイヤー)で視覚化することができ、カスタムシェイプも使えます!



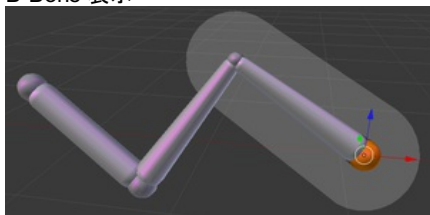
Octahedral 表示



Stick 表示



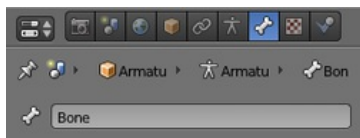
B-Bone 表示



Envelope 表示

アーマチュアはボーンでできているので、[アーマチュアの視覚化](#) の説明を読むとボーンについてさらに理解できるでしょう。

アーマチュア の Display(表示) パネルで Axes(軸) を有効にすると、各ボーンのテールにローカル座標軸が表示されます。ヘッドからテールに向けて、Y 軸はいつもボーンに沿って並びます。そのため、これはボーンの *回転軸* です。



Bone コンテキスト

Bones properties

(ボーンのプロパティ)

ボーンが選択されると(したがって 編集モード か ポーズモード) そのプロパティがプロパティウィンドウの ボーン コンテキストに表示されます。

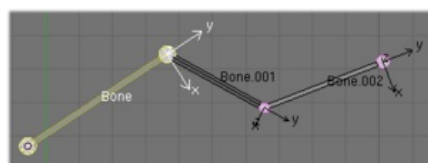
ここには選択した各ボーンの機能を操作するさまざまなパネルが表示され、パネルは作業中のモードに応じて切り替わります。

Bones Rigidity

(ボーンの剛性)

ボーンは棒のように固く途中で曲がりませんが、セグメント(節) できています。小さな要素をつなげて固定したもので、セグメント間では回転させることができます。デフォルトでは新しいボーンはどれもひとつのセグメントしか持っていないので、その範囲内で“曲げる” ことはできません。堅いボーンです。

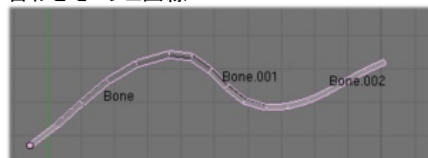
ボーンを B-bones(Bボーン) として表示すれば、セグメントは オブジェクトモード でも ポーズモード でも見ることができます。一方 編集モード ではいつも硬い棒のように描画されます。なお単独のボーンの場合は特殊で、セグメントが整列しているのでオブジェクトモード でもセグメントは見えません。



編集モードでの B-ボーンのアーマチュア



ハンドルをボーンの終端にあわせ、チェーンに重ね合わせたベジエ曲線



オブジェクトモードでの同アーマチュア

ボーンをつなげて [チェーン](#) を形成すると、Blender はボーンの全ての末端を通るベジエ曲線を計算し、チェーン内のボーンのセグメントはこの見えないカーブに従うように曲げられ、回転します。

この曲線に直接アクセスすることはできません。[編集の解説ページ](#) にあるようなボーンのプロパティを使用したある程度の制御だけができます。

図「編集モードでの B-ボーンのアーマチュア」では5つのセグメントでできたボーンを3つつなげました。これらのボーンは B-ボーン ですが、ご覧のように 編集モード では曲がらない要素のように表示されます。図「オブジェクトモードでの同アーマチュア」を参照してください。オブジェクトモード で見ると、向きを変えただけのボーンのセグメントが滑らかに一体化する様子がわかります。

もちろんチェーンの影響を受けたジオメトリは滑らかにベジエ曲線に合わせて変形します！ 実際滑らかなボーンは、たくさんの小さな堅いボーンの長いチェーンに IK を使用してポーズをつけるよりも簡単な方法です。

しかしチェーンがジオメトリよりもオブジェクトに影響を与える場合は、セグメントの方向は考慮されません(詳細は [スキニング](#) で説明されています)。B-ボーンとして表示していないときは、ボーンのセグメントが依然として存在し、効果を及ぼしている場合でも、ボーンは常に堅い棒として表示されます ([ObData へのスキニング \(eng\)](#) 参照)。

これは例えば 八面体 表示にしているときにチェーン内の一部のボーンに複数のセグメントがあっても、滑らかにジオメトリが変形されるということです。

Bones influence

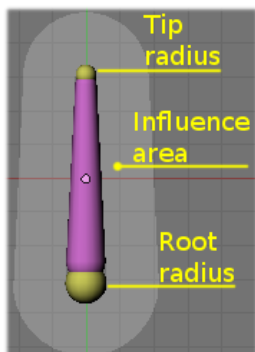
(ボーンの影響)

基本的にボーンは、頂点がボーンに「従う」ときにジオメトリを操作します。ちょうど指を動かしたとき、指の骨にあわせて指の筋肉と皮が動くようなものです。

このためにはボーンが特定の頂点に **どれだけ** 影響するか定義する必要があります。

最もシンプルな方法はジオメトリの、ボーンからの指定範囲内にある部分にボーンに影響させるものです。各ボーンがそれ自身の影響範囲に “包まれた

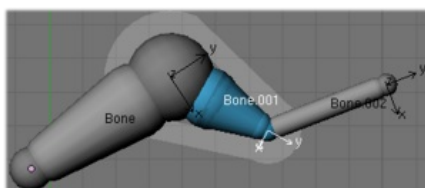
(enveloped)” ジオメトリだけを操作できるので、これはエンベロープ技法と呼ばれています。



編集モードでの Envelope
表示されたボーン

ボーンが Envelopeとして可視化される場合、編集モード および ポーズモード では、次の項目によって決まる影響範囲を見ることができます:

- Distance (距離) のプロパティ
- ヘッドの半径とテールの半径

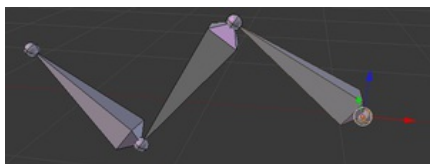


ポーズモードでの Envelope 表示されたアーマ
チュア

すべてのこれらの影響力のパラメータは [スキニングのページ](#) で詳しく説明されています。

アーマチュアの表示方法

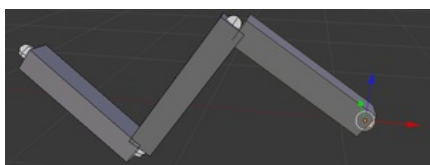
ボーンには4つの基本的な表示法のタイプがあります:八面体、スティック、B-ボーン、およびエンベロープ:



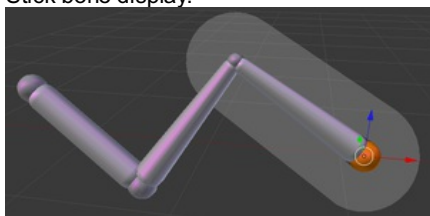
Octahedral bone display.



Stick bone display.



Stick bone display.



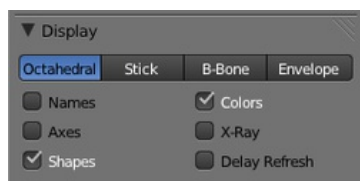
Stick bone display.

ディスプレイパネル

Mode: Object, Edit and Pose modes

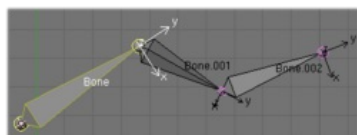
Panel: Display Object Data context

ですが、まずはObject dataコンテキストのDisplayパネルにある、アーマチュアのプロパティの一般的な可視化のいくつかを、まず見てみましょう。



The Display panel.

ボーンの種類



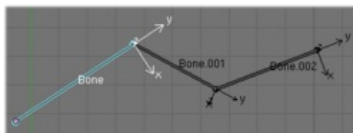
A basic armature in Octahedron visualization, Edit mode.

Note the **40°** rolled Bone.001 bone.

八面体のボーン

これはデフォルトの可視化で、ほとんどの編集作業によく適します。それは具体化します:

- ボーンのルート (“大きい”端部)、ティップ (“小さい”端部)。
- ボーンの “サイズ” (その厚さはその長さに比例します)。
- ボーンの回転 (as it has a square section)。

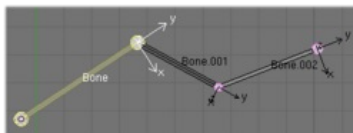


The same armature in Stick visualization, Pose mode.

Note that `Bone.001` roll angle is not visible (except by its XZ axes).

棒状のボーン

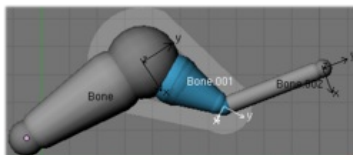
これは最も単純で、最もでしゃばらない視覚化です。単に定数 (および細い) 太さの棒でボーンを具体化するので、ルートとティップ、またボーンの色やロール角についての情報は付与しません。



The same armature in B-Bone visualization, Edit mode.

B-Bone bone

この可視化は、“滑らか”なマルチセグメント化されたボーンのカーブを示し、詳細については [bone page](#) を参照してください。



The Bone Groups panel.

エンベロープのボーン

この可視化は、ボーンの変形の影響を具体化します。これについてのより多くのことは [bone page](#) で。

属性

Names

有効にすると、各ボーンの名前が描かれます。

Colors

これは、ポーズモードにのみ関わり、[ここで](#) 詳細に説明されています。

Axes

有効にすると、それぞれのボーン (ローカル) の軸が描画されます (エディットモードとポーズモードにのみ関連)。

X-Ray

有効にすると、アーマチュアのボーンは常にソリッドオブジェクト (メッシュ、サーフェス、...) の上に描画されます – すなわち、それらは常に可視化され、かつ選択可能になります (これはオブジェクトデータコンテキストのディスプレイパネルにあるものと同じオプションです。ワイヤフレームモード以外では非常に有用です)。

Shapes

有効にすると、デフォルトの標準のボーン (棒) の形状を、オブジェクトモードとポーズモードで、選択したオブジェクトの形状に置き換えます (詳細については [以下](#) を参照)。

Delay Refresh

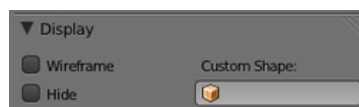
有効にすると、ポーズモードでボーンを操作するときに、子を変形しなくなります

Shaped Bones

Mode: Object and Pose modes

Panel: Display panel from Bone context.

ブレンダーでは、アーマチュアのそれぞれのボーンに特定の形状を与え (オブジェクトモードとポーズモード)、“テンプレート”として別のオブジェクトに使用することができます。まず第一に、Shapesボタンを有効にする必要があります (Armature パネル)。



The Display panel.

属性

Wireframe

有効にすると、ボーンはビューポートの描画モードに関係なくワイヤフレームモードで表示されます。非閉塞にカスタムしたボーンチェーンには便利です。

Hide

エディットモードでない場合に、ボーンを表示しません。

Custom Shape

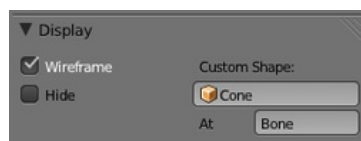
選択したボーンのカスタムシェイプを定義するオブジェクト。

Custom At

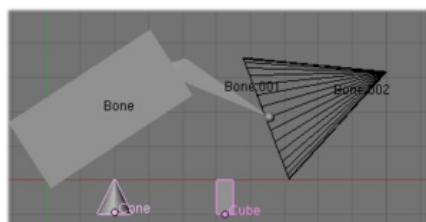
ボーン定義すると、この形のボーンに変換して表示します

ボーンへカスタム図形を割り当てるには、以下のようにする必要があります:

- ポーズモードに切り替えます (Ctrl⇧ Tab)。
- 関連するボーンを選択してください (RMB をクリック)。
- ディスプレイパネルの Custom Shape フィールドに移動し、以前に現場で作成した3Dオブジェクトを選択します。この例では立方体と円錐を使用しています。必要に応じて、別のボーンにフィールドで設定することができます。

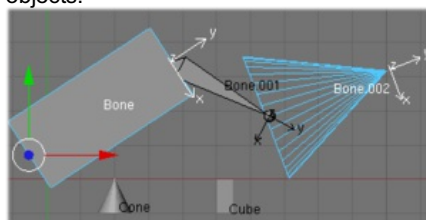


The Display panel.



The armature with shapes assigned to two bones, in Object mode.

Note the centers of the Cone and Cube objects.



The same armature in Pose mode...

Note that:

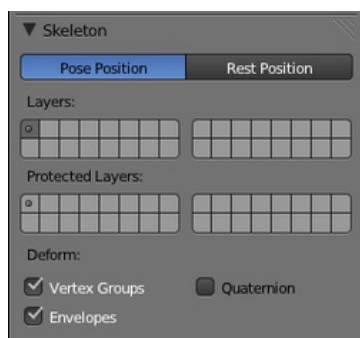
- これらの図形がレンダリングされることはありません – すべてのボーンと同様に、それらは3Dビューでのみ表示されます。
- たとえ任意の型のオブジェクトが OB フィールド (メッシュ、カーブ、さらにメタ情報...)によって受け入れているように見えても、唯一メッシュだけが実際に作用します – 他のすべてのタイプではボーンは見えなくなり、何も描かれません...
- 図形オブジェクトの中心が ボーンのルートになります ([bone page](#)の root/tip を参照)。
- オブジェクトの形状のプロパティは無視されます (すなわち、あなたがObjectモードで立方体の寸法を変更して直方体を作っても、立方体状のボーンのままでしょう...)
- “ボーンに沿う”軸はYで、形状のオブジェクトは常に、Blender のユニットがボーンの全体の長さに沿うように引き伸ばされてスケールされます。
- ボーンのカスタムシェイプを削除する必要がある場合は、Custom Shape フィールドを右クリックして、ポップアップメニューの Reset to default value を選択するだけです。

このすべての要約として、形状のオブジェクトのメッシュは低い側のY端を中心に、および全体的なYの長さを **1.0 BU**、として使用する必要があります。

アーマチュアレイヤー

Mode: Object, Edit and Pose modes

Panel: Skeleton panel, Object data context




The Skeleton panel.


各アーマチュアは32の“Armature layers”を持っていて、レイヤー内のボーンセットを“再編”して整理でき、シーンレイヤーに似た動作をします (それらはあなたのオブジェクトを含む)。さらに特定のレイヤーにボーンを“移動”でき、一つまたは複数のレイヤーを表示または非表示にすることもできます。

ボーンのレイヤーの表示/非表示

アクティブなレイヤーのボーンだけが編集可能/表示可能です – しかし、常にそれらはアクティブなレイヤーかどうかに影響されます (つまり、オブジェクトの移動や形状の変形)。レイヤーを(非)アクティブにするには、現在使用中のモードに応じて、いくつかのオプションがあります:

- すべてのモードでは、Armature パネルの上部にある Display Options グループの小さなボタンの列を使用してください。一度に複数のレイヤーを有効/無効にする場合は、いつものように **⇧** Shift キーを押したままクリックします ...
- エディットモードとポーズモードでは、3D Viewからこの操作を行うことができ、メニューを使うか (Armature » Switch Armature Layers or Pose » Switch Armature Layers)、または **⇧** ShiftM ショートカットで、上記と同じボタンを含む小さなポップアップを表示させます (ここでもまた、**⇧** Shift LMB  クリックで複数のレイヤーを一度に (非)選択 できます)。

保護されたレイヤー

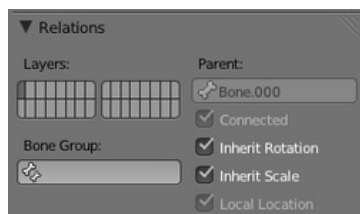
アーマチュアのすべての [代理](#) として、特定のボーンのレイヤーをロックすることができます。すなわちレイヤー内のすべてのボーンを編集できなくします。これを行うには、Skeleton のパネルで、Ctrl LMB  で該当するボタンをクリックすれば、ロックされたレイヤーが有効になります。

プロキシで保護されたレイヤーは、リロードするとファイルのプロキシ設定に復元され、元に戻ります。

ボーンレイヤー


Mode: Object, Edit and Pose modes

Panel: Relations panel Bone context



The Relations panel.

レイヤー間のボーンの移動

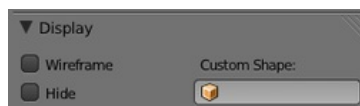
当然、エディットモードや ポーズモードでレイヤー間のボーンの移動をしなければならないこともあるはずですが – オブジェクトのように、ボーンは一度に複数のレイヤーにレイアウトすることも覚えておきましょう。いつものように **⇧** Shift LMB  クリックを使用するだけです... まず第一に、決めたボーンを選択する必要があります!

- Buttonウィンドウで、選択した各ボーン“サブパネル” (Armature Bones パネル)に “layer buttons” を使用し、それをどのレイヤーに置くのかを制御します。
- 3D View ウィンドウで、メニューを使用するか (Armature » Move Bone To Layer or Pose » Move Bone To Layer)、または M と打って通常のポップアップレイヤーダイアログを表示させます。この方法は、あなたが選択したすべてのボーンに同じレイヤーを割り当てることに注意してください。

ボーンを隠す

Mode: Edit and Pose modes

Panel: Display panel, Bone context



The Display panel.

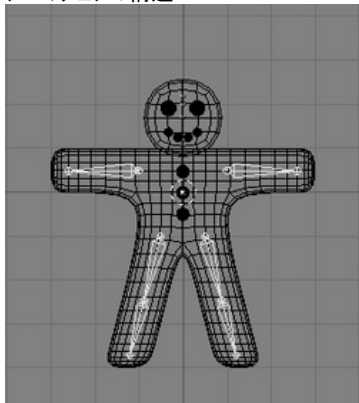
ボーンが表示/非表示切り替えにボーンレイヤーを使用する必要はありません。オブジェクトと同様に、頂点またはコントロールポイントに、Hキーを使用できます:

- Hで選択したボーン(複数可)を非表示にできます。
- \diamond ShiftHですべてのボーンを選択しているもの(複数可)を非表示にします。
- AltHですべての隠されたボーンが表示されます。

また、DisplayパネルのBoneコンテキストのHide チェックボタンを使用することもできます)。

隠されたボーンは、モードに固有であることに注意してください – つまりエディットモードで、あるボーンを隠すことができても、それらはポーズモードではまだ表示されたままで、そしてその逆も同様です。ポーズモードで隠されたボーンは、Objectモードでもまた不可視です。そしてエディットモードでは、ボーンを非表示にするには、そのルートやタイプだけでなく、全体を選択する必要があります...

アーマチュアの構造



The very basic armature of the [Gingerbread Man tutorial](#).

前に言ったように、アーマチュアは現実の骨格を模倣しています。それらは、(デフォルトでは)剛体要素であるボーンから作られています。しかし、あなたは現実の骨格よりも多くの可能性を持っています... 骨の“自然な”回転に加えて、さらに位置を変えたり拡大縮小することもできるのです! そしてあなたのボーンは必ずしも相互に接続する必要はありません、必要に応じてそれらは完全にフリーにすることができます。しかし、最も自然で便利なセットアップでは、ボーンはお互いの連携を伴い、いわゆる“chains of bones”を形成し、アーマチュアにある種の“limbs”を形成します。詳細は [以下](#)を。

ボーンのチェーン



An armature with two chains of bones.

アーマチュア内部のボーンは互いに完全に独立にすることができます (すなわち、一方への変更が他に影響しない)... しかし、これが有用なセットアップはあまりありません! 足のようなものを作成するときには、の“後に続く”すべての骨が、親子として“一緒に”動いて欲しくなります。さて、まさにこのときにアーマチュアに何が起るかというと – 互いに“親子”のボーンによって、“リム”が作成され、Blender ではこれを“chains of bones”と呼んでいます。これらのチェーンはもちろん分岐でき、例えば、5本の指として片“手”のボーンに接続されます...

ボーンは、子のルートに親のタイプをつなぐことにより連鎖します。ルートとタイプは *接続*でき、すなわちそれらを常にぴったり同じ点にしたり、またそれらを標準的な親子関係のように、*フリー*にすることもできます。

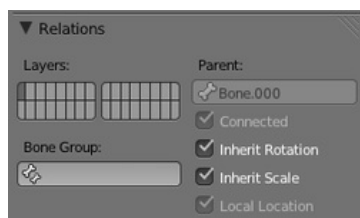
同じボーンを複数の子の親にでき、そしてそれ故に、同時に複数のチェーンの一部にすることができます。

チェーンの先頭にあるボーンは *ルートボーン*と呼ばれ、チェーンの最後のボーンは *ティップボーン*です (ボーンの端部の名前と混同しないように...).

ボーンの鎖は、[ポーズング](#)において特に重要なトピックです(特に、標準的な *フォワードキネマティクス*と、それに対する“自動”的な *インバースキネマティクス*というポーズングテクニックでは...). エディットモードでそれらを作成/編集できますが、接続されているボーンを除いて、その関係はこのモードでのボーンの変形に影響を与えません (すなわち、親ボーンを変形させてもその子には影響しません...).

ボーンの関係の編集

これは、[編集のページ](#)に詳述されていますが、ここではこの重要な機能について簡単に見てみましょう。



The Armature Bones panel with two bones selected, and their Child of settings highlighted.

ボーンの間を管理する最も簡単な方法は、Relations パネルの Bone コンテキストを使用することです:

- 最初に、編集したいボーンを **選択**します (順序はここでは関係しません)。
- ボーンを別のボーンの *親*にするために、この親の名前をそのドロップダウン Parent リスト内で選択します。
- ボーンを *unparent*するには、Parent リスト内のvoidのエントリを選択するだけです。
- ボーンを親に *接続*するには、その小さな Con ボタンを有効にします。
- ボーンを *disconnect*するには、その Con ボタンを無効にします。

アーマチュアの要素を選択する

Mode: Edit mode

Panel: Bone panel

エディットモードとポーズモードではアーマチュアのボーンを選択 (と編集) をすることができます。

このページでは、エディットモードでボーンを選択する方法を見ていきますが、ポーズモードでの選択も非常によく似ています (いくつかの特殊なところは [posing part](#) で詳述します)。

エディットモードでは、メッシュのエッジと同様に、ボーン全体を選択する2つの方法があります:

- 直接ボーンを選択する、または
- その末端の両方を選択する

メッシュの [頂点/辺の選択](#) と非常に類似しています。

これは理解に重要なポイントで、ボーンの末端の選択は、非自明な動作につながる可能性があるため、実際にどのボーンを選択しているのかに注意し、よく確認してください。

またメッシュの描画タイプとは異なり、アーマチュアの描画タイプは選択の動作に影響しないことにも気をつけましょう: それは、あなたがボーンの先端を同じ方法で選択する時はいつでも、あなたの選んだようにボーンを可視化することです。

ボーンの端を選択する

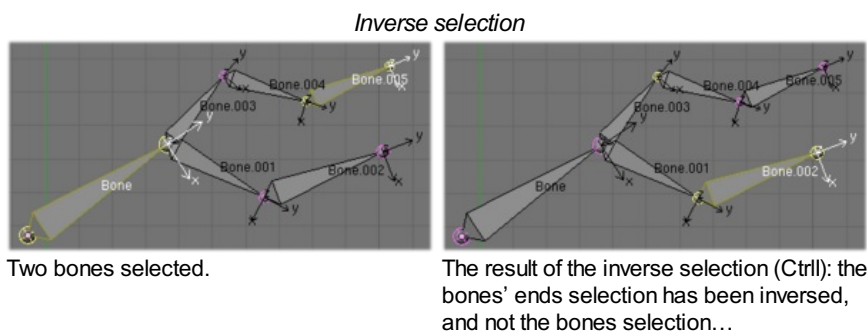
ボーンの端の選択には、標準の選択方法があります。

action	shortcut	menu	mouse
ボーンの端を選択します。			RMB それを右クリック
現在の選択から追加または削除します。			⇧ Shift RMB
すべてのボーンの両端を(非)選択します	A	Select » Select/Deselect All	
現在の選択を反転	Ctrl	Select » Inverse	
ボックス選択のツールを ONにする	B	Select » Border Select	
ボックス選択			LMB でクリックとドラッグし、現在の選択に追加したい端の周りをボックスで囲みます LMB でクリックとドラッグし、現在の選択から削除する LMB を離して有効にします Esc か RMB クリックでキャンセル
ボックス選択ツールを OFF	B or Esc		RMB
なげなわ選択			Ctrl LMB でクリックとドラッグし、現在の選択に追加したい端の周りを投げ縄で囲みます Ctrl⇧ Shift LMB クリックとドラッグで、現在の選択から削除する LMB を離して有効にします Esc または RMB クリックでキャンセル

選択範囲を反転

上で述べたように、これらの選択ツールはボーンの端だけのものであることを覚えておく必要があります: そうしないと、結果に混乱するかもしれません。

例えば、選択の反転オプション (Ctrl) は、ボーンではなくボーンの端の選択を反転します (*Inverse selection* 参照)。



接続されたボーンの両端を選択する

別の例は次の通り: その親に接続されているボーンのルートを選択すると、あなたは暗黙的にこの親のティップも選択します (またその逆も同様)。

Remember: ボーンの端を選択するとき、親のボーンのティップは、その子のボーンのルートと“同じもの”です。

ボーンを選択

ボーンのボディを RMB でクリックすると、それを選択します (それゆえ、暗黙のうちにそのルートとティップを選択します)。

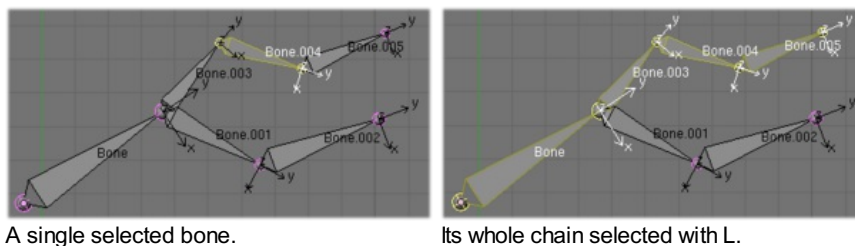
選択した各ボーンは Armature Bones パネルのサブパネルに対応します (Editing context, F9)。これらのサブパネルには、いくつかのボーンのプロパティの設定が含まれています (これに関しては、例えばボーン間の関係や、ボーンのジオメトリ変形への影響、など)、後で見えてきます。

⇧ Shift RMB を使用すると、選択範囲から追加/削除できます。

また、それらの関係に基づいた、いくつかの **先進的な選択オプション** もあります。

アクティブな(最後に選択した)ボーンが属しているチェーンの、すべてのボーンを一度に選択するには *linked selection*、Lを使用できます。

Linked bones selection



A single selected bone.

Its whole chain selected with L.

アクティブなボーンを選択解除や、その直接の親や子それぞれの選択は Select » Select Parent (I) または Select » Select Child (J)で行えます。選択中のボーンをアクティブのままにしたい場合は、Select » Extend Select Parent (CtrlI) または Select » Extend Select Child (CtrlJ)を使用します。

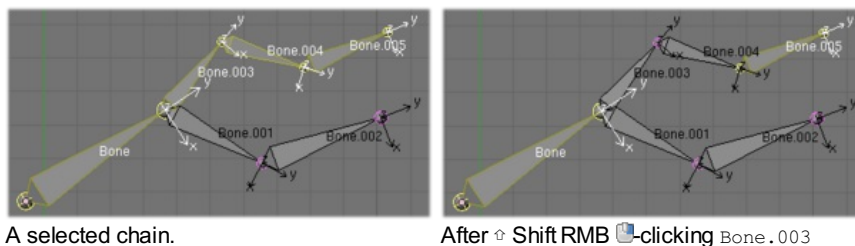
接続されているボーンを選択解除

接続されたボーンについては微妙な点があります。

接続されたボーンが複数選択されているとき、もしあるボーンを選択を解除すると、タイプはその通りに選択解除されますが、そのルートは別の選択されたボーンのタイプでもあるため、解除されません

*Bone deselection in a selected chain*を見てください。

Bone deselection in a selected chain



A selected chain.

After ⇧ Shift RMB -clicking Bone.003

After ⇧ Shift RMB -clicking Bone.003:

- Bone.003のタイプは選択解除されており、それ故に Bone.004のルートも
- Bone.003のルートの選択が解除されていないのは、Bone が Bone.003の親で、まだ選択されたままだから

アーマチュアの編集

Mode: Edit mode

Hotkey: ⇆ Tab

他のオブジェクトと同様に、エディットモード (⇆ Tab) でアーマチュアを編集します。

アーマチュアの編集は、2つの主要な行動の領域を意味します:

- [ボーンの編集](#) – 言い換えると、追加/挿入/削除/押し出し/分割/それらの結合...
- [ボーンのプロパティの編集](#) – これにはトランスフォームプロパティ(すなわちグラフ、スケール、等...) やボーン間の関係 (親子化や接続)のような主要機能が含まれるだけでなく、ボーンの名前、影響、ポーズモードでの振る舞い、なども含みます

standard editing methodsというものがあり、前例の [meshes](#) の編集に非常に類似しています。また Blender の特色として、[Etch-a-Ton](#)と呼ばれる、より高度な“アーマチュア・スケッチ”ツールがあります。同ツールは[templating](#)でもまた使用されることがあります、すなわち他のアーマチュアを現在のものへのテンプレートとして使用します...



アーマチュアの編集について理解するうえで重要なひとつのことは、あなたが編集するのはアーマチュアの **rest position**、すなわちその“デフォルトの状態”だということです。アーマチュアの *rest position* では、すべてのボーンは回転なしでボーン自身のローカル空間に **1.0** にスケールされます。

この *rest position* に基づく別の [ポーズ](#) を後で作成するかもしれませんが – もしあなたが エディットモードでそれを修正すると、すべての既存のポーズも変更されます。したがって、通常あなたのアーマチュアの [スキン](#) と [ポーズ](#) を開始する前に、決定的であることを確認してください!!



ボーンの末端に何かツールが作用しているときは、ボーン自体への他の作用に注意してください。混乱しないように気をつけて。

ボーンの編集


Mode: Edit mode

Hotkey: ⇆ Tab

ここでは、ボーンを [追加](#), [削除](#), [細分割](#) する方法を学びます。Editモードでの、[ボーンのゆがみを防止する方法](#)や、X軸に沿って編集の動きを [自動ミラーリング](#) するオプション機能も見ていきます。

ボーンを追加

アーマチュアへのボーン追加には、メッシュを編集する際とだいたい同じオプションがあります:

- 追加メニュー,
- 押し出し,
- Ctrl LMB  クリック,
- 関節の間を埋める,
- 複製.

追加メニュー

Mode: Edit mode

Hotkey: ⇆ ShiftA

3Dビューで、⇆ ShiftA » Bone でアーマチュアに新しくボーンを追加します。

このボーンは、次のようになります:

- Blender単位で1の長さ,
- ビューの正のY軸方向,
- そのルートは3Dカーソルの位置
- アーマチュアの他のボーンとは何の関係も持っていない.

突き出し

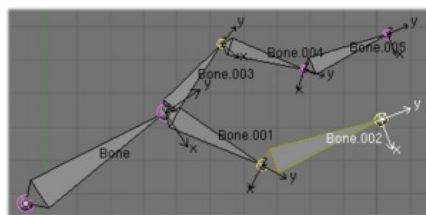
Mode: Edit mode

Hotkey: E, ⇆ ShiftE

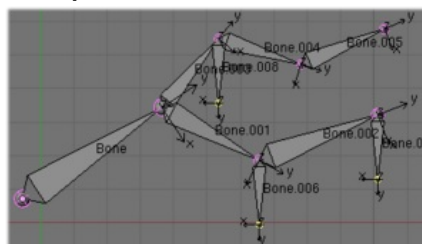
Menu: Armature » Extrude

E キーを押すと、選択している各ティップ (明示的か暗黙的にかかわらず)に、新しいボーンが作成されます。このボーンはティップ“自身の”所有者の子になり、そしてそれに接続されます。いつものように、いったん押し出しを行うと、新しいボーンのティップだけが選択され、グラブモードで、あなたの好みに合わせて配置することができます。 (*Extrusion example*)参照。

Extrusion example



An armature with three selected tips.



The three extruded bones.

また、回転/拡大縮小をとまぬ押し出しも使用でき、[here](#)でメッシュについて説明したように、ER and ESをそれぞれを押すことで行えます – グローバルまたはローカル軸に沿って [ロックされた](#) 押し出しも同様です。

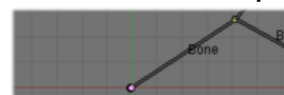
ボーンには特別な“ミラー押し出し”ツールがあり、⇆ ShiftEで呼び出します。デフォルトでは、標準的な押し出しと同様に振る舞います。しかし、ひとたびX軸ミラー編集オプションを有効にすると ([下](#) 参照)、押し出されたそれぞれのティップは 二つの新しいボーンを生成し、末尾の“_L_ R”を除いて同じ名前を持ちます (left/rightにあたる、[next page](#)参照)。“_L_”ボーンは、デフォルトの押し出しによって生成されたものとして振る舞います – まったく同じようにグラブ/回転/拡張することができます。“_R_”ボーンは、そのミラーの対応となります (アーマチュアのローカルX軸に沿って)、 (*Mirror extrusion example*)参照。



注意点として、メッシュ編集とまったく同様、Esc を押した直後に E を押した場合、押し出されたボーンがそこにあっても、その長さがゼロになり、これはあなたの頭痛の種となる可能性が非常に高い。問題にすぐに気づいた場合は、CtrlZを押すことで元に戻せます。

不思議に思われるかもしれませんが、X を押すだけではメッシュの編集で行うような解決はできず、その理由は押し出し時は新しく作成されたティップが選択されているため、下で説明するようにdeleteコマンドはボーンの末端を無視します。アンドゥを行わずにこれらの押し出したボーンを取り除くためには、ティップを移動しなければならず、その後ボーンを選択してそれらを [delete](#) します。

Mirror extrusion example



A single selected bone's tip.



The two mirror-extruded bones.


Mouse Clicks

Mode: Edit mode

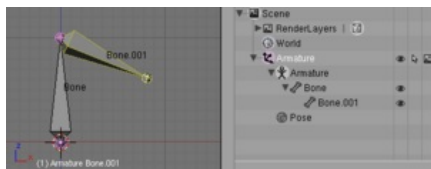
Hotkey: Ctrl LMB 

少なくとも一つのボーンが選択されている場合、Ctrl LMB -clicking すると新しいボーンを追加します。

新しいボーンのティップについて:

- この後 Ctrl LMB -clicked すると、それをアーマチュアのアクティブな要素とします
- あなたがクリックした場所のすぐそばに現れますが、しかし...
- ...(メッシュの編集と同様)ビューと3Dカーソルを通る平行な平面上に置かれます。

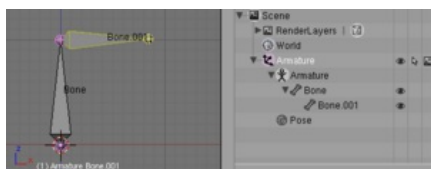
ルートの位置と新しいボーンのパレンティングは、アクティブな要素に依存します:



Ctrl-clicking when the active element is a **bone**

アクティブな要素が **ボーン** の場合は

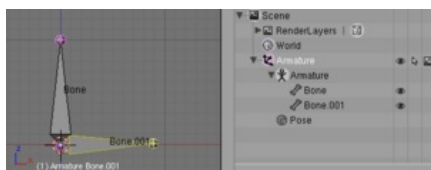
- 新しいボーンのルートはアクティブボーンのティップに配置されます
- 新しいボーンは、アクティブボーンに親子化され、かつ接続されます (*Ctrl-clicking when the active element is a bone* で概要を確認してください)。



Ctrl-clicking when the active element is a **tip**

アクティブな要素が **ティップ** である場合:

- 新しいボーンのルートはアクティブなティップに配置されます
- 新しいボーンは、ボーンの持つアクティブな先端と親子化され、かつ接続されます (*Ctrl-clicking when the active element is a tip* で概要を確認してください)。

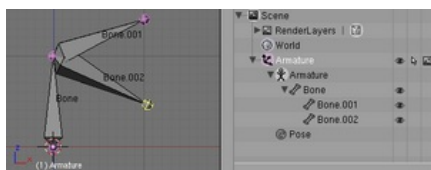


Ctrl-clicking when the active element is a **disconnected root**

アクティブな要素が **未接続のルート** である場合:

- 新しいボーンのルートがアクティブなルートの上に配置され
- 新しいボーンは、アクティブなルートを所有しているボーンとはペアレント化 **されません** (*Ctrl-clicking when the active element is a disconnected root* 概要を確認してください)。

それゆえ、新しいボーンは**いずれのボーンにも接続されません**。



Ctrl-clicking when the active element is a **connected root**

アクティブな要素が **接続されたルート** である場合:

- 新しいボーンのルートはアクティブなルートの上に配置され
- 新しいボーンが、アクティブなルートを所有している親ボーンに親子化され接続されます (check the outliner in *Ctrl-clicking when the*

active element is a connected root).

これは明白でしょう。アクティブ要素が既接続のルートの場合、親ボーンのタイプもまたアクティブな要素であるためです。ですから二番目のケースと同じです。

新しいボーンのタイプがアクティブ要素になっているとき、これらの ctrl-clicks を何度か繰り返すことで、再帰的に同じチェーンの最後に複数のボーンを追加できます。

関節の間を埋める

Mode: Edit mode

Hotkey: F

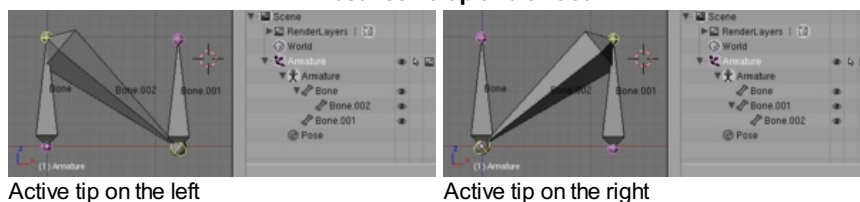
Menu: Armature » Fill Between Joints

このツールの主な用途は、F 押すことにより選択した2つの端部の間に1つのボーンを作成することで、メッシュの編集における“エッジ/フェースの作成”に似ています。

選択しているのが1つのルートと1つのタイプの場合、新しいボーンは:

- 選択されたタイプにルートが置かれます
- 選択されたルートにタイプが置かれます
- 選択された先端を所有しているボーンに親子化され接続されます

Fill between a tip and a root



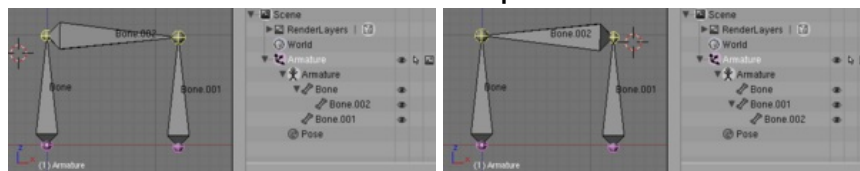
Active tip on the left

Active tip on the right

選択しているのが2つのタイプの場合、新たなボーンは:

- 選択したタイプのうち3Dカーソルに最も近いほうにルートを配置します
- 他方の選択したタイプにはタイプが配置されます
- 新しいボーンのルートとして使用されるタイプを所有しているほうのボーンと親子化され接続されます。

Fill between tips



3D cursor on the left

3D cursor on the right

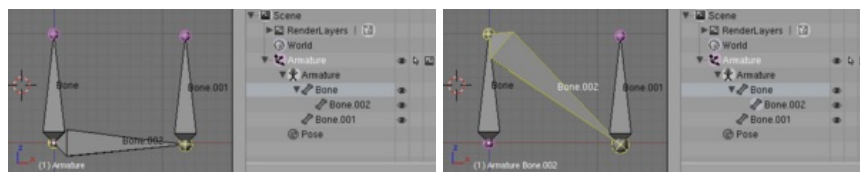
選択したルートが2つある場合、リアルタイムではインタフェースを更新しない Blender のイベントシステムに起因する小さな問題に直面するだろう。

F をクリックすると、前のケースと同様に、新しいボーンが表示されます:

- 選択されたルートのうち3Dカーソルに最も近いほうにルートが置かれます
- 他方の選択されたルートにはタイプが置かれます
- 新しいボーンのルートとして使用されるルートを所有している方のボーンと親子化され接続されます

あなたが新しいボーンを移動しようとするとき、Blender はインタフェースを更新しようとし、新しいボーンのルートが、親のボーンのタイプに移動するところを目撃することになるでしょう。

Fill between roots

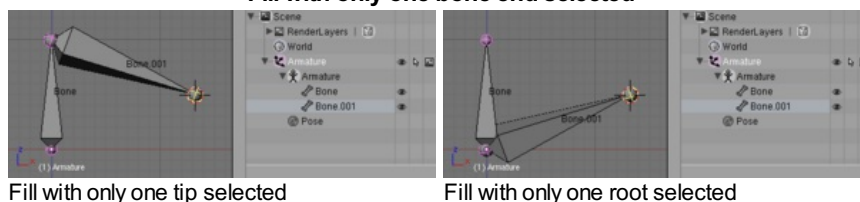


Before UI update (3D cursor on the left)

After UI update, correct visualization

選択された唯一のボーンの端で F をクリックすると、選択した端部から3Dカーソルの位置へとボーンを作成し、そしてそれはアーマチュアのどのボーンとも親子化されません。

Fill with only one bone end selected



以下のときにエラーが表示されます:

- 同じボーンの両端を埋めようとする、または
- 2つ以上のボーンの端部を埋めようとする。

複製

Mode: Edit mode

Hotkey: ⇧ ShiftD

Menu: Armature » Duplicate



このツールは選択状態のボーンに対して動作するものです: 端部が選択状態になっていても無視します。

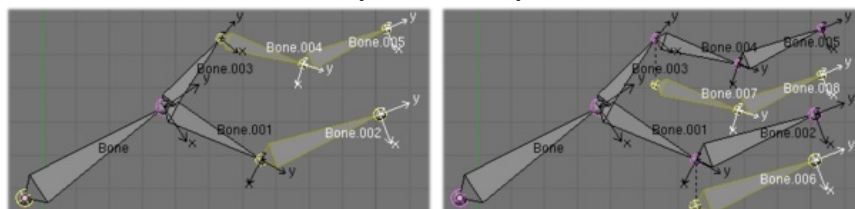
メッシュ編集と同様に、⇧ ShiftDを押すと:

- 選択状態のボーンを複製します、
- 複製したボーンは、選択した要素になり、グループモードに置かれるので、どこへでも好きな場所にそれらを移動することができます。

チェーンになっている部分を選択すると、複製によって、選択しているチェーンのコピーが得られ、コピーされたボーンは元のものとは完全に相互接続されません。

別のボーンと親子化されているボーンを複製すると、同じボーンと親子化されます、たとえルートボーンが重複時に選択されていなかったとしても。ボーンが選択されていないボーンと親子化されていてかつ接続されている場合でも、そのコピーは選択されていないボーンと親子化されては接続はされない、ということ心得ておいてください (*Duplication example* を参照)。

Duplication example



An armature with three selected bones and a selected single root.

The three duplicated bones. Note that the selected chain is preserved in the copy, and that Bone.006 (resp. Bone.007) is parented but not connected to Bone.001 (resp. Bone.003), as materialized by the black dashed lines.

ボーンの削除

アーマチュアからボーンを除去するには2つの方法があります: 標準の削除、および複数のボーンを一つにマージ。

標準的な削除

Mode: Edit mode

Hotkey: X

Menu: Armature » Delete



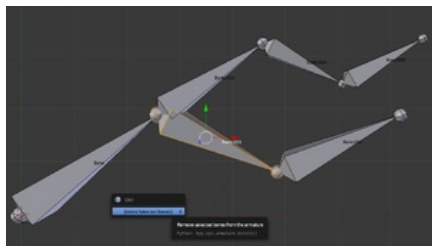
このツールは、選択状態のボーンに対して動作します: 端部が選択状態になっていても無視されます。

ボーンの削除は、次の操作で実行できます:

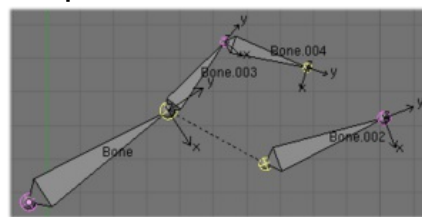
- 標準の X を押して確定するやり方、または
- メニューの Armature » Delete を使用しそして確定する。

チェーン内のボーンを削除した場合、その子(ら)は自動的に自身の親に再度親子化されますが、**しかし接続はされず**、全体のアーマチュアを歪めないようになっています。

Deletion example



An armature with two selected bones, just before deletion.



The two bones have been deleted. Note that Bone.002, previously connected to the deleted Bone.001, is now parented but not connected to Bone.

マージ

Mode: Edit mode

Hotkey: AltM

Menu: Armature » Merge

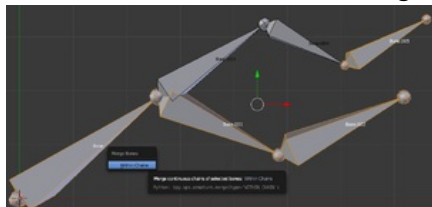
複数の選択したボーンは、**それらがチェーンを形成している限り**、一緒にマージすることができます。選択したボーンによって形成された各サブチェーンは、ひとつのボーンとなり、そのルートはルートボーンのルートとなり、そしてそのタイプはティップボーンのタイプになります。

確定するには、Merge Selected Bones ポップアップのWithin Chains をクリックします。

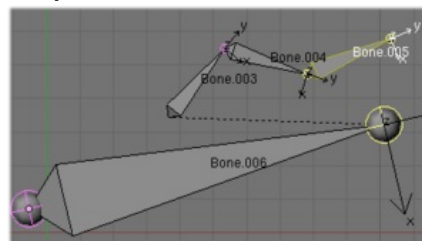
別の (非選択の) チェーンがマージされたボーンチェーンの内部から出ている場合、マージの結果として生じるボーンに親子化されます。それらが接続されていた場合、新しいボーンに接続されます。

ここで奇妙な繊細さが (*Merge example* 参照): 接続されているにもかかわらず (マージされていないチェーンのルートボーンにはルートの球がない)、ボーンは視覚的には接続されていない – これはあるボーンを編集するとすぐに行われ、どのボーンを編集するかに応じて違ってきます (これよりよく理解するためには例を参照してください)。

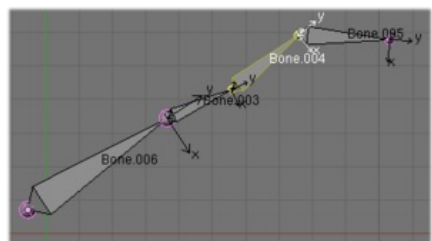
Merge example



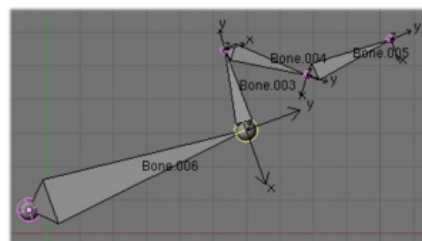
An armature with a selected chain, and a single selected bone, just before merging.



Bones Bone, Bone.001 and Bone.002 have been merged in Bone.006, whereas Bone.005 wasn't modified. Note Bone.003, connected to Bone.006 but not yet "really" connected.



Bone.004 has been rotated, and hence the tip of Bone.006 was moved to the root of Bone.003.



The tip of Bone.006 has been translated, and hence the root of Bone.003 was moved to the tip of Bone.006...

ボーンの細分化

Mode: Edit mode

Hotkey: W1, W2

Menu: Armature » Subdivide, Armature » Subdivide Multi

ボーンを分割することができ、たった一つのボーンを2つあるいはそれ以上にできます。このツールは、既存の関係を維持しながら、選択したすべてのボーンを細分化します: 分割によって作成されたボーンはいずれの場合も接続されボーンチェーンを形成します。

選択した各ボーンから2つのボーンを作成するには:

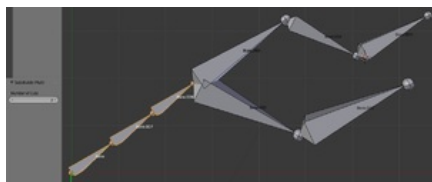
- W » Subdivideと押します、W1も同様、もしくは
- ヘッダメニューから Armature » Subdivide と選ぶ

選択した各ボーンから任意の数のボーンを作成するには:

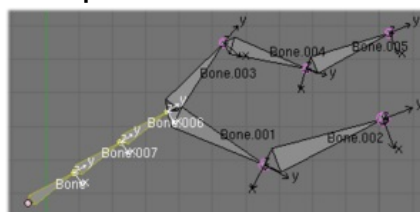
- W » Subdivide Multiと押す、W2でも同様、もしくは
- ヘッダメニューから Armature » Subdivide Multi と選ぶ、an

次に、ポップアップ内でカットの数を指定します。メッシュの編集と同様に、 n カットと設定した場合、選択した各ボーンは $n+1$ bones for each のボーンを得ることになります。

Subdivision example



An armature with one selected bone, just before multi-subdivision.



The selected bone has been "cut" two times, giving three sub-bones.

ボーンのロック

いくつかの方法でエディットモードでのボーンの変形を防ぐことができます:

- アクティブなボーンをロックするには、Transform Properties パネル (N in a 3D view) で Lock をクリックします;
- すべてのボーンをロックするには、Armature Bones パネルでそれらのサブパネルの Lock ボタンをクリックします;
- \diamond ShiftW » Toggle Settings » Locked と押す
- Armature » Bone Settings » Toggle a Setting と選ぶ).

ロックされたボーンのルートがロックされていないボーンのティップに接続されている場合、それはロックされません、つまり、あなたの好きなように移動できてしまいます。これは、接続されているボーンのチェーンでは、あるボーンをロックするとき、実際にはそのティップだけをロックすることを意味します。未接続のボーンでは、ロックはボーンの両端に有効です。

X軸ミラー編集

ほかにも非常に便利なツールとして、X-Axis Mirror 編集オプションがあり (アーマチュアがエディットモードで選択されている状態で、Tool panel > Armature Options)、[メッシュ編集ツール](#)における同ツールと少し似た働きをします。“side suffix” だけが異なる同じ名前のボーンのパアを持っているときに (たとえば `.R.L`, または `_right/_left...`)、このオプションを一旦有効にすると、あなたがボーンを変形 (move/rotate/scale...) するときいつでも、それに対応する“反対側”もそれに応じて変形され、アーマチュアのローカルX軸に沿って対称になります。ほとんどのリグは、少なくとも1つの対称軸 (動物, 人間, ...)を持つので、簡単な方法であなたの編集作業の半分を節約できます! ボーンの命名の詳細については [次のページ](#) も参照してください。

新しいArmatureのボーンの分離

あなたは、メッシュと同様に、新しいArmatureオブジェクトの中の選択されたボーンを分離することができます (Armature » Separate, CtrlAltP) – そしてもちろん、オブジェクトモードで、すべての選択したアーマチュアを1つに結合することもできます (Object » Join Objects, CtrlJ)。

ボーンのプロパティの編集

あなたはこのページで、ボーンのプロパティを編集/制御する方法のほとんどを学びます – まず先に [前のページ](#) を読んでおく必要があります! 我々は [ボーンの関係の管理](#)、[それらの名前の変更](#) などの方法を見ていきます。

ボーンの変形

ここではボーンのままの変形の詳細までは述べませんが、軸のロック、ピボットポイント、などは、ほとんどのオブジェクトの編集に共通していて、そして [ここ](#) 既に説明されています(スナップのような、一部のオプションは、それらが使用可能であっても、動作しないことに注意...)。同じことはミラーリングにも当てはまり、[メッシュ編集](#) とほとんど同様です。ただ、ボーンのルートとティップはおおよそメッシュの頂点のように動作し、ボーン自体はメッシュの辺のように動作することは覚えておいてください...

ご存知のように、ボーンの連結には2つのタイプがあります: 親子化および、接続です。親子化されたボーンのエディットモードでの振る舞いは、それらが関係を持っていない場合とまったく同じです – あなたはその子孫に影響を与えることなく、親ボーンに、グラフ、回転、スケールなどを行うことができます... しかし、接続状態のボーンは常に親のティップが子のルートに接続されている必要があり、そのためボーンの変形は、その接続されているすべての親/子供/兄弟に影響を与えます。



The Transform Properties panel for armatures in Edit mode.

最後に、選択したアクティブボーンの両端の、位置と半径の編集は、Transform Properties パネル (N) で [roll rotation](#) と同じように行えます。

エンベロープの可視化における半径とスケール

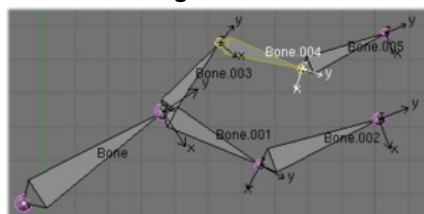
Mode: Edit mode, Envelope visualization

Hotkey: S

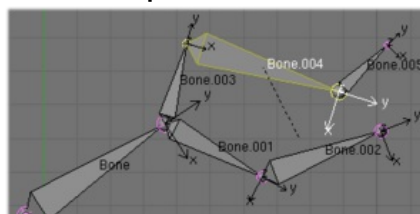
Menu: Armature » Transform » Scale

Octahedron, Stick または B-Bone 可視化でのスケールが予想通りに動作したとしても、Envelope可視化でも同じ印象をうけるとは限りません: 選択したボーンの両端の半径をスケールします、[skinning part](#) を参照してください。あなたがコントロールするのは1つの値 (半径) だけなので、軸のロックはここにはありません。そしていつものように、接続状態のボーンでは、親のティップと子のルートの半径は同時にスケールされます。

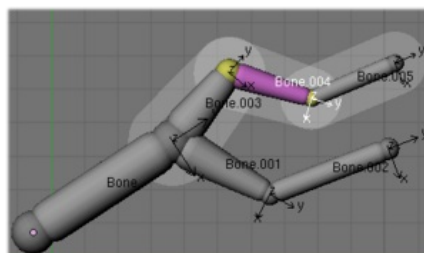
Scaling of a bone in Octahedron and Envelope visualizations.



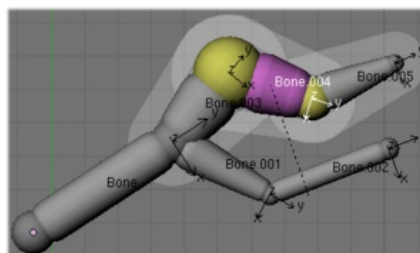
A single selected bone...



... Scaled in Octahedron visualization.



A single selected bone...



... Scaled in Envelope visualization – its length remains the same, but its ends' radius are bigger.

ボーンをリサイズする時、端部の半径は、サイズ変更按比例して、Blenderによって自動的に調整されることに注意してください (直接それをスケールするとき、またはその端の一方を移動するとき)... そのため、これらのプロパティの編集は、必要であれば、すべてのボーンがうまく配置された後のみ行うべきです!

Bとエンベロープのスケール

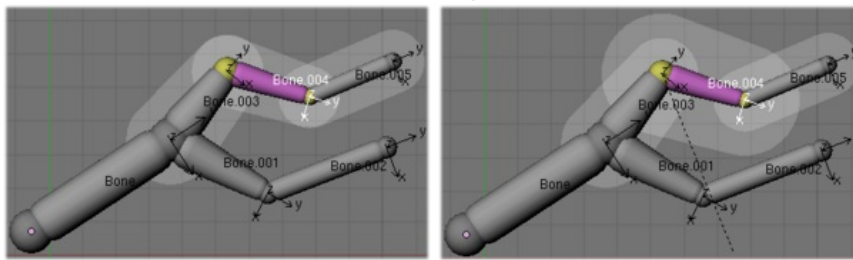
Mode: Edit mode

Hotkey: AltS

AltS はアーマチュア固有のひとつの変換ツールを起動し、そしてどの可視化がアクティブかに応じて異なる動作をします。

Envelope可視化では、選択したボーンの影響を編集することができます (それらの Dist プロパティについては、[skinning part](#)を参照してください) – この視覚化での“標準の”スケーリングと同様、これは単一値のプロパティですので、軸のロックなどはありません。

Envelope scaling example

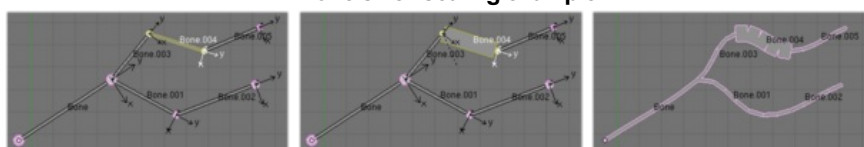


A single bone selected in Envelope visualization.

Its envelope scaled with AltS.

他のビジュアライゼーションでは、“bone size”を編集することができます。これはB-Boneの可視化だけが持つ視覚効果のように思いますが、OctahedronとStickでも利用可能です... このツールの特異性がここにもあります: 他のいずれの変換ツールでも、“ローカル軸”はオブジェクトのもので、その反面ここではそれらはボーンのもので (これが意味するのは、ローカル軸にロックするとき、関連するキーを二度押すことで選択されるのは、ボーンのローカル軸であって、アマチュアオブジェクトのものではないということです)。ついでながら、あなたがボーンを複数選択している場合、このツールでローカル軸をロックしようにしないでください: それは、Blenderをクラッシュさせる非常に効率的な方法です!

“Bone size” scaling example



A single “default size” bone selected in B-Bone visualization.

Its size scaled with AltS.

The same armature in Object mode and B-Bone visualization, with Bone.004's size scaled up.

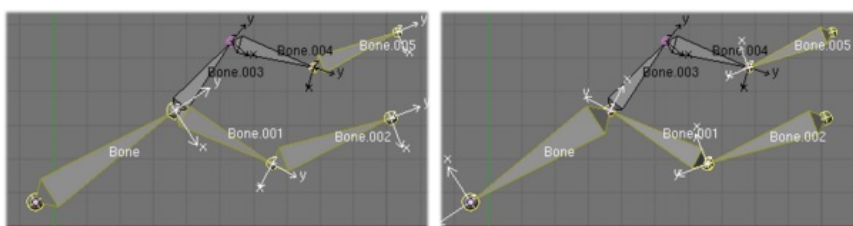
ボーンの方角

Mode: Edit mode

Hotkey: W3

Menu: Specials » Switch Direction

このツールは、Armatureメニューからは利用できず、唯一 Specials ポップアップ (W)から利用できます。選択したボーンの方角を切り替えることができます、すなわち、ルートがティップになり、そしてその逆も同様。ボーンの方角を切り替えると、通常それが属するチェーン(s)は切れます。しかし、チェーンの全体(部位)を切り替える場合は、切り替えたボーンはペアレント/接続されたままになり、“逆の順序”にされます。(Switching example)参照。



An armature with one selected bone, and one selected chain of three bones, just before switching.

The selected bones have been switched. Bone.005 is no more connected nor parented to anything. The chain of switched bones still exists, but reversed (Bone.002 is now its root, and Bone its tip). Bone.003 is now also a free bone.

Switching example.

ボーンの回転

Mode: Edit mode



Hotkey: CtrlR, CtrlN

Menu: Armature » Bone Roll » ...

Editモードでは、ボーンのロール回転(すなわちボーンのY軸周りの回転)の制御専用のオプションがあります。あなたが新しいボーンを追加するたびに、ロールを、Z軸が現在の3Dビューと可能な限り垂直になるようにします。そして、ボーンを変形するたびに、Blenderはその最高のロールを決定しようと

します...

しかしこれによりアーマチュアがわかりづらくなることもあり、すべての角度でボーンのロールが... きたない! この問題に対処するには、次の3つのオプションがあります:

- Armature » Bone Roll » Set Roll (CtrlR) で、ロール固有の回転を起動します。これは他の変換操作と同様に動作します (つまり、マウスを動かして LMB  で確定、もしくは数値をタイプしENTERを打ちます – また、RMB  クリックか Esc を叩けばすべてをキャンセルします)。
- Armature » Bone Roll » Clear Roll (Z-Axis Up) (もしくは CtrlN1 » Recalculate Bone Roll Angles » Clear Roll (Z-Axis Up)) で選択中のボーンのロールをリセットし、そのZ軸をできるだけグローバルZ軸と揃うようにします。
- Armature » Bone Roll » Roll to Cursor (もしくは CtrlN2 » Recalculate Bone Roll Angles » Align Z-Axis to 3D-Cursor) で選択中のボーンのロールをセットすると、そのZ軸は可能な限り3Dカーソルの近くを指します。

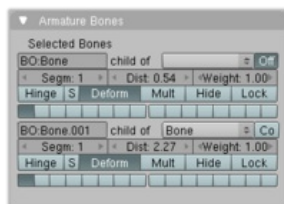
プロパティ

Mode: Edit mode

Panel: Armature Bones (Editing context, F9)

Hotkey: ⇧ ShiftW, Ctrl⇧ ShiftW, AltW

Menu: Armature » Bone Settings » ...



The Armature Bones panel in Edit mode.

ボーンのほとんどのプロパティは(変形のものを除いて)アーマチュアボーンパネルの、それぞれのボーンのサブパネルに再編成されています (Editing context, F9)。詳述していきます。

そのうちのいくつかは、3つのポップアップメニューの Toggle Setting を介して、3Dビューでも利用可能であることも覚えておきましょう(⇧ ShiftW または Armature » Bone Settings » Toggle a Setting)、(Enable Setting (Ctrl⇧ ShiftW or Armature » Bone Settings » Enable a Setting)、および (Disable Setting (AltW または Armature » Bone Settings » Disable a Setting) – 3つすべてのエントリは同じもので、それぞれの効果はすぐにはかからないはずで...

BO

ボーンの名前のフィールドです、[下文参照](#)。

child of

これらの2つの設定は、ボーンの間接関係を制御します、[下で詳述](#)。

Segm

この設定は、ボーンのセグメント数を制御します、[下文参照](#)。

Dist, Weight, Deform (also ⇧ ShiftW » Deform & co), Mult (also ⇧ ShiftW » Mult VG & co)

これらの設定では、ボーンが形状にどのように影響するかをコントロールします – ボーンの末端の半径と共に。これは [スキンのパート](#) で詳述します。

Hinge (also ⇧ ShiftW » Hinge & co), S (also ⇧ ShiftW » No Scale & co)

これらの設定は、ポーズモードでの自身の親の変形にともなう子のボーンの状態に影響するもので、そのため [ポーズモードのパート](#) で詳述します!

Hide

ボーンを非表示にします (3Dビューで H を押すのと同じです、[こちらのページ参照](#))。

Lock (also ⇧ ShiftW » Locked & co)

Edit モードでボーンの状態を編集すべてを防ぎます、[前のページ参照](#)。

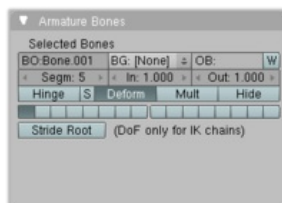
Layers button

これらの小さなボタンは、ボーンが属するどのボーンレイヤーでも制御できるようにします、[こちらのページ参照](#)。

ボーンの剛性の設定

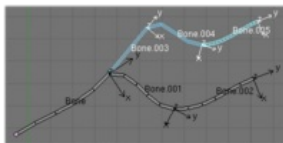
Mode: Edit and Pose modes

Panel: Armature Bones (Editing context, F9)



The Armature Bones panel in Pose mode.

Segmの設定をエディットモードで利用していたとしても(ボーンサブパネル、アーマチュアボーンパネル内)、“スムーズ”ボーンプロパティを編集するときにはポーズモード(Ctrl⇨Tab)に切り替えるべきです – なぜエディットモードではいけないのか不思議に思うかもしれないのでひとつ補足しておく、B-Bone可視化では特に、ボーンが棒のように描かれるので、これらの設定の効果を視認できないのです...



An armature in Pose mode, B-Bone visualization:

Bone.003 has one segment,

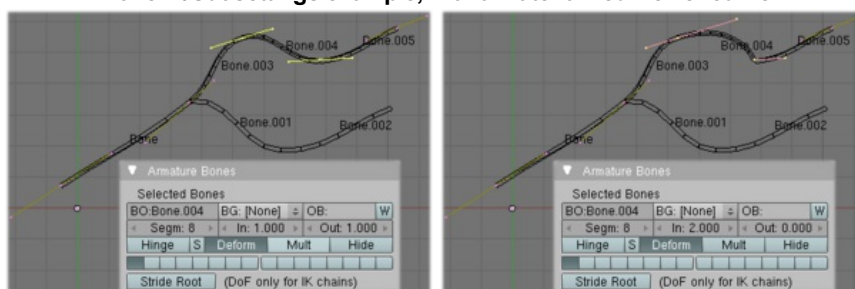
Bone.004 has four, and

Bone.005 has sixteen...

[このページ](#) で見てきたように、ボーンは“仮想”ベジエ曲線にマッピングされた小型の剛体セグメントで構成されます。Segm 数値フィールドでは、特定のボーンの内側のセグメント数を設定できます – デフォルトでは 1 で、これは標準の剛体ボーンを示します! この設定を高くすると (max **32**)、ボーンがより滑らかになり、しかしポーズの計算は重くなります...

それぞれのボーンの端は“自動”操縦でカーブします。それらの方向を制御することはできませんが、In and Out 数値フィールドを使用して、その“長さ”に影響を与えたり、ボーンの“root handle”と“tip handle”それぞれで制御することはできます。これらの値はデフォルトの長さに比例し、もちろんボーンの長さや、チェーン内の前の/次のボーンとの角度などによって自動的に変化します。

Bone In/Out settings example, with a materialized Bézier curve.



Look at Bone.004: it has the default In and Out values (**1.0**).

Bone.004 with In at **2.0**, and Out at **0.0**.

チェーンの編集

Mode: Edit mode

Panel: Armature Bones (Editing context, F9)

Hotkey: CtrlP, AltP

Menu: Armature » Parent » ...

3Dビューおよびボタンウィンドウのどちらからも、ボーン間の関係(つまりボーンチェーンの作成/変更も)を編集することができます。どれでもお好みの方法で、各ボーンを決定しさえすれば、別のボーンと親子化され、また然らば、接続されるはずで。

ボーンの親子化および/または接続は、次の操作で実行できます:

- 3Dビューで、ボーンを選択しその後将来親となるボーンも選択し、CtrlPと打ちます(もしくは Armature » Parent » Make Parent...). ポップアップ内の小さい Make Parent メニューで、子をその親に接続したい場合は Connected を選択、そうでない場合は Keep Offsetをクリック。2つ以上のボーンを選択した場合、それらはすべて最後に選択されたものにペアレントされます。既に親になっているボーンを一つだけ選択した場合、または選択したすべてのボーンがすでに最後に選択したものを親としている場合は、唯一の選択肢はそれらを接続することです、まだ行っていない場合。親でないボーンを一つだけ選択した場合、Need selected bone(s) というエラーメッセージを受け取るようになります...

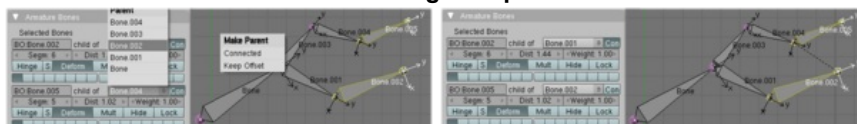
この方法では、新規の子のボーンは、拡大縮小も回転もされることはありません – それらが変換されるのは、親のタイプにそれらを接続するように選択していた場合だけです。

- ボタンウィンドウの、アーマチュアボーンパネルで、選択状態のボーンそれぞれについて、その親を選択することができ、サブパネルの右上隅にある Parent ドロップダウンリストで行えます。それらを接続したい場合は、リストの右側の小さな Con ボタンを有効にするだけです。

この方法では、子のボーンの前端は決して変換されません – ですから Con が有効になっている場合、子のボーンの変形は完全に手動になりま

す。

Parenting example.



The starting armature, with Bone.005 parented and connected to Bone.004.

Bone.005 re-parented to Bone.002, but not connected to it (same result, using either CtrlP2 in 3D view, or the Armature Bones panel settings).



Bone.005 parented and connected to Bone.002, using CtrlP1 in 3D view.

Bone.005 parented and connected to Bone.002, using the Parent drop-down list of Bone.005 sub-panel.

ボーンの切断およびまたは遊離は、次の操作で実行できます:

- 3Dビューで、目的のボーンを選択し、AltP同時押し (またはArmature » Parent » Clear Parent...). 小さな Clear Parent メニューがポップアップするので、そこで Clear Parent を選べば選択状態のボーンすべてが完全にフリーになり、またその接続を断ちたいだけの場合は Disconnect Bone を選びます。
- ボタンウィンドウの、アーマチュアのボーンパネルの、選択状態の各ボーンのサブパネルにある Parent ドロップダウンリストでは、全ての親子化を選択解除することもでき、完全にフリーな状態にもできます。単に親からそれを切断したい場合は、Con ボタンを無効にします。

非選択の子との関係が変更されることはありません。

ボーンの名前

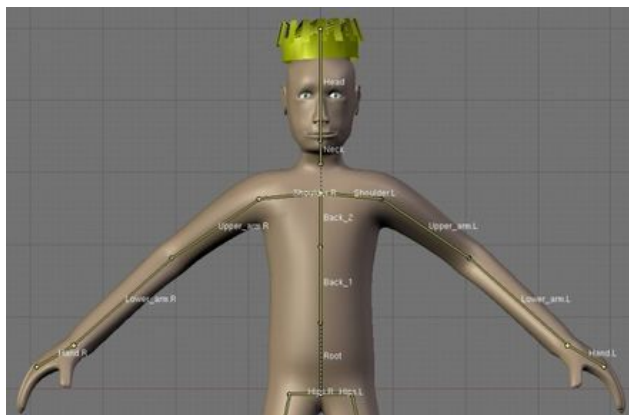
Mode: Edit mode

Panel: Armature Bones (Editing context, F9), Transform Properties (3D views, N)

ボーンの名前を変更するには、アクティブなボーンに(N)で呼び出す、3DビューのTransform Properties パネルにあるBoneフィールドを使用するか、またはArmature Bonesパネルの各ボーンのサブパネルで BO フィールドを使用します (Editing context, F9).

またBlenderでは、左/右対称の形でボーンの名前を指定したり、ほかにも自動的にアーマチュア名前をボーンに付けるいくつかのツールも導入されています。詳細を見ていきましょう。

命名規則



An example of left/right bone naming in a simple rig.

Blenderの命名規則が有用なのは、右のボーンを見つけるために使用する場合だけでなく、それらのどれが任意の2つのカウンターパートなのかをBlenderに判別させる時にも便利です。

アーマチュアの半分がミラー化されているケースでは (つまり、左右対称のときには)、左/右の命名規則に固執する価値があります。このいくつかのツールを使用できれば、あなたの時間と労力を節約できるでしょう (上で見た X軸ミラー 編集ツールのような...).

- まず、あなたのボーンに、有意義なベースネームをつける必要があります。たとえば leg, arm, finger, back, foot、等のような
- あなたが他の側面 (a pair)にコピーのあるボーンを持っている場合、armのような場合、それらには、次の区切り文字のいずれかを付与します:
 - 左/右のセパレータは、第2の位置でも (L_calfbone) または最後から一つ前 (calfbone.R) でもどちらでもかまいません
 - 小文字または大文字のある場合は L, R, left or right、Blenderが正しく反対側の部分を処理します。有効な区切り文字のリストは以下を参照してください。1つを選んだら、リギングしている間はできるだけそれに近づけて一貫しましょう。そうすればうまくゆくはずですよ。たとえば:

有効な区切り文字.

Separator	example
(nothing)	handLeft → handRight

<code>_ (underscore)</code>	<code>Hand_L</code>	→ <code>Hand_R</code>
<code>. (point)</code>	<code>hand.l</code>	→ <code>hand.r</code>
<code>- (dash)</code>	<code>Foot-l</code>	→ <code>Foot-r</code>
<code>(space)</code>	<code>pelvis LEFT</code>	→ <code>pelvis RIGHT</code>

上記すべての例で、名の前に左/右の部分の置いても有効であることも覚えておきましょう。短いL/Rコードは、区切り文字を用いる場合にだけ使用できます (つまり、`handL/handR` は動作しません!)

- Blender がアーマチュアのミラーリングまたはフリップを処理する前には、最初に数字の拡張子を削除します、それがあがる場合は (たとえば `.001` のような)
- `W` » Flip Left-Right Names を使用して、ボーンの名前を `bla.l` として反転させてコピーすることができます。Blender はコピーを `bla.l.001` と名付け、そして反転させた名前を `bla.r` とします。

ボーンの名前のフリッピング

Mode: Edit mode

Hotkey: W4

Menu: Armature » Flip Left & Right Names

選択したボーン名の左/右のマーカー (上記参照) を反転させるには、Armature » Flip Left & Right Names, or Specials » Flip Left-Right Names (W4)のどちらかを使用します。これが便利なのは、対称的なリグ (左または右側のマークが付けられた) の半分を構成したり、複製やミラー化した場合で、新しい側の名前を更新したいときです。Blender は、ボーンの名前のうち上記の命名規則に従った部分のテキストを交換し、可能であれば数値の拡張子を削除します。

ボーンの自動ネーミング

Mode: Edit mode

Hotkey: W5, W6, W7

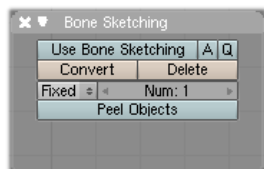
Menu: Armature » AutoName Left-Right, Armature » AutoName Front-Back, Armature » AutoName Top-Bottom

Armature メニューと Specials(W) メニューにある三つの AutoName エントリは、アーマチュアの中心とボーン自体のローカル座標を基準とする位置に基づいて、選択状態のすべてのボーンに自動的にサフィックスを追加できるようにします:

- AutoName Left-Right は、ルートが正のX座標にあるボーンすべてに `.L` 接尾辞を追加します、そしてルートが負のX座標にあるボーンすべてに `.R` 接尾辞を追加します。ルートがX-軸上のぴったり `0.0` にある場合は、タイプのX座標が使用されます。ボーンの両端が、X-軸上 `0.0` にある場合は、ボーンの L/R を判断できなくなります (Blender は、ボーンが左なのか右なのかどうかを決めることができないということです...).
- AutoName Front-Back は、ルートが正のY座標にあるボーンすべてに `.Bk` 接尾辞を追加し、ルートが負のY座標にあるボーンすべてに `.Fr` 接尾辞を追加します。Y座標 `0.0` のボーンについては AutoName Left-Right と同じことがあてはまります...
- AutoName Top-Bottom は、ルートが正のZ座標にあるボーンすべてに `.Top` 接尾辞を追加し、そしてルートが負のZ座標にあるボーンすべてに `.Bot` 接尾辞を追加します。Z座標 `0.0` のボーンについては AutoName Left-Right と同じことがあてはまります...

ボーンスケッチ

ボーンを一つ一つ手作業でリギングするのがとても面倒くさいと思っているなら、喜んでください。Blenderの開発者の中にもそう感じている人がいて、スケルトンスケッチツールを作りました。これは旧称Etch-a-tonツールと呼ばれていたもので、複数のボーンの流れを「描く」ことができるツールです。



The Bone Sketching panel in its default (inactive) state.

このツールは3DビューのEditモードのみで使え、Bone Sketchingパネル(PまたはArmature » Bone Sketching)とマウス(LMB で線を描く、⇧ Shift LMB でジェスチャ)でコントロールします。ツールパネルを表示しただけではスケッチは有効にならないので注意してください。Use Bone Sketchingボタンをクリックしてからボーンのチェーンを描き始めます。

スケッチは二つのステップで行います。

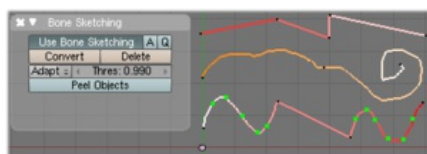
1. **滑らかな、または折れ線状のラインを描く** (「ストローク」と呼びます)。一つのストロークが一組のチェーンになります。
2. **ストロークを実際にボーンのチェーンへと変換する**。これにはいくつかの変換方法があります。

視点は重要です。これによって、ボーンが生み出されたときのロール角度が決まります。ボーンのZ軸は、その元になるストロークを描いたときの3DビューのZ軸(奥行き方向)に沿った角度になります(Template変換を使った場合以外)。ストロークは、ビューに平行で3Dカーソルを通る平面上に描かれます。ただし、後述のAdjustオプションを使うと3次元的なストロークを描くこともできます。

Quick Sketchオプションを有効にすると、上記の二つのステップが一体化します。ストロークの描画を終了すると(後述)、直ちに(現在アクティブになっている方式で)ボーンに変換され、ストロークは消去されます。このオプションをつかえばすばやく効果的にボーンスケッチができますが、その代わりストロークを編集する機会は失われます。

スケッチはBlenderファイルに保存されません。よってスケッチを中断すると作業は失われます！ また、スケッチはすべてのBlenderセッションに共通しています。すなわち、存在するストロークのセット(スケッチ)は、アーマチュアにつき一つでもなく、ファイルにつき一つでもなく、全体でたった一つだけです。

チェーンを描く



ストロークの例。上から順に:

*選択されている折れ線状ストローク。4つのストレートセグメントから成っている。ボーンは左から右につながる。

*選択されていない曲線状ストローク。2つのフリーセグメントから成っている。ボーンの流れは左から右。

*選択されているミックスストローク。1つのストレートセグメントと2つのフリーセグメントから成っている。ボーンの流れは右から左。

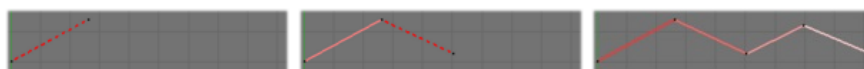
このアーマチュアにはまだボーンは無いので、オブジェクトの中心点だけが見えています(画像下部)。

描かれた一つ一つのストロークがそれぞれボーンのチェーンになります。チェーンはストロークの開始点(最も赤い、または最もオレンジの部分)から終了点(最も白い部分)に向かってつながっていきます。ストロークは黒い点で区切られたいくつかのセグメントから成っています。一つのセグメントが、少なくとも一つのボーンになります(Template変換を使う場合以外。[次ページ\(eng\)](#)参照)。よって、黒い点はすべてボーンの端を意味します。セグメントには二種類あり、どちらも一緒に使えます。

ストレートセグメント

ストレートセグメントを作るには、開始点でLmbをクリックし、それから何も押さずにマウスを動かします。赤い破線が表示され、再度Lmbをクリックするとそれがセグメントになります。

それぞれのストレートセグメントは、どの変換方法でもちょうど一つだけのボーンになります(Template変換以外)。






The first segment has been started (LMB click) and the mouse moved to its end point.

Drawing straight segments example.

The first segment has been finalized by a second LMB click, which also started a new segment...

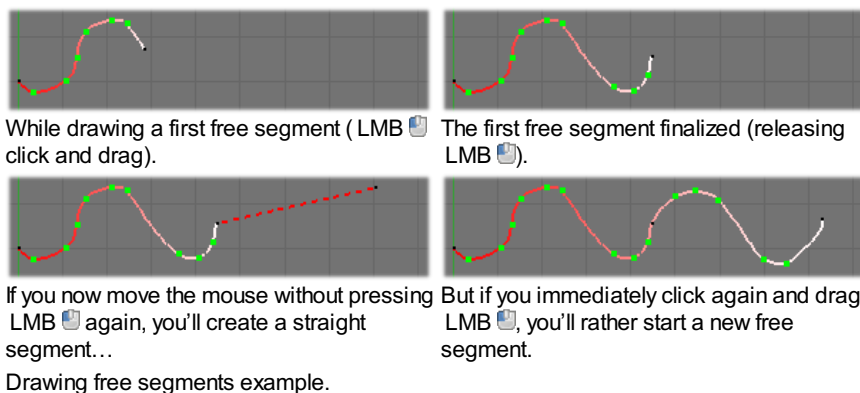
Repeating these steps, we now have a four segments polygonal stroke.


フリーセグメント

フリーセグメント(カーブ状のセグメント)を作るには、開始点でLMB  を押し、そのままペイントソフトのようにドラッグしてセグメントを描きます。そしてLMB  を離すとセグメントの描画が終了します。すると、次のストレートセグメントを描画しようとしている状態になるので、フリーセグメントをまた描きたい場合はすぐにLMB  を押しなおしてください。




フリーセグメントは、変換方法によっていくつのボーンに変換されるかが変わります。選択中のストロークに関しては、現在の変換方法でボーンの数(つなぎ目)となる位置が緑の点で示されます。

フリーセグメントの描画の際、いつどこに点が打たれるかということは、[グリースペンシル](#)と同じManhattan Dist設定で決まります(User PreferencesウィンドウのEdit Methodsパネル)。もしもフリーセグメント細かく分割されすぎると感じたら数値を上げ、粗すぎると感じたら数値を下げてください。



ひとつのストローク全体の描画を終えるときは、RMB  をクリックします。描画中のストロークをキャンセルするにはEscを押します。Ctrlを押しながら描くと、ストロークがその下のメッシュにスナップします。ところで、Bone Sketchingパネルの下部にあるPeel Objectsボタンは、ヘッダーのスナップコントロールでVolumeを選んでいる場合に現れる「猿」のボタンと同じものです。[スナップ\(eng\)](#)のページを参照


ストロークの選択

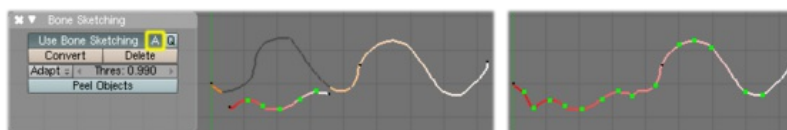
ストロークは選択(赤から白へ変わる色のライン)、および非選択(オレンジから白のライン)の状態があります。上の図"Strokes example"を見てください。他の場合と同じように、RMB  で選択、 Shift RMB  で追加選択/選択解除、Aで全選択/選択解除です。

ストロークの消去

ストロークを選択し、XまたはDeleteボタン(Bone Sketchingパネル)で消去できます(確認や警告は出ないので気をつけてください)。後述の[ジェスチャ](#)も参照。

ストロークの修正

Bone SketchingパネルのOverdraw Sketchingオプションを有効にすると、ストロークを「描きなおす」ことができます。この場合、LMB  クリックまたはドラッグすると、新しいストロークが作成されるのではなく、最も近いストロークが修正されます。この操作で置き換えられるストロークはグレーで表示されます。このオプションは選択状態とは関係ありません。つまり、どのストロークでも修正される可能性があります。また、このオプションが有効でも、存在しているストロークから遠い場所で描き始めた場合はオプションが無効の場合と同じように新しいストロークが始まります。





Adjusting a stroke: the gray part of the "unselected" (orange) stroke will be replaced by the currently drawn "replacement".

Stroke adjusted.

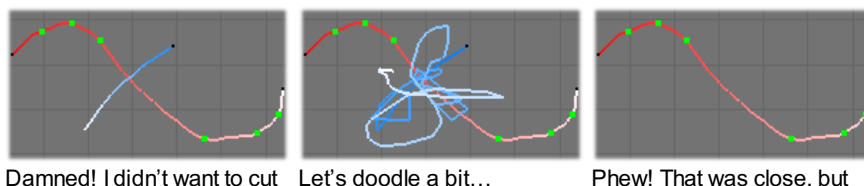
Adjusting stroke example.

最後に、ストロークは取り消し/やり直しができないので注意してください...

ジェスチャ

ストロークの編集に関して、ジェスチャを使わなければならない操作があります。ジェスチャは Shift LMB  ドラッグで開始し(ストロークの作成中以外)、青から白へ変わる色のラインで示されます。ジェスチャは複数のストロークに一度に影響を与えられます。

ジェスチャの軌道を描き始めたら、それをキャンセルする直接的な方法はありません。よって、ジェスチャの最中に考えが変わった(または間違ったラインを描いてしまった)場合は、ストロークの上をぐちゃぐちゃに殴り書きするのが最もよい方法です。ようするにジェスチャシステムが認識できないようにしてしまうわけです!



this stroke here!

the stroke is still in one piece...

Canceling gesture example.

カット

セグメントをカット(つまり黒い点を追加して二つのセグメントに分割する)には、カットしたい位置でセグメントを横切るように、まっすぐなラインを描きます。



Gesture.

Result.

Cut gesture example.

消去

ストロークを消去するには、Vの字を描いてストロークを二度横切ります。



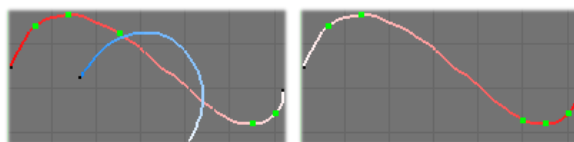
Gesture.

Result.

Delete gesture example.

反転

ストロークを反転(ボーンの流れる方向を逆にする)には、Cの字を描いてストロークを二度横切ります。



Gesture.

Result.

Reverse gesture example.

ボーンに変換する

Bone SketchingパネルのConvertボタン(または上記のジェスチャ)で、選択した一つまたは複数のストロークをボーンに変換できます。それぞれのストロークは、赤い端から白い端に向かうボーンのチェーンを生成します。ストロークをボーンに変換しても、もとのストロークは消去されません。



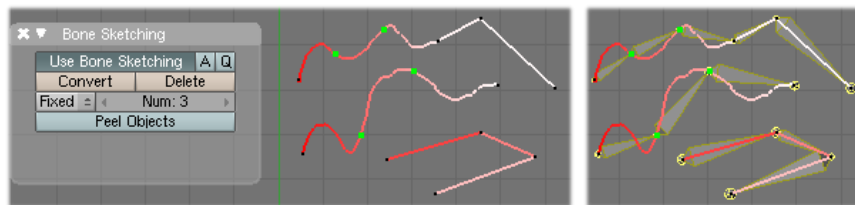
The Subdivision Method drop-down list of the Bone Sketching panel.

変換には4つの方法があります。3つの「シンプルな方式」と、1つの高度で複雑な方式、Templateです。Templateは、スケッチしているアーマチュアまたは別のアーマチュアのボーンを雛形として使います。これは[次のページ\(eng\)](#)で解説します。ストレートセグメントは、Template以外のどの変換方法を使った場合もちょうど一つのボーンになります。フリーセグメントは、緑の点の位置で区切られてボーンになります。

生成されたボーンのロール角度は、そのストロークを描いたときに決まります(Template以外)。ストロークを描いたときのビューのZ軸方向がボーンのZ軸方向になります。

Fixed

この変換方法では、個々のフリーセグメントがNum欄で指定した数に分割されます。

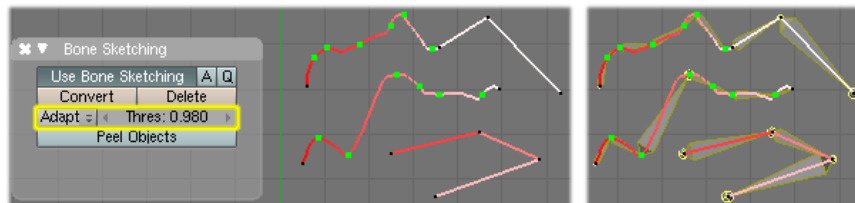


The Fixed conversion settings and its preview on selected strokes. The Fixed conversion result.

Fixed stroke-to-bone conversion example.

Adaptive

この変換方法では、ストロークの形状に十分に密接に沿うのに必要なボーンが生成されます。この「十分に密接」という程度をThresで決めます。数値を高くすると、より多くのボーンを使ってストロークに忠実に沿ったチェーンが作られます。ストロークが曲がりくねっているほどボーンが多くなります。



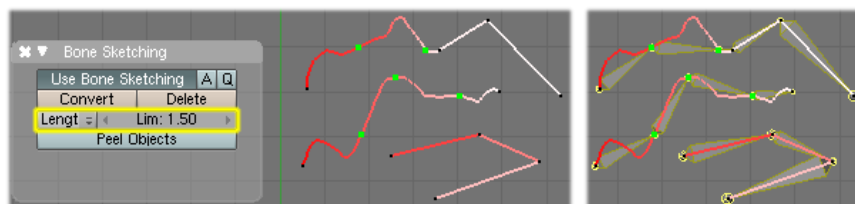
The Adaptive conversion settings and its preview on selected strokes.

The Adaptive conversion result.

Adaptive stroke-to-bone conversion example.

Length

この変換方法では、必要に応じてボーンが作られ、どのボーンもLenで指定した長さ(Blender単位)以上にはなりません。



The Length conversion settings and its preview on selected strokes.

The Length conversion result.

Length stroke-to-bone conversion example.

Template

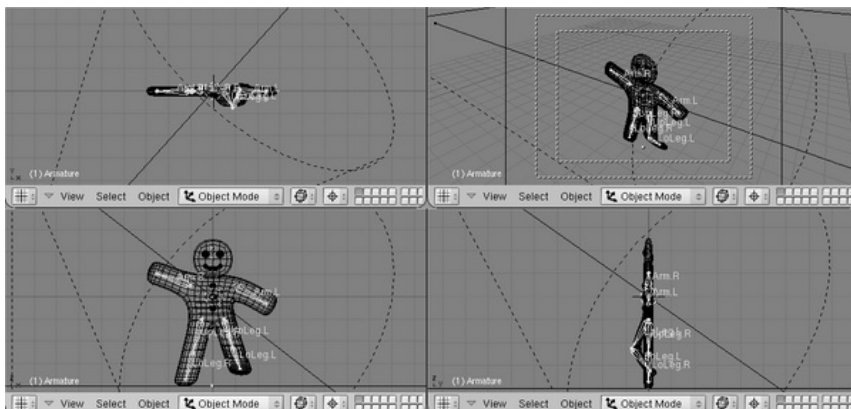
これはより複雑な話題なので、[専用のページ\(eng\)](#)で解説します。

Retarget

...

スキニング

[前のページ](#)では、アーマチュアをデザインしたり、ボーンチェーンを作成する方法等を見てきました。さて、良いリグを持つことは、最終的な目標ではありません – あなたが“死の舞踏”のアニメーションを作成しようとしているのでもなければ、スケルトンには肉をつけたくるはずですよ！ 驚いたことに、オブジェクト(複数可)にアーマチュアを“リンク”させて、変形および/または変換させることは、“スキニング”プロセスと呼ばれています...



The ginebread mesh skinned on its armature...

ブレNDERでは、主に2つのスキニングのタイプがあります:

- [Parent/オブジェクトをボーンに親子化/拘束](#)できます – その後、ポーズモードでボーンを変換するときに、それらの“子”オブジェクトも、標準的な親/子関係とちょうど同じように、変換されます... **この方法を使用するときに、“子たち”は変形されることはありません。**
- [メッシュ全体にアーマチュア修飾子を使用](#)できます、そしてこのアーマチュア内部のいくつかのボーンに、このオブジェクトの一部を対応させることも。これはより複雑で強力な方法であり、そして **実際にオブジェクトの形状を変形させる唯一の方法**です。すなわちその頂点/コントロールポイントの相対位置を変更します。

ポーシング

必要なオブジェクトのアーマチュアをいったん [スキン](#)化してしまえば、ポーズをつけ始めることができます。基本的には、そのボーンを変換することで、スキンのオブジェクトを変形したり、変換したりできます。しかしこれは [エディットモード](#)で行なってはけません - このモードで編集するのは、[アーマチュアのデフォルト](#)、基本となる、“rest” position だということに注意してください。オブジェクトモードも同じく使用することはできず、それはオブジェクト全体としての変換しかできないからです...

そういうわけで、Blenderのアーマチュアは、このプロセスに特化した、第三のモード、ポーズモードを持っています。これは“ボーンのためのオブジェクトのモード”のようなものです。レストポジション ([エディットモード](#)で編集された位置)では、それぞれのボーン的位置/回転/スケールはニュートラルの値になります (すなわち位置と回転は **0.0** で、スケールは **1.0**)。したがって、ポーズモードでボーンを編集するときは、その静止位置から差し引いて、トランスフォームプロパティを作成することになります - 実際これは [メッシュの相対的な形状キー](#)と非常によく似ています。

ポーズセクションの概要

このセクションで見ていくのは:

- ポーズモード特有の [可視化機能](#)。
- このモードで [ボーンを選択して編集する](#) 方法。
- [ポーズライブラリを使用する](#) 方法。
- ボーンの被写界深度(自由度)を調整する [制約の使用](#) 方法。
- [インバースキネマティクスの機能を使用する](#) 方法。

完全に静的な目的に使用する場合でも、ポーシングは [アニメーション機能とテクニク](#)に密接に関連しています。このパートでは、アニメーションからは独立したポーシングに焦点を当ててみますが、これは常に可能とは限りません。ですからBlenderのアニメーションについて何も知らないのであれば、前もって関連する章 ([上記のリンクに従って](#))を読み、それからここに戻ってくるというのも良い考えかもしれません...

関連項目

いつものように、さらなる実践例としては [tutorials](#) や、そして特に [this BSoD one](#)を参照してください。

Visualization

(視覚化)

どのモードでも使えるアーマチュアの視覚化オプションについては [こちらのページ](#) をご覧ください(視覚化の種類やボーンの形状など)。

ポーズ(Pose) モードではさらに、ボーンを見やすく分類するのを助ける [色とグループ\(colors and groups\)](#)、アーマチュアのアニメーションを見やすくする [ゴースト\(ghosts\)](#) および [モーションパス\(motion paths\)](#) の機能を使えます。

Colors

(色)

ポーズ(Pose) モードでは、次の二つの手順により、各ボーンは別々の色を持つことができます。Color ボタンから操作します(このボタンは 編集 (Editing) コンテキストでアーマチュア(Armature) パネル F9):

- 無効にすると、ボーンはその状態 (state) を元に着色されます (たとえば制約の有無、ポーズの有無など)。
- 有効にすると、ボーンはどのボーングループに所属するかによって着色されます (どのグループにも属さない場合は上のルールに従います)。

両方の着色方式を混ぜることもできます。 [この下](#) をご覧ください。

Coloring from Bone State

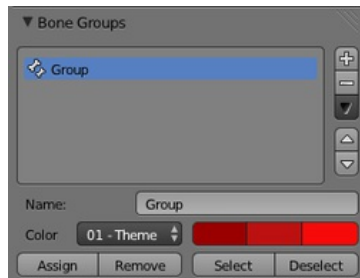
(ボーン状態による着色)

これはデフォルトで、もっとも古い方法です。優先度順に6つのカラーコードがあります。ボーンの色は、ボーンの状態が当てはまる一番上の色になります:

- 紫色: [Stride Root bone \(eng 2.4\)](#)。
- オレンジ色: 対象のない処理制約を持つボーン。
- 黄色: [IK処理制約 \(IK Solver constraint\)](#) を持つボーン。
- 緑色: その他の制約を持つボーン。
- 青色: ポーズのつけられたボーン(キープフレームを持つもの)。
- 灰色: デフォルトの状態。

Coloring from Bone Group

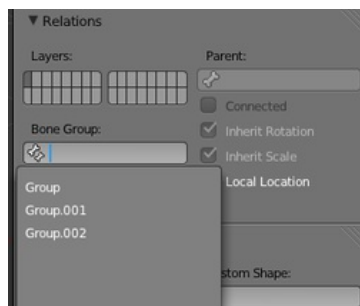
(ボーングループによる着色)



The Bone Groups panel with a bone group (default colors).

ボーングループパネルはアーマチュアの オブジェクトデータ (Object Data) エディタから利用でき、複数ボーンの色、テーマ分けをしやすくします。

新しいボーングループを作るには ボーングループ (Bone Groups) 欄にある [グループの追加](#) ボタンをクリックします。グループが作られると、上に並ぶコントロールを使うようになり、ドロップダウンリストから他のグループを選んだり(矢印のボタン)、現在のグループを改名したり(テキスト欄)、削除したりできます ('X'ボタン)。



ボーングループ (Bone Group) のドロップダウンリスト

指定ボーングループに選択中のボーンを割り当てるには、以下のいずれかを行います:

- ボーングループ (Bone Groups) 内で既存のグループをひとつ選び、割り当て/適用 (Assign) をクリックします。
- ボーン (Bones) パネルのリレーション欄 (Relations) にある ボーングループ (Bone Group) ドロップダウンリストから選びます。

3Dビューでは ポーズ(Pose) » ボーングループ(Bone Groups) メニュー項目や ボーングループ(Bone Groups) ポップアップメニュー (CtrlG) を使って以下のことができます:

新規グループに割り当て (Assign to New Group)

選択中ボーンを新規ボーングループに割り当てます

グループに割り当て (Assign to Group)

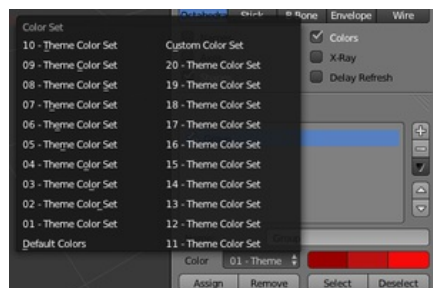
選択中ボーンを選択中のボーングループに割り当てます

ボーングループから選択中のものを除去 (Remove Selected from Bone Groups)

選択中のボーンをすべてのボーングループから取り除きます

ボーングループを除去 (Remove Bone Group)

アクティブなボーングループを取り除きます



ボーングループの ボーンカラーセット (Bone Color Set) 一覧と選択されたカラーテーマの色見本

グループに「カラーテーマ (color theme)」を割り当てることもできます (所属ボーンがすべてこの色になります)。なお、この色を見るには Armature の表示パネルにある 色 (Colors) が有効である必要があります。以下の選択には ボーンの色セット (Bone Color Set) ドロップダウンリストを使ってください:

- デフォルト色 (Default Colors) - デフォルト色 (灰色)
- テーマ色セット (Theme Color Set) - 20 ある Blender の既定セットのうちの一つ (二桁の番号で始まる名前です)。全グループで共有されています。
- カスタム色セット (Custom Set) - 独自定義の色セット。グループごとに固有です

この一覧の下に、3つの色見本とボタンがあります。

- 最初の色見本は、非選択のボーンの色です。
- 二番目の色見本は選択中のボーンの縁取りの色です。
- 三番目の色見本はアクティブなボーンの縁取りの色です。

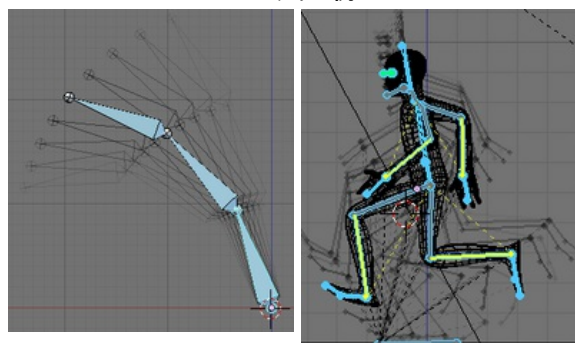
色見本をクリックすると、標準的な色編集ダイアログが開いて色の変更を行えます。この操作をするたびに、Custom Set オプションが自動的にオンになります。

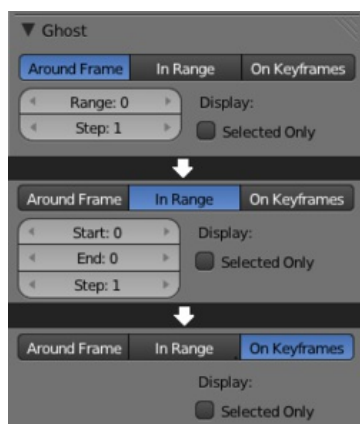
Ghosts

(ゴースト)

伝統的なアニメーション制作に少し詳しくれば、デッサン職人が作業中のフレームの直前数フレームを見るのに、トレーシングペーパーを多用していることをご存知かもしれません。こうすることでキャラクターの動き全体を、再生し直すことなしに可視化できます。Blender のゴースト (ghosts) はこれにとってもよく似ています。ゴーストはアーマチュア用で、ポーズモードで使います。

ゴーストの例





ゴースト(Ghost) パネルにはモードに応じたオプションが表示されます。

ゴーストは、あるフレームにおけるボーンの輪郭を、単純な黒い線で(目立たないように)描いたものです。

ゴーストの設定パネルはプロパティ(Properties)のオブジェクトデータ(Object Data)にあります。ゴーストは3種類あり、オプションの一部は同じもの değildir:

現在のフレーム周辺 (Around Current Frame)

現在のフレームの前後に、指定数分のゴーストを表示します。ゴーストは不透明に描かれる現在のフレームから次第に透明度を増して、もっとも離れたフレームで透明に描かれます。3つのオプションがあります:

範囲 (Range)

「両側」にゴーストをいくつ出すか指定します。たとえば値 **5** はゴーストを10(現在のフレームの前に5つ、後ろに5つ)出します。

ステップ (Step)

ゴーストを毎フレームごとに出すか(デフォルトの値 **1**)、2フレームごとに出すか、3フレームごとに出すか、といった指定をします。

選択したもののみ (Selected Only)

有効にすると、選択中のボーンのゴーストだけが見えます(無効にするとアーマチュア内のすべてのボーンがゴーストを持ちます)

指定範囲内 (In Range)

このアーマチュアの、指定フレーム範囲にあるボーンのゴーストを表示します。ゴーストは開始フレームでは透明に描かれ、次第に不透明度を増して最後のフレームでは不透明に描かれます。4つのオプションがあります:

開始 (Start)

範囲の開始フレームを数値で指定します(指定フレームそのものは含まない)。なお、残念ながら null やマイナスの値は入力できません。したがってフレーム2以降(2を含む)のゴーストしか見えないことになります。

終了 (End)

範囲の終了フレームを数値で指定します。開始(Start)より小さな値にはできません。

ステップ (Step)

上の説明と同じです。

キーフレーム上 (On Keyframes)

指定範囲内 (In Range) オプションによく似ていますが、アーマチュアのアニメーション内のキーフレーム(一つ以上のボーンにキーがつけられたフレーム)だけにゴーストが現れます。したがってオプションに **ステップ (Step)** はありません。

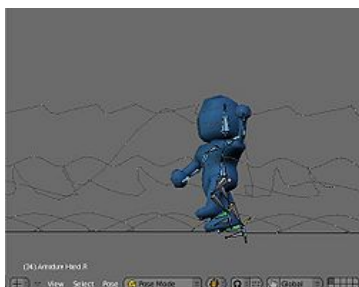
奇妙なことに、ゴーストの陰影は **指定範囲内 (In Range)** に比べて反転したのになります。最初のキーフレームは不透明で、次第に透明度を増して最後のキーフレームは透明です。

最後に、ゴーストはアニメーションの再生中 (AltA) にも有効です。役立つのはもちろん **現在のフレーム周辺 (Around Current Frame)** オプションを使う場合だけです。

注意点として、この表示の有効無効を全体的に切り替えるスイッチはありません。そのためには **ゴースト (Ghost)** を **0** にするか (現在のフレーム周辺 (Around Current Frame) の場合)、**開始 (Start)** と **終了 (End)** に同じフレーム番号を入力します (その他のゴーストの場合)。

Motion Paths

(モーションパス)



モーションパスの例

この機能により、ボーンの両端(デフォルトである先端(tip)、またはルート(root))のパスを曲線として視覚化できます。

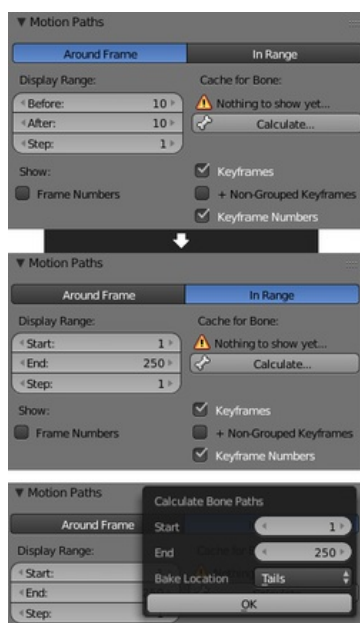
オプションを見る前に、まずパスの表示／非表示の方法を見てみましょう。[ゴースト\(ghosts\)](#)と違って、手動で行う必要があります。また、はじめにモーションパスを表示／非表示にしたいボーンを選択する必要があります。次に、

- パスを表示するには(もしくはパスの更新が必要なら)パスの計算(Calculate Path) ボタンをクリックするか、3D ビューのメニューで ポーズ(Pose) » モーションパス(Motion Paths) » パスの計算(Calculate Paths) を選びます。
- パスを非表示にするには、その横の×印をクリックします。

忘れないでください:これらの操作が影響するのは、選択中のボーンのパスだけです。

パスは非選択のボーンでは明るい灰色、選択中のボーンではやや青い灰色で描かれます。各フレームはパス上に小さな白い点として現れます。

ゴースト同様に、パスはポーズ/キーフレームを編集した際に自動更新されます。また、アニメーション再生中(AltA)にも有効です。現在のフレーム周辺(Around Current Frame) オプションを使う場合には便利です。



The モーションパス パネルにはモードに応じたオプションが表示されます。

それではパスのオプションを見てみましょう:

フレーム周辺 (Around Frame)

現在のフレームの前後、指定数分のフレームのポーズについて、パスを表示します(ゴーストと同様です)。

指定範囲 (In Range)

特定範囲内のポーズについて、パスを表示します。

表示範囲 (Display Range)

前/後 (Before/After)

現在のフレームの前後何フレームを表示させるのか指定します(「フレーム周辺 (Around Frame)」用)

{{Literal|開始/終了 (Start/End)}}

表示/計算させるパス範囲の、開始および終了フレーム(「指定範囲 (In Range)」用)

ステップ (Step)

ゴーストのものと同種のもので、 n フレームおきにパスを表示します。もっとも役立つのは、フレーム番号を表示しているとき(後述)です。3Dビューが雑然とするのを防ぎます。

フレーム番号 (Frame Numbers)

有効化すると、パス上にある各フレーム点の隣に、小さな数字が現れます。これはもちろん、そのフレームの番号です。

キーフレーム (Keyframes)

有効化すると、モーションパス上にボーンのキーフレームを表す、大きな黄色の四角が描かれます(これは、指定フレームの、キーづけされたボーンのパスだけが黄色い四角になるということです)。

キーフレーム番号 (Keyframe Numbers)

有効化すると、フレーム番号のうちキーフレームであるものに色がつきます。上の項目「キーフレーム」を有効にしないと有効になりません。

+ グループにないキーフレーム (Non-Grouped Keyframes)

ボーンのモーションパス用に、名前的一致したグループ内だけでなくキーフレームのすべてのアクションを探します(遅くなります)

計算 (Calculate)

開始 (Start) / 終了 (End)

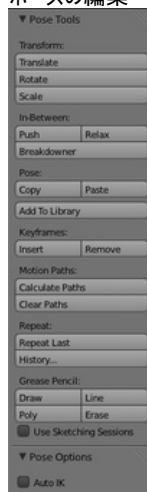
開始/終了はフレームの範囲を定義し、この範囲内でモーションパスが描かれます。この設定を変更したときは、3Dビューのパスを更新するために *再度パスの計算 (Calculate Paths)* をする必要があります。

ゴーストと違って、開始フレームが *範囲に含まれる* ことに注意してください。すなわち開始を 1 にすれば、実際にフレーム 1 がパスの開始点になります。

位置の焼き込み (Bake Location)

デフォルトでは、先端 (tip) のパスを取得します。この設定を終端 (tail) に変えると、ボーンのルートのパスを取得します (なお、Blender の UI ではボーンのルートは「head」と呼ばれます)。この設定の変更後、3D ビューのパスを更新するには、*パスの計算 (Calculate Paths)* を再度行う必要があります。

ポーズの編集



ポーズのツール群

ポーズモードでは、ボーンがオブジェクトのように振る舞います。したがってトランスフォーム操作(移動/回転/拡張など)はオブジェクトモードのものによく似ています(利用可能な操作はすべて、ポーズ(Pose) » トランスフォーム(Transform) にまとめられています)。ただし、重要な違いがあります:

- **後述するように**、ボーン間の関係 (relationship) は非常に重要です。
- 指定ボーンの「トランスフォームの中心(選択が一つだけなら、デフォルトのピボットポイントの位置)」は *ボーンのルート* です。ところで、ピボットポイントのオプションにはいくつか、正しく動作していないものがあるようです。実際のところ、3D カーソルを除いて、他はすべて選択対象全体の重心 (median point) を使うようです(たとえば、アクティブオブジェクトが選択状態のときにアクティブボーンのルートを使いません)。

ボーンを選択

ポーズモードでの選択操作は [編集モード](#) のものによく似ていますが、いくつか特殊な点があります:

- ポーズモードではルート(root/基部)やティップ(tip/先端)ではなく *ボーン全体* だけを選択できます

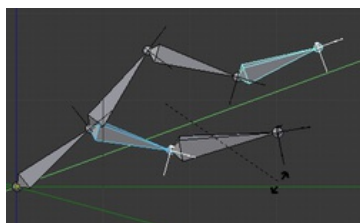


関係で選択 (Select Grouped) ポップアップメニュー

- 関係で選択 (Select Grouped) ポップアップメニュー (⇧ ShiftG) を使えば、所属するグループやレイヤーをもとに、ボーンを選択できます:
 - 選択中のボーンと同じグループに所属する全ボーンを選択するには、グループ (In Same Group) (⇧ ShiftGNum1) を使ってください。
 - 選択中のボーンと同じレイヤーに所属する全ボーンを選択するには、レイヤー (In Same Layer) (⇧ ShiftGNum2) を使ってください。

ポーシングの基礎

以前述べたように、ボーンのトランスフォームはアーマチュアの編集モードで定義された *休止位置 (rest position)* を元に行われます。すなわちポーズモードの休止位置では、各ボーンは **1.0** の拡張を持ち、回転および位置を持ちません(このことは 3Dビューで N を押すと表示される トランスフォーム (Transform) プロパティのパネルで確認できます)。



2つのボーンを選択中にローカル Y軸を固定して回転する例。座標軸を示す2本の緑の線はアーマチュアの中央にあり、各ボーンのルート上にはないことに注意してください

また、こうした操作の行われるローカル空間はボーン自身のものになります(アーマチュアパネルで座標軸 (Axes) オプションを有効にすると見ることができます)。これは特に、座標軸を固定する際に重要です。たとえば、ポーズモードに「ボーンを軸周りに回転 (roll) する」特別なツールはありませんが、回転をローカル Y軸上に固定すると (RYY)、ボーンをその中心軸周りに回転させることができます。複数ボーンを選択した場合も、回転が各ボーン自身のローカル軸に固定されます。

ポーズをつける際に、アーマチュアにはひとつ以上のオブジェクトがスキニングしてあるでしょう。ポーズモードでボーンをトランスフォームしたとき、それに応じて関係するオブジェクトやオブジェクトの形状がリアルタイムで移動/変形します。残念ながら、複雑なリグ構成や重いスキニングオブジェクトがあると、この動作でラグが発生し、インタラクティブな編集がしづらくなることがあります。この問題が起きたときは、アーマチュアパネルにある遅れて更新(Delay Deform) ボタンを有効にしてください。スキニングオブジェクトは、トランスフォーム操作を確定した際にだけ更新されるようになります。

自動IK (Auto IK)

ツールシェルフにある auto IK オプションは、ボーンのポーシング時に、一時的な IK 制約を有効にします。連鎖(chain)は選択されたボーンのティップ(tip/先端)から、最上位の親ボーンのルートまで作用します。なお、このモードにはオプション設定がありません。また、動作するのは連鎖内のボーンにトランスフォーム結果を適用した場合のみです。

休止ポーズ (Rest Pose)

いったんトランスフォームしたボーンを休止位置に戻りたいときは、単にボーンのトランスフォーメーションを消去してください(いちどにすべてを消去するには AltG/AltR/AltS ショートカットか Pose » Clear Transform » トランスフォームの消去 (Clear User Transform)、WNum5 を使います。コマンドは Pose » Clear Transform のサブメニューからも利用できます)。

なお、編集モードでエンベロープ(Envelope)表示にしていると、AltS は拡張の消去ではなく、選択ボーンの影響範囲を拡張する点に注意してください(アーマチュア » トランスフォーム » エンベロープ距離の拡大縮小 (Scale Envelope Distance) メニューからも実行できます)。

逆に、現在のポーズを新たな休止位置として定義することができます(言い換えれば、現在のトランスフォーメーションを編集モードに「適用」します)。これには Pose » デフォルトのポーズに適用 (Apply Pose as Restpose) メニューを使うか、CtrlA のポップアップダイアログを承認します。同時に、スキニングされたオブジェクトやジオメトリも、デフォルトの変形のない状態に戻され、通常は再度スキニングを行う必要があります。

編集モードではアーマチュアは常に休止位置で表示されますが、オブジェクトおよびポーズモードでは、デフォルトでは *ポーズ位置 (pose position)* で(すなわちポーズモードでトランスフォームされた通りに)表示されます。どのモードでも休止位置で表示したいときは、アーマチュアパネルにある、休止位置 (Rest Position) ボタンを有効にします。

中間ポーズ (In-Betweens)

アニメーションのポーズ編集用ツールが複数あります。

ポーズをリラックス (Relax Pose) (Pose » In-Betweens » Relax Pose または AltE)

この機能は前述の話に少し関係がありますが、キーフレームのつけられたボーンでないと役立ちません([アニメーションの章](#)をご覧ください)。ボーンを編集するときこのコマンドを使うと、「キーをつけた位置」の効果が減り、ボーンが少しずつ「キーをつけた位置」に戻ります。近づけば近づくほど変化幅が小さくなります。

ポーズを誇張 (Push Pose) (Pose » In-Betweens » Push Pose または CtrlE)

現在のポーズを大きにします。

ポーズの中割 (Breakdowner) (Pose » In-Betweens » Pose Breakdowner または ⇧ ShiftE)

現在のフレームにふさわしい中割ポーズを作ります

ポーズモードにも、アーマチュア専用の編集オプションやツール群があります。たとえば [自動ボーン名 \(auto-bones naming\)](#)、[プロパティの切り替え/有効化/無効化](#) などです。詳しくはリンク先のページをご覧ください。

ポーズのコピーとペースト

モード: ポーズモード

パネル: 3D ビュー ヘッダー

メニュー: ポーズ (Pose) » 現在のポーズをコピー (Copy Current Pose), ポーズ (Pose) » ポーズをペースト (Paste Pose), ポーズ (Pose) » X軸で反転させたポーズをペースト (Paste Flipped Pose)



ポーズ
モードの
3D View
ヘッダーに
ある、ポ
ーズのコ
ピー
ペーストの
ボタン

Blender でポーズのコピーとペーストを行うには、ポーズ (Pose) メニューを使うか、3D ビューのヘッダーの右側にある3つのコピーペーストボタンを直接使います:

現在のポーズをコピー (Copy Current Pose)

選択中のボーンの現在のポーズをポーズバッファにコピーします。

ポーズをペースト (Paste Pose)

バッファに入れたポーズを現在ポーシング中のアーマチュアにペーストします。

X軸で反転させたポーズをペースト (Paste Flipped Pose)

バッファに入れたポーズの X 軸に対して左右反転させた (X axis mirrored) ポーズを、現在ポーシング中のアーマチュアにペーストします。

以下の点は重要です:

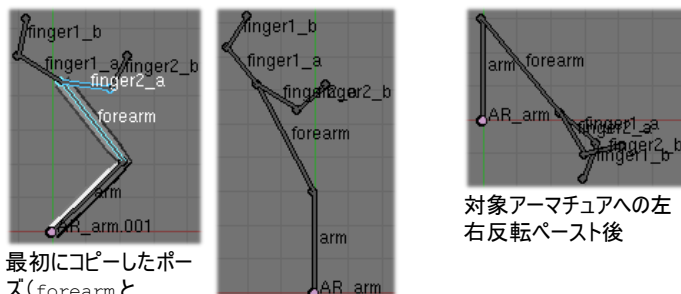
- このツールは Blender のセッションレベルで動作します。別々のアーマチュアやシーン、ファイルをまたいで使えるということです。ただ、ポーズバッファは保存されないで、Blender を閉じると失われます。
- ポーズバッファはひとつだけ存在します。
- コピー操作の対象は選択中のボーンだけです(すなわち、選択中のボーンのポーズのみがコピーできます)。
- 一方、ペースト操作には、ボーンを選択範囲は意味を持ちません。コピーされたポーズは「名前別 (per-name) ルール」に従って適用されます(ポーズをコピーした際に“forearm”ボーンが選択されていたのであれば、ペースト時には、現在ポーシング中のアーマチュアの“forearm”ボーンがそのポーズになります。該当する名前のボーンがなければ、何も起きません...)
- 実際にコピーペーストされるのは、各ボーン、それ自身の空間内での位置/回転/拡縮です。このため次のような条件によっては、コピーした元のポーズと、ペーストしたポーズが異なるかもしれません:
 - ボーンの休止位置
 - ボーンの親の現在のポーズ

コピーペーストの例



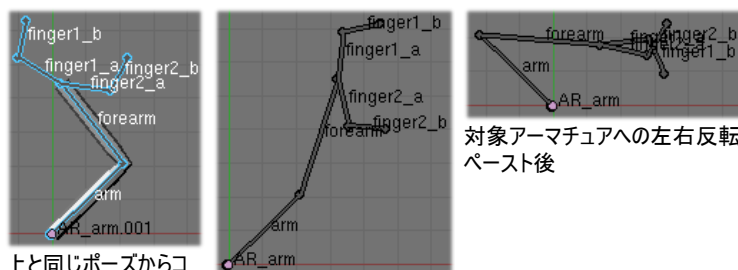
ペースト対象となるアーマチュアの休止位置

元になるアーマチュアの休止位置



最初にコピーしたポーズ(forearmとfinger2_a)だけが選択されており、この二つだけがコピーされます)

対象アーマチュアへの左右反転ペースト後



上と同じポーズからコピーしますが、今度は対象アーマチュアへのペースト後

対象アーマチュアへの左右反転ペースト後

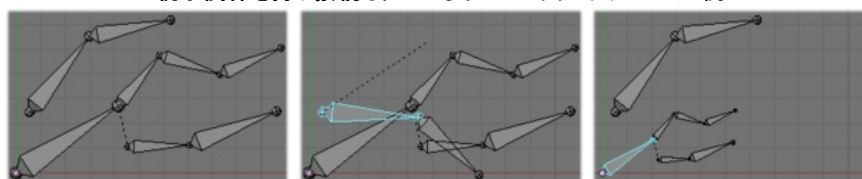
ボーンの関係性の影響

ポーズモードではボーン間の関係が非常に重要です。トランスフォームの挙動に大きな影響があります。

デフォルトでは、子ボーンは以下を継承します:

- 親の位置および、当然ながら子自身のオフセット
- 親の回転(すなわち子は親に対して、一定の回転を維持します)
- 親の拡縮、およびここでも子自身のオフセット

親子関係を持ち接続されているボーンのトランスフォームの例

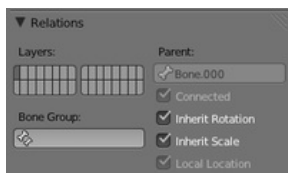


休止位置でのアーマチュア

ルートボーンの回転

ルートボーンの縮小

標準的な子オブジェクトとほぼ同じです。この骨別(per-bone)ルールの挙動は、アーマチュアのボーン(Armature Bones) パネルにあるサブパネルを使って変更できます:



ポーズモードのアーマチュアのボーン(Armature Bones) パネル

回転を継承 (Inherit Rotation)

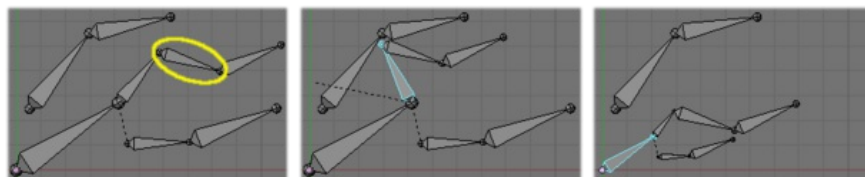
無効にすると、ボーンの親との回転の関係性を壊します。すなわち、親が回転したとき、子はアーマチュアのオブジェクト空間内で、回転を保持します。

拡大縮小を継承 (Inherit Scale)

無効にすると、ボーンの親との拡大縮小の関係性を壊します。

こうした継承の挙動はボーン階層に沿って伝播します。このためあるボーンを縮小したとき、デフォルトではそれに合わせてそのすべての子孫が縮小します。しかしながら同じ「家族」内のいずれかのボーンの拡大縮小の継承 (Inherit Scale) か 回転の継承 (Inherit Rotation) プロパティを無効にすれば、拡大縮小の伝播は途切れます。すなわち、このボーンの子孫のいずれかを拡大縮小したとき、このボーン およびそのすべての子孫はもうその影響を受けません。

回転の継承を無効にした状態での、親子関係を持ち接続されているボーンのトランスフォーム例

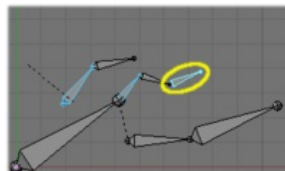


黄色の線で囲まれた、アーマチュアの子孫内の回転の継承が無効化されたボーン
 子孫内の回転の継承が無効化されたボーン
 子孫内の回転の継承が無効化されたボーンの縮小

接続されているボーンは、「移動できない」という特性を持ちます。ルートが親のティップ (tip/先端) 位置になければならないので、親を動かさない限り、子のティップを動かしても (これは子の回転操作になります)、子のルートを動かすことはできません。接続されているボーンを選択中に G を押せば、Blender は自動的に回転操作に切り替えます。

ボーン間の関係は、複数ボーンが選択されトランスフォームされたときの挙動に大きく影響します。状況はさまざまであるため、ここですべての可能性を挙げられるかわかりませんが、いずれにしてもあなたの問題解決の糸口になるでしょう:

- 選択された、関係を持たないボーンは、いつものように独立してトランスフォームされます。



ボーンの縮小。一部が関係を持っています

- 同じ「家族」の複数のボーンが選ばれていると、「一番上の親」だけが実際にトランスフォームされます - 子孫は、選択されていないかのように、親子関係の手順を通じてのみ操作されます (図「ボーンの縮小。一部が関係を持っています」をご覧ください - 黄色の線で囲まれた3番目のティップボーンは、選択状態でかつアクティブであるにもかかわらず、非選択のものと同じように、親子関係を通じてのみ縮小しました。そうでなければ倍小さくなっていました)。
- 接続されたボーン (一番上の親を除く) とされていないボーンを選択して移動操作を始めると、接続されていないボーンだけが影響を受けます。
- 選択した中に子の接続されたヒンジボーンがあり、選択された中で「最年長の親」が接続されているとき、G を押しても何も起きません。Blender は 移動/つかむ操作のまま、もちろん接続されているボーンには何も効果を持ちません。実際のところこれは意図的なものに思えないため、バグかもしれません。

したがって、連鎖するボーンにポーシングをすると、常にそのルートボーンからティップボーンまでの要素が変更されます。この手順は **順運動学/forward kinematics/FK** と呼ばれています。この後のページでは、**逆運動学/inverse kinematics/IK** と呼ばれる、Blender の持つもう一つのポーズ方式を見ていきます。この方式では、連鎖するボーンのティップボーンを動かすだけで、全体のポーズをつけることができます。

なお、[ポーズライブラリ](#) ツールはこの機能をやや拡張・補完します。

Pose Library

(ポーズライブラリ)

Intro

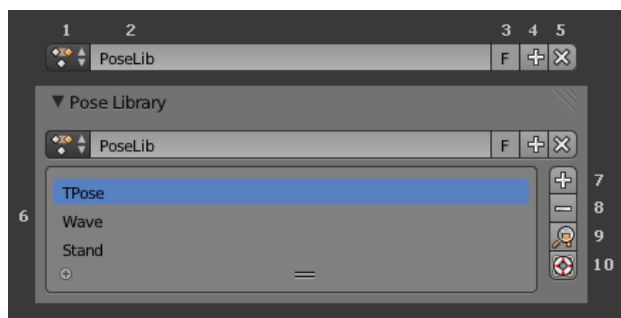
(紹介)

ポーズライブラリ(Pose Library) パネルはアーマチュアのポーズを保存、適用、管理するのに使われます。

ポーズライブラリは アクション(Actions) に保存されます。通常はポーズライブラリはアクションとして使われませんが、変換は相互に行えます。

Pose Library Panel

(ポーズライブラリパネル)



プロパティ > アーマチュア > ポーズライブラリ(Properties > Armature > Pose Library)

1. 接続する アクション / ポーズライブラリを選びます。
2. ポーズライブラリの名前。
3. Fake User の指定
ユーザーを持たない ポーズライブラリを保存します。
4. アクティブなオブジェクトに新たな ポーズライブラリを(ポーズモードのときに)追加します。
5. アクティブなオブジェクトから ポーズライブラリを取り除きます。
6. アクティブな ポーズライブラリの ポーズの一覧。
7. ポーズを追加します。
新規追加

アーマチュアの現在のポーズを使って、新たな ポーズをアクティブな ポーズライブラリに追加します。

新規追加(現在のフレーム)

新規追加と 既存分の置き換えは、ポーズを自動的にアクションのフレームに割り当てます。

新規追加(現在のフレーム)は、タイムカーソル(Time Cursor)の指す現在のフレームをもとに、ポーズを ポーズライブラリに追加します。
あまりサポートされていない機能です。

既存分の置き換え(Replace Existing)

アクティブな ポーズライブラリにある既存の ポーズを、アーマチュアの現在のポーズで置き換えます。

8. アクティブな ポーズを ポーズライブラリから取り除きます。
9. アクティブな ポーズを選択中の ポーズボーン(Pose Bones) に適用します。
10. アクションの最適化(Sanitize)。アクションを ポーズライブラリとして使うのに適した状態にします。
アクションを ポーズライブラリに変換するために使います。
キーフレームを持つ各フレームについて、ポーズが ポーズライブラリに追加されます。

Editing

(編集)

3D ビュー、ポーズモード

ポーズの閲覧 CtrlL

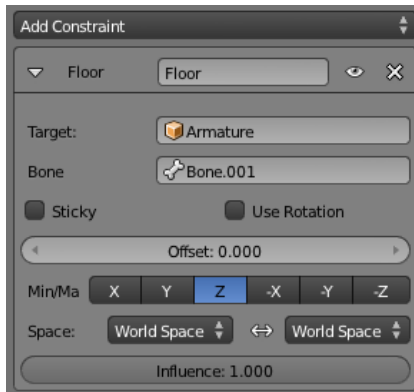
ポーズの追加 ⇧ ShiftL

ポーズの改名 ⇧ ShiftCtrlL

ポーズの除去 AltL

Applying Constraints to Bones

(ボーン制約)



Pose モードでの Constraints パネル。アクティブなボーン (Bone.001) に Floor constraint が使われています。

Pose モードでは ボーンはオブジェクトのように振る舞うので、制約 (Constraint) を課すこともできます。このモードで Constraints パネルが Object と Editing のどちらのコンテキストでも表示されるのは、このためです。パネルには *アクティブなボーン* の制約が表示されます (パネルの最上部にはボーンの名前が表示されます)。

例えば Limit 制約をボーン制約にして、ポーズ変形の自由度を決めることができます。他のオブジェクト/ボーン (同じオブジェクト内のものでも、別のアーマチュアにあるものでも) の動きを追わせる制約も作れます。また、[インバースキネマティクス機能](#) は主に、ボーン専用の IK Solver (IK) 制約を通じて利用できます。

例えば、人間のひじは裏側や横に曲げることができず (骨折していない限り)、前側への曲げやねじりも、一定の範囲に限られています (例: ひじの安静位によって 0° から 160° 、あるいは -45° から 135° になるかもしれません)。そこで Limit Rotation (回転制限) 制約を前腕のボーンに使用 (前腕のボーンの根を中心とした回転の結果、ひじが動くためです)。

制約を持ったボーン、もしくは制約の目標としてのボーンの使い方が [Constraint](#) のページに詳しく書かれています。

Inverse Kinematics

IKは、アニメーションのプロセスをシンプルにする方法で、より高度なアニメーションを少ない労力で作成することを可能にします。

Automatic IK

自動-IKは、簡単なデモポーズを作るためのツールです。3Dビューのポーズモードのツールシェルフの下にあります。即席の荒いデモンストレーションにのみ使用すべきで、拘束の使用はできません。実際の効果は、各ボーンのチェーンの両端に鎖長0(無限大)のIK制約を与えた場合と同じです。

IK拘束

...

スプラインIK

スプラインIKはカーブに沿ってボーンの鎖を整列する拘束です。曲線と予測可能性およびボーンによる統合制御によって、美意識にかなう形状を達成するための、使いやすさと柔軟性を倍増させ、スプラインIKはリグ職人の工具箱の貴重なツールとなります。それは特に、尾、触手、そして棘のような、柔軟な体の部分のリギングだけでなく、ロープのような無機的なものにも適しています。

スプラインIKの設定の完全な説明は、[Spline IK](#) のページに詳しく説明されています。

基本設定

スプラインIK拘束は、厳密には 'インバースキネマティクス' メソッド (つまり、IK拘束)ではなく、さらに 'フォワードキネマティクス' メソッド (つまり、通常のボーンポーシング)でもありません。しかしそれでも、IK Constraintのいくつかの特性を共有しており、例えばオブジェクトと違って多数のボーンの使用可能であることや、他のすべての制約が評価された後に評価されることです。標準的なIKチェーンと Spline IK チェーンが両方同時に骨に影響を与える場合、標準のIKチェーンが優先されることに注意する必要があります。結果の制御が困難なので、このような設定はできれば回避すべきです。

スプラインIKのセットアップに必要なものは、接続されているBoneのチェーンと、これらのボーンを拘束するためのカーブです。

1. 選択されたチェーンの最後のボーンで、プロパティエディタ内のボーンの拘束タブから '[Spline IK](#)' を追加します。
2. 'Chain Length' の設定を、チェーン内のボーンの数で設定します (選択したボーンから開始し、含みます)。これは曲線によって影響されます。
3. 最後に、曲線を制御する 'Target' フィールドを設定します。

おめでとう、ボーンチェーンは曲線によって制御されています。

設定とコントロール

ロールの制御

スプラインIKチェーンの 'ねじれ' や 'ひねり' の制御にも、チェーン内のボーンをy軸に沿って回転させる標準的な方法が適用できます。例えば、その点から順方向へのチェーンのひねりを調整するには、単にチェーン内のボーンをy軸周りに回転させます。回転制約のコピーをボーンに適用すると動作するはずですが。

オフセット制御

ボーンチェーン全体が曲線の形状に沿うように構成でき、'Chain Offset' オプションが有効になっているときなら、3D空間内の任意の点に配置できます。デフォルトでは、このオプションは有効になっておらず、ボーンは変形されていない位置でのカーブに従うように構成されます。

厚さのコントロール

チェーン内のボーンの厚さの制御には、拘束の 'XZ Scale Mode' の設定を使用します。この設定は、チェーン内の各ボーンのX軸とZ軸のスケーリングの限定にどの方法を使用するかを決定します。

利用可能なモードは次のとおりです:

- None - このオプションは、XとZのスケーリング係数を 1.0 として保持します
- Volume Preserve - XとZのスケーリング係数は、Yのスケーリングファクタ (ボーンの長さ)の逆数とみなされ、ボーンの '体積' を維持します
- Bone Original - このオプションは、単にXとZのスケールファクタを使用するもので、ボーンは標準的な方法で評価された後のものである必要があります

これらのモードに加えて、'カーブの半径を使用'というオプションもありあります。このオプションが有効になっている場合、それぞれのボーンのエンドポイントが配置されている曲線上の点の半径からなる平均半径を、XとZのスケーリング係数を導出するために使用します。このスケーリング効果は、上記のモードを使用して決定され、必要に応じて芸術的なコントロールのための微調整を行うことができます。

良好にセットアップするためのヒント

- 最適な変形のために推奨することとして、ボーンはざっくり同じ長さにし、あまりにも長くはせず、曲線へよりフィットしやすくしておきましょう。また、ボーンは、理想的には 'rest pose' の形状での曲線に沿う方法で作成するべきで、伸縮が無効になっているときに特に顕著になる、曲線が極端に折れる領域の問題を最小限に抑えることにもなります。
- 曲線の制御には、フック (2.5で新しく追加された、特別な、ボーンフック) による曲線の制御頂点の制御の使用をお勧めします。制御頂点ごとにひとつフックを持ちます。一般的に、曲線のコントロールに必要とされるは制御頂点はほんの少しです (すなわち、ボーン3から5個につき制御頂点1つで十分です)。
- 使用される曲線の種類が問題になることは実際にはあまり無く、パスの押し出しもフォローパスの制約の使用によって行えます。これは実際には、フックが必要とするコントロールのレベルによって異なります。

- リグをセットアップするとき、一時的に制御用ボーン持つことが必要になり (曲線を制御するため) 変形に使用されているメッシュ (すなわちスプラインIKチェーンを含む変形用のリグ)のものとは別のアーマチュアに分けることも必要になります。これは疑似 "依存サイクル"を作成しないようにするため、ゆえにBlenderの依存関係グラフにできるのは、制御用ボーン、曲線、およびスプラインIK'edボーンなどのオブジェクトの依存関係をオブジェクトベースで解決することだけです。

スプラインIK

スプラインIKはカーブに沿ってボーンを整列するコンストレイントです。カーブによって美しい形状が簡単かつ柔軟に作成でき、またボーンによって扱いやすく一貫した操作性が得られるため、スプラインIKはリギング作業にとって計り知れない価値を持っているツールです。このツールは柔軟に曲がるもの、たとえばしっぽ、触手、脊髄、ロープのようなものに特に有用です。

設定についての全般的な解説は[スプラインIKコンストレイントのページ\(eng\)](#)にあります。

基本的なセットアップ

スプラインIKは厳密には'Inverse (逆方向) Kinematics' (IKコンストレイントなど)ではなく、'Forward (順方向) Kinematics' (通常のポーズ作成時など)です。しかし、いくつかの特徴はIKコンストレイントに通じるところがあります。通常のオブジェクトの場合とは異なる、複数のボーンの処理ができること、また他のコンストレイントの効果のあとで評価されることなどです。通常のIKチェーンとスプラインIKチェーンが同じボーンに同時に影響している場合、通常のIKチェーンが優先されます。しかしながら、そういう状況はコントロールしづらいので、避けたほうが良いでしょう。

スプラインIKを使用するには、一続きになったボーンのチェーンと、それを沿わせるカーブが必要です。

1. チェーンの中で最後のボーンを選択し、プロパティウィンドウのBone Constraintsタブで、[スプラインIK\(eng\)](#)コンストレイントを追加します。
2. 影響を受けるべきボーンの数(選択しているボーン自身を含む)をChain Lengthに指定します。
3. 最後に、Target欄でチェーンをコントロールするカーブを指定します。

これでチェーンはカーブを使ってコントロールできるようになりました。

設定とコントロール

Roll Control

スプラインIKチェーンの「ねじれ」または「ロール」をコントロールするには、通常の場合にボーンを回転させるための方法が使えます。たとえば、単純にボーンをそのY軸を中心に回転させて、そのボーン以降のチェーンのロールを調整したり、copy rotationコンストレイントを適用したりできます。

Offset Controls

Chain Offsetオプションを有効にすると、ボーンチェーンは自由な点に位置した状態で、形状のみカーブに従うようになります。デフォルトでは無効になっていて、チェーンはカーブの無変形時の位置に従います。

Thickness Controls

ボーンの太さはXZ Scale Modeで指定された方法でコントロールされます。この項目はチェーンに属するボーンのXおよびZ方向のスケールリングの仕方を決定します。

使用できるモードは、

- None - XおよびZ方向のスケールは1.0に保たれます。
- Volume Preserve - XおよびZ方向のスケールはY方向のスケール(長さ)の逆数となり、ボーンの体積が保たれます。
- Bone Original - 通常通りに評価されたXおよびZ方向のスケールを使います。

これらに加えて、Use Curve Radiusオプションがあります。これを有効にすると、それぞれのボーンの端の位置でのカーブの曲率半径の平均がXおよびZ方向のスケールに使われます。これにより、上に挙げたモードに対してアーティスト的な制御の必要性に応じて微調整ができます。

セッティングのヒント

- 適切な変形をするためには、それぞれのボーンはだいたい同じ長さで、また長すぎないようにして、よりきれいにカーブにフィットさせたほうが良いでしょう。また、ボーンは最初からなるべくカーブに沿った形で作っておいたほうが良いです。カーブが鋭く曲がる領域での問題を最小限にするためです。そういった問題はとくに伸縮オプションが無効になっているときに起こります。
- カーブをコントロールするには、フック(具体的にはBlender2.5で登場したボーンフック)をカーブの制御点それぞれに適用してコントロールすると良いでしょう。基本的に、カーブにはそれほど多くの制御点は必要ないはずです(たとえば3から5個のボーンに対してカーブの制御点が一つ、など)。
- Follow Pathコンストレイントと同様に、パスが抽出できるかぎりカーブの種類は関係ありません。カーブの種類が関係するのは、フックを使う際のコントロールの度合いです。
- リギングする際、現在のところはカーブをコントロールするためのボーンが別個のアーマチュアとして必要です。これは偽の「依存関係の循環」が発生するのを避けるためです。BlenderのDependency Graphでは、コントロールボーン、カーブ、およびスプラインIKされたボーンの依存関係をオブジェクト単位で決定することしかできません。

拘束

概要

拘束は、オブジェクトの機能、つまりオブジェクトの *transform properties* (位置、回転、スケール)を、様々なやり方でコントロールするものです。実際、拘束はある意味オブジェクトにおいて [モディファイア](#)と対をなすもので、オブジェクト データの上で動作します (メッシュ、カーブなど)。

すべての拘束は、基本的な [共通インターフェイス](#)を共有しており、ここにもまたモディファイアと多くの類似点があります。

拘束の使用

拘束は静的な場面で非常に有用かもしれませんが (それらはオブジェクトの位置/回転/スケールといったことを自動的にこなして手助けしてくれる)、もともとアニメーションのために設計されたもので、絶対位置 (すなわちグローバル空間)、または他のオブジェクトとの相対位置において、オブジェクトの自由を、制限/制御できるようにしてくれます。

また拘束は、内部的に4x4の変換行列のみを使用して処理を行うことに注意してください。特定の拘束を回転や拡大縮小の調整に使用する場合、この情報は ボーンまたは オブジェクトの設定からではなく、行列から導き出されます。特に不均一な回転や負のスケールを組み合わせると、予期しない動作につながるおそれがあります。

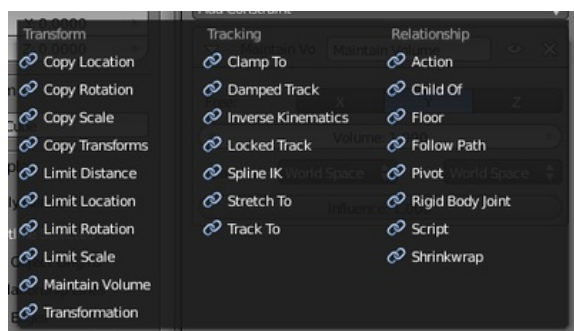
ボーン拘束

最後に、Blenderの偉大なリギング機能があります: ポーズ モードでは、アーマチュアのそれぞれのボーンは、標準的なオブジェクトのように振る舞い、それゆえ、拘束することができます。ほとんどの拘束は、オブジェクトとボーンの両方でうまく動作しますが、いくつかの例外が存在し、関連する拘束のページに記載されています。

さらに詳しく:

- [アニメーションの章](#)で、オブジェクトのアニメーションにするときの拘束の使用についての記事を読みましょう
- [アーマチュア](#)でリギングをするときの拘束の使用についての記事を読みましょう

使用可能な拘束



Constraint menu

拘束にはいくつかの種類があります。3つの系統にそれらを分類することができます:

- [Transform](#)
- [Tracking](#)
- [Relationship](#)

owner オブジェクトに働く拘束もあれば、2番目のオブジェクトを必要とするものもあり (*target*)、時には特殊なタイプもあります (例えば *curve*)。この場合、ターゲットにされる拘束は、オーナーと標的オブジェクトの間に描かれるダークブルーの破線として3Dビューに表示されます。

Transform Constraints

これらの拘束は、そのownerのtransformプロパティを直接制御/制限します。そのターゲットのプロパティに関して絶対的か相対的に関わらず。

Copy Location	ターゲットの位置 (with an optional offset) をownerへコピーし、両方の動きが一緒になるようにします。
Copy Rotation	ターゲットの回転 (with an optional offset) をownerへコピーし、両方が一緒に回転するようにします。
Copy Scale	ターゲットのスケールを (with an optional offset) をownerにコピーし、両方のスケールが一緒になるようにします。
Limit Distance	ownerの位置を制限し、ターゲットから近く/遠く/正確に指定された距離になるようにします。
Limit Location	ownerの位置を、与えられた範囲内に制限します。
Limit Rotation	ownerの回転を、与えられた範囲内に制限します。
Limit Scale	ownerのスケールを、与えられた範囲内に制限します。
Transformation	ターゲットのプロパティ(位置、回転またはスケール)を使用して、ownerのプロパティ(同じまたは別の)を制御します。
Maintain Volume	ボーンまたはオブジェクトの体積を一定にします。

Tracking Constraints

これらの拘束は、さまざまな方法で、ownerのプロパティを調整し、ターゲットを“指す”ようにしたり“ついていく”ようにさせます。

Clamp To	owner を、与えられた曲線ターゲットへ留めつけます。
Damped Track	

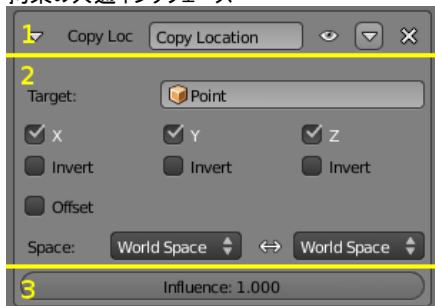
Inverse Kinematics	ボーンのみ。ターゲットによって制御されるボーンのチェーンを作成します。制御には逆運動学を使用します。
Locked Track	所有者は、与えられたターゲットに追跡されますが、与えられた軸の向きでロックされます。
Spline IK	
Stretch To	指定したターゲットにownerを伸ばす。
Track To	ownerは、与えられたターゲットに追跡されます。

Relationship Constraints

これらは“その他”の拘束です。

Action	owner は、ターゲット (driver)によって制御されるアクションを実行します。
Child Of	別のオブジェクトへの親の影響を、選択的に適用できるようにします。
Floor	ターゲットの位置 (および必要に応じて回転) を使用して“壁”や“床”を定義し、ownerが通り抜けられないようにします。
Follow Path	owner は、曲線のターゲットに沿って移動する。
Pivot	
Rigid Body Joint	owner と“target”(子オブジェクト)の間に剛体関節 (ヒンジのような...) を作成します。
Script	Pythonスクリプトを拘束として使用します。
Shrinkwrap	ownerの位置をターゲットの 表面 (among other options) に制限します。

拘束の共通インターフェイス



The three parts of a constraint interface

[モディファイア](#)と同様に、オブジェクト (またはボーン、詳細については [リギングの章](#) 参照) には、一度に複数の拘束を使用することができます。ゆえに、これらの拘束はスタックで構成され、それによって評価の順番を制御するようになっています (上から下へ)。

すべての拘束は、共通の基本的なインターフェイスを共有しており、サブパネルの並びにまとめられ、三つの部分に分かれています:

1. ヘッダー、最も一般的な設定が集まっている。
2. 拘束の特殊な設定を行う。
3. 影響度とアニメーションの制御 (Rigid Body Joint 拘束には影響度の設定はありません)。

拘束のヘッダ



A constraint header

拘束 “サブパネル” のヘッダはすべて同じです。左から右へ、こういう項目があります:

小さな矢印

このコントロールでは、拘束の設定を表示/非表示することができます。上記の (*A constraint stack example*) 図では、Child Of 拘束はこのように省スペース化されます。

拘束のタイプ

これは単に、この拘束が何であるかを示す静的なテキストです...

名前欄

ここではあなたの拘束に、デフォルトの名前よりもっと意味のある名前を付けることができます。このコントロールには、別の *重要な* 用途があります: 拘束が機能していないときに赤色に変わるので (*A constraint header* のように)。ほとんどの拘束は、動作するときに2番目の “target” オブジェクトを必要とし(下記参照)、追加してすぐのときは、“赤の状態”で、Blenderがどのオブジェクトやボーンをターゲットとして使用するか推測できていない状態です。これは無効な組み合わせの設定を選んだ場合にも発生します。例えば [Track To 拘束](#) で、To と Up のベクトルを両方とも同じ軸に設定した場合です。上記に述べたように、“赤の状態”の拘束は、スタックの評価時には無視されます。

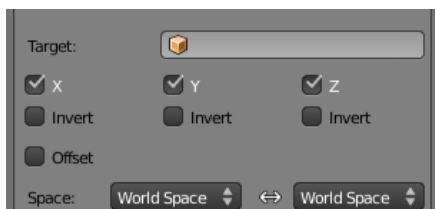
“up”/“down” ボタン

上で見たように、これらはスタック内で拘束を上/下に移動させることができます。

The “X” control

上で見たように、これは拘束を削除 (スタックから削除) します。

拘束の設定



The central part of a constraint's subpanel contains the constraint's settings, the target, and constraint space

拘束の設定のエリアは、もちろん各拘束タイプに固有のもので、しかし、多くの拘束に共通するポイントが2つあるので、ここで解説しておきます。

ターゲット

ほとんどの拘束は、それを“ガイド”する別の “ターゲット” オブジェクトまたはボーンを必要とします。Targetの欄でその名前を選ぶと選択されます。少数の例外を除けば、あらゆるタイプのオブジェクトを使用することができ (camera, mesh, empty...), その中心点がターゲットポイントになります。

OB フィールドにメッシュや格子の名前を入力すると、2番目の Vertex Group 欄がすぐ下に表示されます。それを空白のままにした場合は、メッシュまたは格子が、標準どおりオブジェクトのターゲットとして使用されます。しかし、この Vertex Group の欄に、メッシュか格子の頂点のグループのいずれか

の名前を入力した場合、拘束はこの頂点グループの中央の点をターゲットとして使用します。

同様に、OB 欄にアーマチュアの名前を入力した場合、第2の Bone 欄がすぐ下に出現します。そこにアーマチュアのいずれかのボーンの名前を入力すると、拘束はこのボーンの *root* をターゲットとして使用します。いくつかの拘束では、ターゲットとしてボーンを使用する場合、別の Head/Tail 数値フィールドが現れ、ボーンに沿ってどこにターゲットポイントを置くのかを、ルート (**0.0**) からテイル (**1.0**) (現在、BlenderのUIでは、ボーンのrootは“heads”と呼ばれ、ボーンのtipは“tails”と呼ばれていることを覚えておきましょう...)。

拘束空間 (Space)

多くの拘束では、Owner Space ドロップダウンリストで (右のもの)、どれを評価/適用するのかを空間内で選択することができます。ターゲットを使用するような拘束の場合、Target Space ドロップダウンリスト (左のもの)において、どのターゲットを評価するのかを空間内で選択することができます。両方のリストが同じオプションを持てるかどうかは、要素 (オーナーまたはターゲット) が通常のオブジェクトであるかまたはボーンであるかに依存します。:

Local Space

ボーンのプロパティが自身のローカルスペース、すなわち、そのレストポジションに基づいて評価されます (そのチェーンにおける親ボーンの変形と、そのアーマチュアオブジェクトの変形はどちらも考慮されません)。

Local With Parent (bones only)

ボーンのプロパティは、独自のローカルスペースで評価され、可能な限りの親子関係による変換を *含めず* (すなわちボーンより上のチェーンの変形によるもの)。

Pose Space (bones only)

ボーンのプロパティは、アーマチュアオブジェクトのローカル空間 (すなわち、オブジェクトモードでのアーマチュアの変形からは独立して)で評価されます。したがって、アーマチュアオブジェクトがnull(無効な)変形になっている場合は、Pose Space は World Spaceと同じ効果もちます...

Local (Without Parent) Space (objects only)

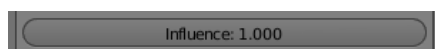
オブジェクトのプロパティは自身のローカルスペースで評価され、親子関係による変形は *されません*。

World Space (default setting)

このオブジェクトまたはボーンのプロパティは、グローバル座標系で評価されます。これは最も理解しやすく自然な動作で、常に“visual”変換プロパティを使用します (すなわち、あなたが3Dビューで目にするものと同様です)。

拘束空間の結果を理解するのは全く容易ではありません (あなたが幾何学の天才でない限り...)。これを理解するために最もよいのは、それらのさまざまな組み合わせを試してみることで、例えば2つエンピティを使ったり (その軸がはっきり実体化される)、および Copy Rotation 拘束を使ってみたり (回転は最も表現的な変換であり、様々なスペースの特異性を可視化してくれます...)。

影響度



Influence

ほとんどすべての拘束の下部には、Influence のスライダーがあり、拘束がその所有者に与える影響を制御できるようになっています。ご想像のとおり、**0.0** は拘束が効果をもたないことを意味し、**1.0** なら拘束がフルに効果を発揮することを意味します。この間の値を使用すれば、同じownerの特性に複数の拘束を持たせてすべてを一緒に作用させることができます。もし拘束が指定されたプロパティへフルに影響を持ってしまうと、スタックの上位にある他のすべての拘束で同じプロパティに作用するものは全く影響を及ぼせなくなってしまうことに注意してください。

とはいえ、影響度の最大の好例は、Ipo curve でそれをアニメートできることでしょう – この詳細については、[アニメーションの章の拘束のページ](#)を参照してください。

拘束のスタック



A constraint stack example

前に言ったように、拘束は、Constraints パネルに表示/実体化されるとおりに、拘束スタックの上から下へと評価されます。

1. (A constraint stack example)では、まずランブの位置がownerのオブジェクトにコピーされます。
2. 回転コピー拘束は無視されます (名前欄が赤くなっている、下記参照)。
3. したがって、次に評価される拘束は Child Of です。現在は折りたたまれています。
4. 最後に、最下段の拘束の Limit Scaleによってキューブのサイズが制限されます。

なので、ここでは、キューブのサイズは最初に Child Of 拘束のターゲットによって制御され、上限は次の Limit Scale 拘束で許可される範囲内になります... モディファイアと同様に、順序は重要です!

そのヘッダで拘束名の右側に描かれている小さな上向き/下向きの矢印ボタンを使用することで、拘束スタックを上下に移動することができます。これらのボタンは必要などきにだけ可視化されます。つまり最上段の拘束には“下”ボタンだけ、最下段の拘束には“上”ボタンだけが表示されます – そして、スタック内に拘束がひとつしかない場合、どちらのボタンも表示されません。

拘束の追加/削除

拘束の追加は、Constraints パネルで... Add Constraint ボタンをクリックすることで行えます! メニューが現れ、現在アクティブなオブジェクトに対して使用可能なすべての拘束が一覧表示されます (またはボーンが Pose モードになっている場合には 拘束はbone constraintsメニューに表示されます)。新しい拘束は、常にスタックの最下部に追加されます。

また、3Dビューで、次のいずれかのようにもできます:

- 将来のオーナーだけを選択し、Ctrl+Shift+Cと押すと、Add Constraint to New Empty Objectメニューがポップアップするので、追加したい拘束を選択しましょう。選ばれた拘束がエンティを必要とする場合は、新しい Empty オブジェクトが自動的にターゲットに追加され、所有者の中央に置かれ、そして null (無効な)回転になります。
- まず、将来のターゲットを選択し、続けて将来のオーナーも選択し、Ctrl+Shift+Cと押すと、Add Constraint to Active Object (または Add Constraint to Active Bone) メニューがポップアップするので、追加したい拘束を選択しましょう。選ばれた拘束が必要とする場合は、他の選択したオブジェクト/ボーンがターゲットとして使用されます。

これらのポップアップメニューは、利用可能なすべての拘束を提示するわけではないことに注意しましょう...

拘束を削除するには、Constraintsパネルで、削除したい拘束のヘッダの“X” ボタンをクリックします。また、すべての拘束を削除するには、選択状態のオブジェクトから、Object » Constraints » Clear Constraints (または Pose » Constraints » Clear Constraints... か Ctrl+Alt+Cと押す)を使用することもできます。

位置コピー拘束

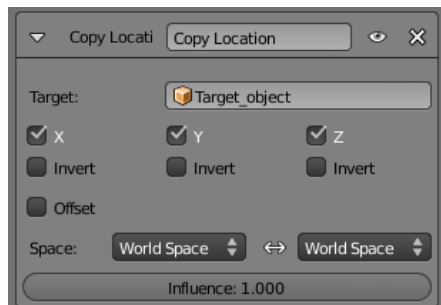
概要

Copy Location 拘束は、そのオーナーがターゲットと同じ位置になるように強制します。



注意点として、このような拘束を *接続された* ボーンに使用する場合、それは効果を持ちません。なので親のタイプが、オーナーのボーンのルートの位置を制御します。

オプション



Copy Location panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

Bone

Target が Armature である場合、新しいフィールドが表示され、個々のボーンを Target として設定するオプションな選択を行えます。

Head/Tail

Bone が Target として設定されている場合、新しいフィールドが表示され、このボーンに沿ってどこにターゲットを置くのか省略可能な選択肢を提供します。

Vertex Group

Target が Mesh である場合、新しいフィールドが表示され Vertex Group をターゲットとして設定するオプションな選択を行えます。

X, Y, Z

これらのボタンで、どの軸 (すなわち座標) を拘束するのかを制御します - デフォルトでは、3つすべてになっています。

Invert

Invert ボタンは、それぞれの向く座標を反転します。

Offset

有効にすると、このコントロールで所有者を、そのターゲットとの相対的な位置に転送 (現在の変換プロパティを使って) することができます。

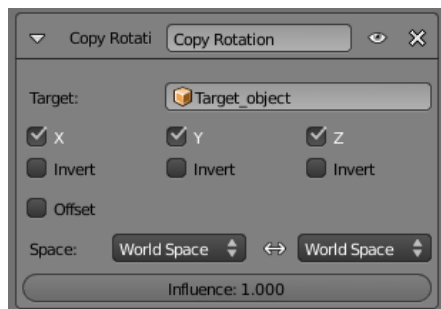
Space

この拘束は、所有者とターゲットのどちらのスペースの変形プロパティで評価を行うのかを選択できるようにします。

回転コピー拘束

Copy Rotation 拘束は、その所有者の回転がその目標の回転と一致するように強制します。

オプション



Copy Rotation panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません(赤の状態)。

Bone

Target が Armature である場合、新しいフィールドが表示され、個々のボーンを Target として設定するオプションな選択を行えます。

Head/Tail

Bone が Target として設定されている場合、新しいフィールドが表示され、このボーンに沿ってどこにターゲットを置くのか省略可能な選択肢を提供します。

Vertex Group

Target が Mesh である場合、新しいフィールドが表示され、Vertex Group をターゲットとして設定するオプションな選択を行えます。

X, Y, Z

これらのボタンで、どの軸を拘束するかをコントロールします – デフォルトでは、3つすべてがオンになっています。

Invert

Invert ボタンは、それぞれの回転の値を反転します。

Offset

有効にすると、このコントロールでオーナーを、そのターゲットとの相対的な向きへ回転させる(現在の変換プロパティを使用して)ことができます。

Space

この拘束は、変形プロパティの評価を、オーナーとターゲットのどちらのスペースで行うのかを選択できます。

スケールコピー拘束

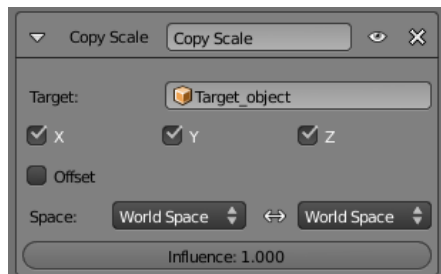
概要

Copy Scale 拘束は、そのオーナーを、そのターゲットと同じ拡大率を持つように強制します。



ここで話すのは **scale**で、**size**ではありません! 実際に、あなたは二つのオブジェクトを、一方を他方よりかなり大きくすることもできますが、それでもまだ両方は同じスケールです。これはまた、ボーンにもあてはまります: ポーズ モードでは、そこにある視覚的な長さがどのようなものであっても、ボーンはすべてレストポジションのときの単一のスケールになっています。

オプション



Copy Scale panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤い表示)。

Bone

Target が Armatureである場合、新しいフィールドが表示され、個々のボーンをTargetとして設定するオプションな選択を行えます。

Head/Tail

Bone が Targetとして設定されている場合、新しいフィールドが表示され、このボーンに沿ってどこにターゲットを置くのか省略可能な選択肢を提供します。

Vertex Group

Target が Meshである場合、新しいフィールドが表示され、Vertex Group をターゲットとして設定するオプションな選択を行えます。

X, Y, Z

これらのボタンは、どの軸に沿ってスケールが拘束されるかをコントロールします – デフォルトでは、3つすべてについて有効になっています。

Offset

有効にすると、このコントロールで、所有者の拡大・縮小を (現在の変換プロパティを使用して)、そのターゲットとの相対的なスケールにもとづいて行うことができます。

Space

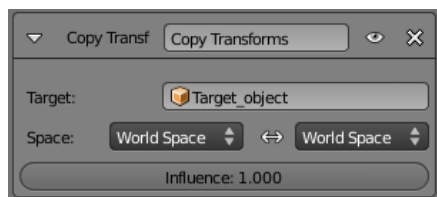
この拘束は、変形プロパティの評価を、オーナーとターゲットのどちらのスペースで行うのかを選択できるようにします。

変形コピー拘束

概要

Copy Transforms 拘束は、その所有者がターゲットと同じ変形をするように強制します。

オプション



Copy Transforms panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

Bone

Target が Armature, である場合、新しいフィールドが表示され、個々のボーンを Target として設定するオプションな選択肢を提供します。

Head/Tail

Bone が Target として設定されている場合、新しいフィールドが表示され、このボーンに沿ってどこに標的点を置くのかという省略可能な選択肢を提供します。

Vertex Group

Target が Mesh である場合、新しいフィールドが表示され、Vertex Group をターゲットとして設定するオプションな選択肢を提供します。

Space

この拘束は、変形プロパティの評価を、その所有者とターゲットのどちらの空間で行うのか選択できるようにします。

距離制限拘束

概要

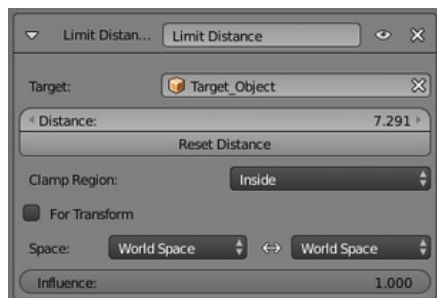
Limit Distance 拘束は、指定したターゲットから、遠く、近く、あるいはきっかりの距離に、所有者を強制します。言い換えれば、所有者の位置を、ターゲットを中心とした球の、外側、内側、または表面のいずれかに拘束します。

あなたが (新しい) ターゲットを指定すると、Distance 値が、所有者とこのターゲット間の距離に合わせて、自動的に設定されます。



注意点として、このような拘束を *接続された* ボーンに使用する場合、効果を持ちません。なので親のタイプが、オーナーのボーンのルートの位置を制御します。

オプション



Limit Distance panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

Bone

Target が Armature である場合、新しいフィールドが表示され、個々のボーンを Target として設定するオプションな選択を行えます。

Head/Tail

Bone が Target として設定されている場合、新しいフィールドが表示され、このボーンに沿ってどこにターゲットを置くのか省略可能な選択肢を提供します。

Vertex Group

Target が Mesh である場合、新しいフィールドが表示され、Vertex Group をターゲットとして設定するオプションな選択を行えます。

Distance

この数値フィールドは、限界距離を設定します。言い換えれば、拘束球の半径です。

Reset Distance

クリックすると、この小さなボタンは Distance の値をリセットし、所有者とそのターゲット間の実際の距離に一致するようにします (つまりこの拘束が適用される前の距離に)。

Clamp Region

Limit Mode ドロップダウンメニューを使用すると、Distance の設定とターゲットの中心によって定義される球をどのように使用するか選ぶことができます:

Inside (default)

所有者は球の *内側* に拘束されます。

Outside

所有者は、球体の *外側* に拘束されます。

Surface

所有者は球の *表面上* に拘束されます。

位置制限拘束

概要

オブジェクトまたは 接続されていないボーンは、X、Y、Z軸に沿ってシーンを移動できます。この拘束は、上限と下限を介して、各軸に沿って移動を認める量を制限します。

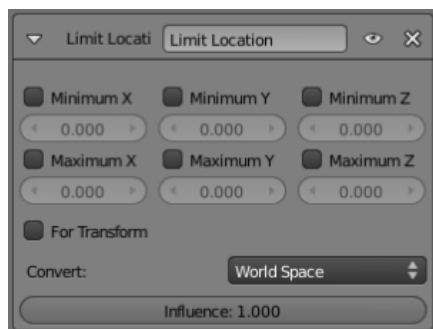
オブジェクトの制限は、その中心から算出され、ボーンの制限は、そのルートから算出されます。

興味深いことに、たとえ拘束がそのオーナーの視覚的なレンダリングされた位置を制限していても、そのオーナーのデータブロックは(デフォルトでは)、まだオブジェクトまたはボーンが最小値と最大値の範囲外の座標を持つことを許容します。これは、その *Transform Properties* パネル (N)で見ることができます。オーナーをつかんで制限範囲の外に移動させようとする、視覚的にもレンダリング時にもその範囲に拘束され、しかし内部的には、その座標はまだ限界を超えて変更されたままになります。拘束が削除された場合、元のオーナーは、その内部的に指定された場所にジャンプするように見えるはずで

同様に、そのオーナーの内部的な位置が制限を超えている場合、制限エリアに引っぱり戻しても、内部位置の座標が制限のしきい値内に戻るまで何もしないように見えます (あなたが *For Transform* オプションを有効にしない限り、以下参照)。

軸の最小値と最大値を等しく設定すると、その軸に沿ってオーナーの動きをロックします... こうすることも可能ではありますが、*Transformation Properties* の軸ロック機能を使用する方が簡単です!

オプション



Limit Location panel

Minimum X, Minimum Y, Minimum Z

これらのボタンは、選択されたSpaceの X, Y, Z 軸それぞれに沿って、オーナーの中心位置の下限を有効化します。その下の数値フィールドは、その制限値をコントロールします。minの値が対応する最大値よりも大きい場合には、この拘束は、maxと同じ値を持っているかのように動作することに注意してください。

Maximum X, Maximum Y, Maximum Z

これらのボタンは、選択されたSpaceの X, Y, Z 軸それぞれに沿って、オーナーの中心位置に対する範囲制限を有効化します。上記と同じオプションがあります。

For Transform

デフォルトで見ると、視覚的には拘束されているにもかかわらず、オーナーはまだ境界の座標外にあります (*Transform Properties* パネルで示すように)。さて、このボタンを有効にすると、もうこれは起こりません - オーナーの変形プロパティは、拘束によって制限されます。拘束は座標を直接変更するわけではないことに注意しましょう: これを有効にするには、どちらにせよオーナーを扱う必要があります...

Convert

この拘束をつかると、どの空間でそのオーナーの変形プロパティを評価するのか選択できるようになります。

回転制限拘束

概要

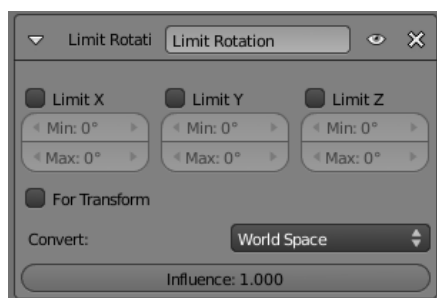
オブジェクトやボーンは、X、Y、Z軸周りに回転させることができます。この拘束は、上限と下限を介して、各軸の周りに許容する回転量を限定します。

興味深いことに、たとえ拘束がそのオーナーの視覚的なレンダリングされる回転を制限していても、そのオーナーのデータ・ブロックは (デフォルトでは)、まだオブジェクトまたはボーンの最小値と最大値の範囲外の回転値を持つことを許容します。これは、Transform Properties パネル (N)で見ることができます。オーナーを回転させて制限範囲の外に回転させようとすると、視覚的にもレンダリング時にもその範囲に拘束されますが、内部的には、その回転の値はまだ限界を超えて変更されたままになります。拘束が削除された場合、その元のオーナーは、内部で指定されたところへ急回転するように見えるはずですが。

同様に、そのオーナーが内部的に制限を超えた回転を持っている場合、制限エリアに戻すように回転させても、内部の回転の値が上限のしきい値内に戻るまでは何もしないように見えます (あなたが For Transform オプションを有効にしない限り、以下参照)。

軸の最小値と最大値を等しく設定すると、その軸の周りにオーナーの回転をロックします... こうすることも可能ではありますが、おそらく Transformation Properties の軸ロック機能を使用したほうが簡単です。

オプション



Limit Rotation panel

Limit X, LimitY, LimitZ

これらのボタンは、選択されたSpaceで、オーナーの X, Y, Z 軸それぞれの周りの回転の制限を有効にします。その右にある Min と Max の数値フィールドは、それぞれの軸の下限と上限の値をコントロールします。

注意事項:

- minの値が対応するmax値より大きい場合、拘束はmaxのものと同じ値を持っているかのように動作します。
- [位置制限の拘束](#)とは異なり、個別に下限や上限の制限を有効にすることはできません...

For Transform

デフォルトでは、たとえ視覚的には拘束されていても、オーナーはまだ境界をこえた回転を持つことができました (Transform Properties panelで示すように)。さて、あなたがこのボタンを有効にすると、これはもはや起こらなくなります - オーナーの変形プロパティもまた、拘束によって制限されます。

しかし拘束が直接回転値を変更するわけではないことに注意しましょう: これを有効にするためには、いずれにしてもオーナーを回転させる必要があります...

Convert

この拘束をつかうと、オーナーの変換プロパティをどの空間で評価するか選択できるようになります。

スケール制限拘束

概要

オブジェクトやボーンはX、Y、Z軸に沿ってスケールリングすることができます。この拘束は、上限と下限を介して、各軸に沿って拡大縮小を許容する量を制限します。



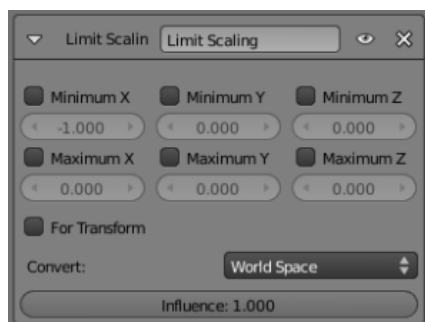
この拘束は、負のスケール値に耐えられません (オブジェクトのミラー化に使用するかもしれないものに...): オブジェクトかボーンにそれを追加するとき、たとえ軸の制限が全く有効にされていない場合でも、または For Transform ボタンが有効にされていない場合でも、オブジェクトを拡大縮小すればすぐに、すべての負のスケール値は瞬時に正の値に反転されます... そして境界の設定は、厳密に正の値だけしか取ることができません。

興味深いことに、たとえ拘束がそのオーナーの視覚的なレンダリングされたスケールを制限していても、そのオーナーのデータブロックはまだ (デフォルトでは)、オブジェクトやボーンが最小値と最大値の範囲外のスケール値を持つことを許容します (正の値である限り!)。これは、Transform Properties パネル (N)で見ることができます。オーナーをつかんで制限範囲の外に移動させようとする、視覚的にもレンダリング時にもその範囲に拘束され、しかし内部的には、その座標はまだ限界を超えて変更されたままになります。拘束が削除された場合、その元のオーナーは、内部的に指定された場所にジャンプするように見えるはずですが。

同様に、そのオーナーの内部のスケールが制限を超えたものである場合、範囲内に戻すほうに縮小しても、内部のスケール値が上限しきい値に入るまでは、何も見えないように見えます (For Transform オプションを有効にしない限り、以下参照 – またはオーナーが負のスケール値を持っている場合)。

軸のminとmaxを等しい値に設定すると、その軸に沿ってオーナーのスケールリングをロックします。こうすることも可能ではありますが、Transformation Properties の軸ロック機能を使用する方が簡単です。

オプション



Limit Scale panel

Minimum/Maximum X, Y, Z

これらのボタンは、選択されたSpaceのX、Y、Z軸それぞれに沿って、オーナーの中心位置の下限を有効化します。その右にあるMinとMaxの数値フィールドは、それぞれその下限と上限の値をコントロールします。minの値が対応する最大値よりも大きい場合には、拘束はmaxと同じ値を持っているかのように動作することに注意してください。

For Transform

デフォルトで見ると、たとえ視覚的には拘束されていて、負の値が除かれていても、オーナーはまだ範囲外のスケールを持つことができます (Transform Properties パネルで示すように)。さて、あなたがこのボタンを有効にすると、これはもはや起こらなくなります – オーナーの変形プロパティもまた、拘束によって制限されます。拘束が直接スケール値を変更するわけではないことに注意してください: 効果を得るには、どちらにせよオーナーを拡大・縮小する必要があります。

Convert

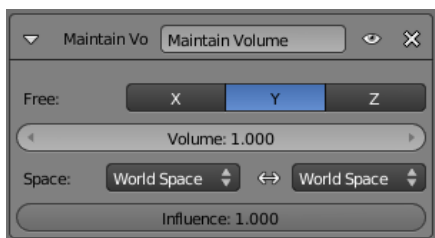
この拘束は、そのオーナーの変形プロパティをどの空間で評価するのか選択できるようにします。

Maintain Volume Constraint

概要

Maintain Volume 拘束は、メッシュやボーンの体積を、元の体積から指定した比率に維持します。

オプション



Maintain Volume panel

Free X/Y/Z

オブジェクトのフリースケール軸。

Volume

ボーンの安静時の体積。デフォルトでは 1.0。

Space

この拘束をつかうと、オーナーの変形プロパティをどの空間で評価するかを選択できるようになります。

See also

- [Harkyman on the development of the Maintain Volume constraint](#), March 2010

Transformation Constraint

概要

この拘束は、他の “transform” 拘束よりも複雑で多用途です。ターゲットの変形プロパティ (すなわち、位置、回転やスケール) のタイプを、オーナーの同じまたは別のタイプの変形プロパティに、指定した範囲の値でマップすることができます (ターゲットとオーナーの各プロパティは異なるかもしれません)。また、座標軸間の切り替え、および限界値としてではなく範囲の値を使用することもできます。しかし、マッピングを定義する “マーカー” は、入力 (ターゲット) と出力 (オーナー) の間の値にしかできません。

だから、例えば、ターゲットのX軸に沿った位置を使用して、オーナーのZ軸周りの回転を制御することができます。ターゲットのX軸に沿った **1 BU** が、オーナーのZ軸の周りの **10°** に対応するといった具合です。これの典型的な用途には、歯車 (下記の注を参照)、および場所のセットアップに基づく回転が含まれます。

歯車について

残念ながら、ギアとしては動作しません。うまく動作しないことを意味する数学に関連する根本的な問題がいくつかあるからです...

オプション



Transformation panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

Bone

Target が Armature である場合、新しいフィールドが表示され、個々のボーンを Target として設定するオプションな選択肢を提供します。

Head/Tail

Bone が Target として設定されている場合、新しいフィールドが表示され、このボーンに沿ってどこに標的点を置くのかという省略可能な選択肢を提供します。

Vertex Group

Target が Mesh である場合、新しいフィールドが表示され、Vertex Group をターゲットとして設定するオプションな選択肢を提供します。

Extrapolate

デフォルトでは、min と max の値が入力値と出力値を範囲内に縛り、この範囲外のすべての値はそこに抑えられます。このボタンを有効にすると、min と max の値はそれほど厳密には制限しなくなり、入力とそれに対応する出力値の間の比例的な (線形) マッピングを “マーカー” が定義します。

2つのグラフで解説していきます (*The Extrapolate principles*)。これらの画像では、入力範囲 (横軸) は **[1.0, 4.0]** に設定されており、それに対応する出力範囲 (縦軸) は **[1.0, 2.0]** です。黄色の曲線は、入力と出力の間のマッピングを表します。

The Extrapolate principles.



Extrapolate disabled: the output values are bounded inside the **[1.0, 2.0]** range.

Extrapolate enabled: the output values are “free” to proportionally follow the input ones.



注意事項:

- 変換プロパティのマッピングをlocationにすると(すなわち Loc Destination ボタンが有効になっていると)、オーナーの既存の位置が、この拘束の評価の結果に追加されます ([Copy Location 拘束](#)のOffset ボタンを有効にしたときとまったく同様です...)
- 逆に、変換プロパティのマッピングを回転やスケールにすると、オーナーの既存の回転やスケールがこの拘束の評価の結果に上書きされます。
- ターゲットの回転変換のプロパティを入力として使用する場合、実際の値が何であれ、拘束は常に $[-180^\circ, 180^\circ]$ の範囲に “それらを戻” します (例えば、ターゲットがX軸の周りに 420° の回転を持っている場合、拘束によってXの入力として使用される値は次のようになります $((420 + 180) \bmod 360) - 180 = 60^\circ \dots$)。この拘束が実際に歯車には適さないのはこれが理由です!
- 同様に、ターゲットのスケール変換プロパティを入力として使用する場合、実際の値が何であれ、拘束は常にその絶対値をとります (つまり負のものは反転される)。
- min の値が対応する max の値より高い場合は、両方とも max の値に等しいと見なされます。これは “裏返し” のマッピングは作成できないことを意味します...

Source

ここには、入力 (ターゲットから) の設定が含まれています。

Loc, Rot, Scale の3つのトグルボタンは、相互に排他的で、どの種類のプロパティを使用するのか選択することができます。X, Y, Z の min と max の min と max の数値フィールドは、各軸ごとに独立して、入力値の範囲の下限と上限をコントロールします。min の値が対応する max の値よりも大きい場合には、拘束は max と同じ値を持っているかのように動作することに注意してください。

Destination

ここには、出力 (オーナーへ) の設定が収められています。

- Loc, Rot and Scale の3つのトグルボタンは、相互に排他的で、どの種類のプロパティでコントロールするのか選択できます。
- 3つの Axis Mapping ドロップダウンリストを使用すると、どの入力軸で、X, Y, Zそれぞれ(上から下)の出力 (オーナー) 軸をマッピングするかを選択することができます。
- min と max の数値フィールドは、マップされたそれぞれの軸ごとに独立して、出力値の範囲の下限と上限をコントロールします。min の値が対応する最大値よりも大きい場合には、拘束は max と同じ値を持っているかのように動作することに注意してください。

Space

この拘束は、変形プロパティの評価を、そのオーナーとターゲットのどちらの空間で行うのか選択できるようにします。

Clamp To Constraint

概要

Clamp To 拘束は、物の移動を、手書きでは難しい大規模で複雑なパスに沿って行わせるときに特に有用な拘束です。

Clamp To 拘束のアイデアは、[Follow Path](#) のものと非常によく似ていますが、大きな違いがひとつあります: 後者はターゲットカーブに time IPO を使用し、Clamp To はそのオーナーの実際の位置のプロパティを取得し (Transform Properties パネルに示されている、N)、そしてターゲット・カーブに沿って “mapping” してどの場所にそれを置くのか判断します。

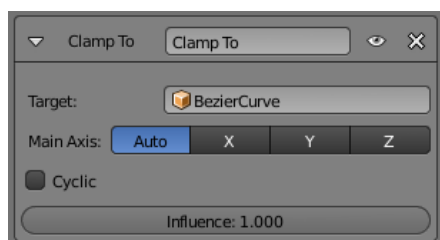
大抵の物事と同様に、もちろん、明るい面と暗い面が存在します。

明るい面は、Clamp To をつかって作業すると、オーナーがどのような動きをするのか確認しやすくなるということです。3Dビューで作業するとき、time IPO でキーをあちこちスライドさせたり繰り返しアニメーションを再生させるよりはるかに的確です。

ダークサイドは、[Follow Path 拘束](#) のそれとは異なり、Clamp To はオーナーの回転 (ピッチ、ロール、ヨー) を追跡する任意のオプションを持たず、ターゲット曲線に傾斜をつけられないことです。とはいえ、いつも回転を必要とするわけではなく、このような場面では、Clamp To を起動してから、回転が必要になったら何か他の方法でさせたほうが手軽でしょう。

まとめると、様々な動きをカーブをまたいでアニメーション化するとき、おそらく Follow Path を使用するよりも簡単に行えるはずですが。しかし簡単でなかったとしても、興味深い代替策となり、個人的な選択肢にはなるでしょう。

オプション



Clamp To panel

Target

この拘束は1つのターゲットを使用し、これは *曲線のオブジェクトでなければならない*、それを持っていないときは機能しません (赤の状態)。

Main Axis

このボタンのグループでは、ターゲット・カーブに沿ってどのグローバル軸 (X、Y または Z) を、オーナーを配置する基準として使用するかを制御します。

間違えそうな選択項目は全くなく、軸を選択するだけで簡単に動作してくれて、また現在の状況で最適に動作してくれます。ターゲット・カーブの最長の軸を選ぶのは良いアイデアで、そのようにすればオブジェクトのグローバルおよび拘束される位置がお互いにより似かよいます – またデフォルトの Auto オプションをそのまま使用することもでき、最適になるように Blender が軸を推測してくれます。

Cyclic

デフォルトでは、同じ方向に移動させ続けた場合、オーナーがそのターゲット曲線の一方の端に達すると、そこで立ち止まります。Cyclic オプションを有効にすると、カーブの一方の端に達したとき、瞬時にそのもう一方の端に移動されます。

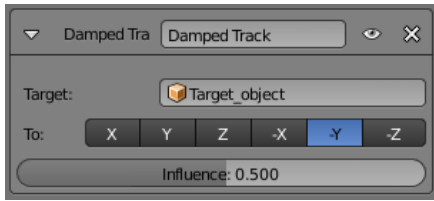
これはもちろん、主に閉じた曲線 (円 & co) のために設計されたものなので、オーナーは何度でもそのまわりを移動することができます。

Damped Track Constraint

概要

Damped Track 拘束は、オーナーのいずれかのローカル軸が、常に Target の方向を指すように拘束します。

オプション



Damped Track panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません(赤の状態)。

Bone

Target が Armature である場合、新しいフィールドが表示され、個々のボーンを Target として設定するオプションな選択肢を提供します。

Vertex Group

Target が Mesh である場合、新しいフィールドが表示され、Vertex Group をターゲットとして設定するオプションな選択肢を提供します。

To

追跡するローカル軸 (デフォルトでは Y)、すなわちオーナーの軸のうちのターゲットのほうを指すもの。負のオプションは、対応する軸がターゲットとは逆方向を指すように強制します。

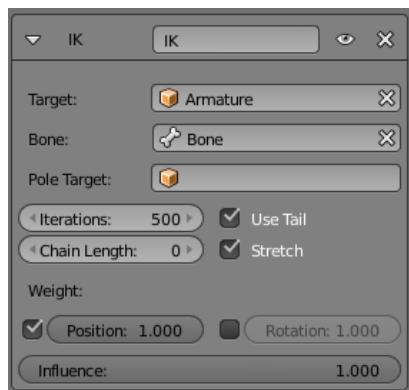
IK Solver Constraint

概要

Inverse Kinematics 拘束は、アーマチュアのポーシングのテクニックである *inverse kinematics* を実装するものです。したがって、ボーンにのみ使用可能です。

この拘束は、リギングの章の [inverse kinematics page](#) に詳しく記載されています。

オプション



Inverse Kinematics panel

Target

アーマチュアでなければなりません

Bone

アーマチュア内のボーン

Pole Target

ポールの回転のためのオブジェクト

Iterations

反復を解決するための最大数

Chain Length

どのくらい多くのボーンがIK効果に含まれるか。0に設定するとすべてのボーンを含めるようになる

Use Tail

チェーン内の最後の要素としてボーンのtailを含める

Stretch

IKストレッチングを有効にする

Weight

Position

For Tree-IK: このターゲットの位置制御のWeight

Rotation

チェーンがターゲットの回転に従うようにする

Target

ターゲットなしのIKを無効にする

Rotation

チェーンがターゲットの回転に従うようにする

Locked Track Constraint

概要

Locked Track 拘束は、グラフィックやテキストのどちらをつかっても、説明が少しまわりくどくなってしまいます。基本的に、それは [Track To 拘束](#) なのですが、軸のロックを伴い、つまり軸回転させることは(その向きを変更することは)できません。したがって、オーナーはこの軸周りを回転することのみによってそのターゲットを追跡することができ、ターゲットがロックされた軸に垂直な平面にあるのでない限り、およびオーナーと交差しているのでない限り、オーナーはそのターゲットを本当に指すことはできません。

現実世界でこれに相当する最良のものを見てみましょう: コンパスです。回転してそのターゲットの(磁北、または近くの磁石)大まかな方向を指すことはできますが、それを直接指すことはできず、これは軸の上で車輪のように回転するためです。コンパスがテーブルに置かれ、磁石がそのまっすぐ上にある場合、コンパスは磁石を指すことはできません。コンパスの複数の側に磁石を移動させる場合、ターゲットを指すことはできないままですが、しかしターゲットの大まかな方向を指すことはでき、そして芯の軸もその制限に従ったままになります。

Locked Track 拘束を使用するとき、ターゲットを磁石として、羅針盤をオーナーとして考えることができます。Lock の軸はオーナーがスピンする芯の軸として機能し、To の軸はコンパスの針として機能します。どの軸に何をさせるかはあなた次第です!

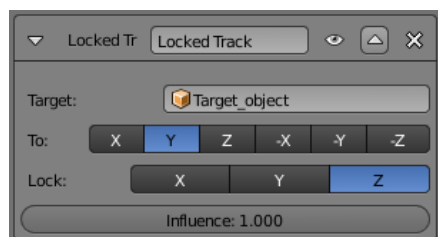
この拘束のボタンの理解に苦心するようなら、tool-tipsを読むと、かなり良いでしょう。オブジェクトの軸がどこにあるかわからない場合は、Object メニューのDraw パネルで Axis ボタンをオンにしましょう。または、あなたがボーンで作業している場合、Armature メニューの Display パネルで、Axes ボタンをオンにしましょう。

この拘束は、Track To 拘束と協調動作するように設計されています。これらの2つの拘束のための軸ボタンを適切に設定すれば、Track To を使用して第一ターゲットの車軸を指すようにでき、Locked Track でオーナーを第二ターゲットの軸の周りにスピンさせることができます。

この拘束は、2Dビルボードにも非常にうまく動作します。

これに関連する話題はすべて、[2.49 BSoD tracking tutorial](#)で長々と論じられています。

オプション



Locked track panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

To

追跡するローカル軸 (デフォルトではY)、すなわちオーナーの座標軸のうちのターゲットのほうを指すもの。負のオプションは、対応する軸がターゲットとは反対を指すように強制します。

Lock

ロックされるローカル軸 (デフォルトではZ)。すなわち、オーナーの軸は、ターゲットを追跡するために再配向することができなくなります。



To と Lock に同じ軸を選択した場合、拘束は機能しません (赤状態)。

Spline IK Constraint

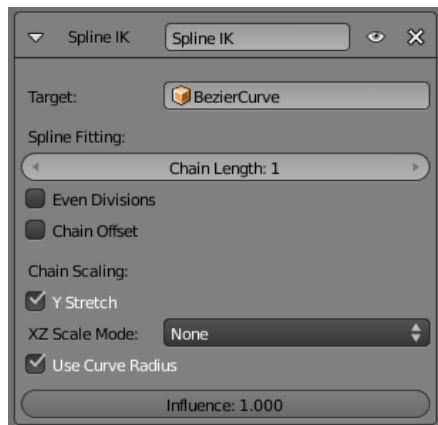
概要

Spline IK 拘束は、曲線に沿ってボーンの連鎖を整列させます。使いやすさと柔軟性を倍増させ、曲線と予測可能性およびボーンによる統合制御によって、美意識にかなう形状を達成できるようにします。Spline IK はリグ職人の工具箱の貴重なツールとなります。それはロープのような無機的なものだけでなく、尾、触覚、そして棘のような、体の柔軟な部位のリギングにも特に適しています。

Spline IK のセットアップに必要なものは、接続されているBoneのチェーンと、これらのボーンを拘束するためのカーブです。

1. 選択されたチェーンの最後のボーンで、Properties Editor内のBone ConstraintsタブからSpline IK 拘束を追加します。
2. 'Chain Length' の設定を、チェーン内のボーンの数に設定します (選択したボーンから開始し、含みます)。これが曲線によって影響されます。
3. 最後に、Target へ曲線を設定します。これで曲線を制御します。

オプション



Spline IK panel

Target

ターゲット曲線

Spline Fitting

Chain Length

どのくらい多くのボーンがチェーンに含まれるか

Even Division

カーブにフィットさせる際に、ボーンの相対的な長さを無視する

Chain Offset

root 関節との相対的な、チェーン全体のオフセット

Chain Scaling

Y stretch

カーブに合うようにボーンのY軸を伸ばす

XZ Scale Mode

None

XとZ軸を拡大縮小しない (デフォルト)

Bone Original

ボーンの名元のスケールリングを使用する

Volume Preservation

X軸とZ軸のスケールを、Yスケールの逆数にする

Use Curve Radius

エンドポイントの平均半径を、XとZのスケールモードの上で、ボーンの名元のスケールを調整するために使用します

See also

この主題のより深いところは、[Rigging/Posing section](#)で見ることができます。

- [Blender.org 2.56 Release Log for Spline IK](#)

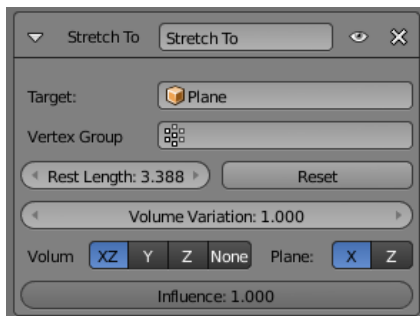
Stretch To Constraint

Stretch To 拘束は、そのオーナーを回転させるとともに、そのターゲットに向かってY軸方向に拡大します。ですから、追跡の動作には [Track To constraint](#) と同じような部分もあります。しかし、Y軸を追跡と引き伸ばしに軸とすることを前提としており、別の軸を使用するオプションは用意されていません。

また、オプション的に未加工の体積測定機能も持ち、ターゲットが近づけるようにオーナーを押し潰したり、ターゲットが遠くに離れられるように薄く延ばすこともできます。しかしながら、これはそのようなオーナーの保持する実際のポリウムではなく、そのスケールの値によって定義された仮想的なものだということに注意しましょう。したがって、この機能は、エンティ、2Dメッシュやサーフェス、カーブのような、非容積オブジェクトでも動作します。

ボーンでは、“体積の” 変動がそれら自身のローカル軸に沿ったスケールをさせます (ボーンのローカルY軸は、rootからtipに整列されていることを覚えておきましょう)。

Options



Stretch To panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

Bone

Target がアーマチュアの場合は、ボーンのための新しいフィールドが表示されます。

Head/Tail

ボーンをターゲットとして使用するとき、このボーンに沿ってどこに目標点を置くのか選ぶことができます。

Vertex Group

Target がメッシュである場合、新しいフィールドに、どこの頂点グループが選択できるかが表示されます。

Rest Length

この数値フィールドは、オーナーとそのターゲットの間の静置距離を設定します。すなわち、オーナーが変形 (伸縮) しない距離を。

Reset

クリックすると、この小さなボタンは Rest Length の値を再計算し、オーナーとそのターゲットの間の実際の距離に一致するようにします (すなわち、この拘束が適用される前の距離に)。

Volume Variation

この数値フィールドは、伸縮の量に比例した “体積” 変化の量を制御します。注意点として、**0.0** の値は許可されていないので、ポリウム機能を無効にしたい場合は None ボタンを使用しましょう (下記参照)。

Volume

これらのボタンは、XおよびまたはZのどちらの軸に影響を与えたらよいかをコントロールし (scaled up/down)、Y軸に沿って伸ばす際に仮想体積を維持します。

NONE ボタンを有効にすると、体積の機能は無効になります。

Plane

これらのボタンは、[Track To constraint](#) という Up と同等のもので: Y軸でターゲットを追跡させるときに、X軸またはZ軸のどちらを維持するのか (可能な限り)、グローバルZ軸に揃えたらよいかを制御します。

Track To Constraint

概要

Track To 拘束は、そのオーナーに回転を適用し、指定された "To" の軸がそのターゲットの方向を指すようにします。別の "Up" 軸は、可能な限りグローバルZ軸(デフォルトでは)に揃うように恒久的に維持されます。この追跡は、3Dにおける "billboard tracking" に似ています(下記の注を参照)。

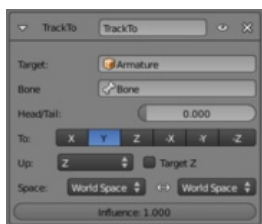
これは最適なトラッキング拘束で、拘束の仕組みを制御しやすいです。

この拘束は、ある意味で、[Inverse Kinematics constraint](#) に密接に関係しています。リグの設計において非常に重要であり、[2.49 BSoD tracking tutorial](#)の、これらの両方の拘束の使いかたを中心に、読んで理解しておいたほうがよいでしょう。

 **Billboard tracking**

"billboard" という用語は、リアルタイムCGプログラミング(すなわちビデオゲーム!)では特定の意味を持っています。常に平面オブジェクトがカメラのほうを向くときに使用されます(それらは確かに "trackers" であり、カメラがその "target" です)。主な用途は、ツリーまたは霧のテクスチャのサポートです: もしにそれらが永続的にはカメラのほうに面を向けなくなってしまうと、木が平べったく潰れて何も無くなってしまったり、霧がミルフィーユの中身に変身するところを頻繁に見ることになるでしょう。これは面白いですが、説得力は全く無くなってしまいうでしょう。

オプション



Track To panel

Targets

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません(赤の状態)。

Bone

Target がアーマチュアの場合、ボーンのための新しいフィールドが表示されます。

Head/Tail

ボーンをターゲットとして使用するとき、このボーンに沿ってどこに目標点を置くのか選ぶことができます。

Vertex Group

Target がメッシュである場合、新しいフィールドに、どの頂点グループが選択できるかが表示されます。

To

追跡するローカル軸(デフォルトではY)、すなわちオーナーの軸のうちのターゲットのほうを指すもの。負のオプションは、関連する軸がターゲットから逆の方向を指すように強制します。

Up

"ほとんどのとき上向き" ローカル軸(デフォルトではZ)、すなわちオーナーの軸のうち(できるだけ)グローバルZ軸(または Target ボタンが有効になっている場合は、ターゲットのZ軸)に揃えられるもの。

Target Z

デフォルトでは、オーナーのUp 軸は、トラッキング回転時には、グローバルZ軸に(できるだけ)揃えられます。このボタンが有効になっている場合、Up 軸は、ターゲットのローカルZ軸に(できるだけ)整列されます...

Space

この拘束は、変形プロパティの評価を、そのオーナーとターゲットのどちらの空間で行うのか選択できるようにします。



To と Up で同じ軸を選択してしまうと、拘束は機能しません(赤の状態)。

Action Constraint

Description

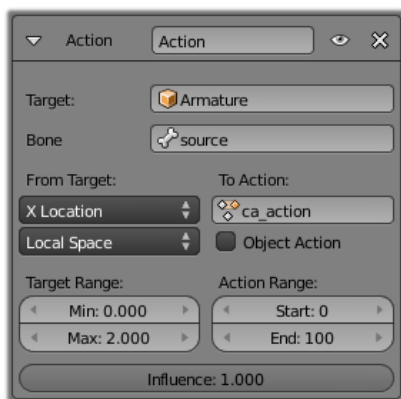
Action 拘束は非常に強力なものですが、理解するのが少し難しいです。基本的には、アクションをオーナーにマップすることができ、ターゲットのいずれかのtransformプロパティを使用して、オーナーをアクションのどこに（もしくは、いつ）置くかを制御します。

ええ、わかっています、これではさっぱりでしょう。まず第一に、Blenderにおける [action](#) とは何であるか知っておいたほうがよいでしょう、他にもリンクをたどって... そしてそれらを知っている人に向けてですが、Action 拘束の基本的な考え方は、[lpo drivers](#)の背後にあるものと非常によく似ていて、ただし前者はアクション全体使用し（すなわち、同じタイプのIPOカーブの束）、後者の場合はその“owner”のlpoカーブを単独でコントロールします...

拘束が Mesh のアクションタイプを受諾していても、実際に作用するのは Object, Pose および Constraint タイプのみであって、拘束が影響を与えるのはメッシュの形状ではなくオブジェクトかボーンの変換プロパティのみだということに注意してください...

例として、Object のアクションを定義するときのことを考えてみましょう（任意のオブジェクトに割り当てることができます、またはオブジェクトがまったく無くても）。そして Action 拘束を介してオーナーにそれをマッピングして、ターゲットを移動がX軸に沿った範囲で [0.0, 2.0] になるようにし、アクションの中身をオーナーにマップして、[0, 100] フレームの範囲に指定します。これは、ターゲットのXプロパティが **0.0**の場合、オーナーはリンクされたアクションのフレーム **0** にいる場合と同様になることを意味します。ターゲットのXプロパティが **1.0**だと、オーナーはリンクされたアクションのフレーム **50** にいる場合と同様になります、etc.

オプション



Action panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません（赤の状態）。

Action

使用するアクションの名前を選択します。



たとえ赤の状態ではないかもしれないとしても（UIリフレッシュの問題...）、このフィールドに有効なアクションが含まれていない場合、この拘束は明らかに機能しなくなります。

Transform Channel

このドロップダウンリストでは、どのターゲットからの変換プロパティ（そのいずれかの軸に沿った / 周りの位置、回転またはスケール）を“action driver”として使用するかをコントロールします。

Action Length Start/End

マッピングされるアクションの開始および終了フレーム

注意事項:

- これらの値は厳密に正でなければなりません。
- デフォルトでは、両方の値は **0** に設定され、それによりマッピングは無効になっています（それによりマッピングは無効になっています（すなわち、オーナーは、リンクされているアクションのフレーム **0** で定義されたプロパティを取得します...）。

Target Range Min/Max

駆動変換プロパティの上限と下限の値。

デフォルトでは、両方の値が **0.0** に設定されています



残念ながら、ここで再び拘束の限界にぶつかります:

- 回転プロパティを“driver”として使用する場合、これらの値は [-180.0°, 180.0°] の範囲に“戻されてマップされます”。
- スケールプロパティを“driver”として使用する場合、これらの値はnullまたは正の値に制限されます。

Convert

この拘束は、そのオーナーの変形プロパティをどの空間で評価するのか選択できるようにします。

Notes

- リンクされたアクションが位置プロパティのいくつかに影響を与える場合、オーナーの既存の位置が、この拘束の評価結果に追加されます（正確には [Copy Location constraint](#) の Offset ボタンが有効になっているとき...）。
- リンクされたアクションがスケールプロパティのいくつかに影響を与える場合、オーナーの既存のスケールは、この拘束を評価した結果と乗算されま

す。

- リンクされたアクションが回転プロパティのいくつかに影響を与える場合、オーナーの既存の回転は、この拘束を評価した結果で上書きされます。
- 通常とは異なり、Start の値を End のものより高くでき、また Min の値を Max のものより高くできます: これはアクションのマッピングを裏返しにします (すなわち、逆 “再生” します...)。両方の設定を逆さまにしない限り、明らかに!
- Constraint actionを使用する場合、アクションに適用されるのがどのオーナーの拘束かを決定するときに使用されるのは拘束 チャンネルの名前です。例えば、“trackto_empty1”という名前の拘束チャンネルを持っている場合、そのキーの与える影響 および/または Head/Tail の値 (キーにできるのは一方のみ) は “trackto_empty1”という名前の拘束のいずれかのオーナーにマップされます。
- 同様に、Pose アクションを使用する場合 (明らかに、ボーンを拘束する際にのみ意味を持ち機能するものです!)、どのボーンの チャンネルの名前をアクションから使用するかをボーンの名前を使用して決定します (例えば、拘束されているボーンに “arm”という名前が付けられている場合、それが使用され、“arm”という名前のボーンチャンネルのアクションだけが使用されます...)。残念ながら、アーマチュアオブジェクト全体の Pose アクションを使用している場合 (一度のアクションでキーのついているすべてのボーンに影響する)、動作しません...
- また [pose library feature](#) を Pose action datablockの作成/編集に使用できることにも注目しましょう... このような状況ではちょうど、1フレームごとにポーズがあることを忘れずに!

Child Of Constraint

Description

Child Of は、オブジェクト間の標準的な親子関係の拘束バージョンです (3Dビューで、CtrlP のショートカットを通して確立されたもの)。

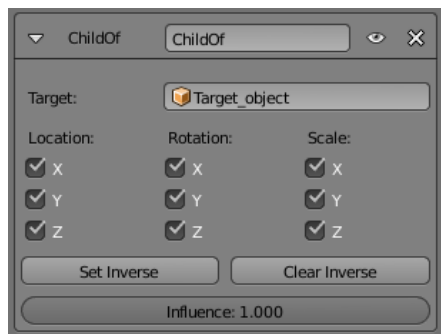
拘束を使った親子化は、従来の方法に比べて、いくつかの利点および機能強化があります:

- 同じオブジェクトに複数の異なる親を持たせることができます (それぞれの影響力の重みを... Influence スライダーで)。
- すべての拘束と同様に、そのInfluence設定をキー (すなわちアニメーション) にでき、したがって、例えばあなたのオーナーを開始時の最初のオブジェクトに完全に親子化し、アニメーションの終了時は別のオブジェクトにすることができます。



この "basic" なオブジェクトの親子化を、アーマチュア内部の [chains of bones](#) が定義するものと混同しないようにしましょう。この拘束を使用して 行えるのは、オブジェクトをボーンに親子化すること (いわゆる "[object skinning](#)")、または同様にボーンにボーンを親子化することです。しかし、ボーンのチェーンを定義するためにそれを使おうとしないでください。

オプション



Child Of panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。Target がアーマチュアまたはメッシュの場合、新しい名前フィールドが表示され、Bone や Vertex Group の名前を選択できます。

Location X, Y, Z

これらのボタンはそれぞれ、対応する軸に沿った位置に親が影響を与えるようにします。

Rotation X, Y, Z

これらのボタンはそれぞれ、対応する軸周りの回転に親が影響を与えるようにします。

Scale X, Y, Z

これらのボタンはそれぞれ、対応する軸に沿った拡大縮小に親が影響を与えるようにします。

Set Inverse

デフォルトでは、オーナーをターゲットの親にするときに、ターゲットがオーナーの空間の原点になります。これは、オーナーの位置、回転、スケールがターゲットの同じプロパティによって相殺されることを意味します。言い換えれば、オーナーは、それをターゲットの親にするときに変換されます。これは望ましくないかもしれません! ですので、オーナーを親子になる前の状態に復元したい場合は、Set Inverse ボタンをクリックしましょう。

Clear Inverse

このボタンは上記のもの効果を反転し (取り消し)、そのターゲット/親に関しての、オーナー/子のデフォルトの状態に復元します。

Tips

この拘束を使用して、新しい親子関係を作成するとき、通常は親を割り当てた後に Set Offset ボタンをクリックする必要があります。上で述べたように、これは親からの不要な変換を相殺するので、オーナーは拘束が適用される前にあった位置/回転/スケールに戻ります。気をつけてほしいのは、Set Inverse の適用は、他のすべての拘束、特にChild Of拘束を無効にした状態で (それらの Influence に設定して **0.0**) 行い、そしてターゲット/親を変換する前に行う必要があります (下記の例を参照)。

ターゲットの (すなわち、親の) 個々のトランスフォームプロパティがオーナーに与える影響を制御するトグルボタンについては、通常すべて有効のままにすることが最善で、または与えられた変換において全体の3つのものを無効にするのがよいでしょう。

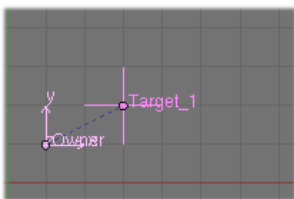
この拘束をすべてのチャンネルでオンにして使用する場合は、親の関係にストレート行列乗算を使用し、親の行列をloc/rot/sizeに分解することはありません。これは、正しく適用された任意の変換、さらに回転や不均一にスケーリングされた親の組み合わせにも当てはまります。

例

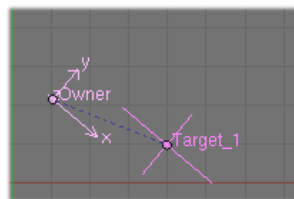


1. No constraint

Owner となるエンペティの位置に注目してください – X軸とY軸に沿って**1.0 BU**。

**2. Child Of just added**

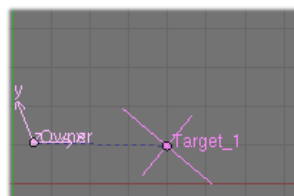
ここでは Owner のエンペティが今、X軸とY軸に沿ってTarget_1となるエンペティから **1.0 BU** 離れているところを見ることができます。

**3. Offset set**

Set Inverse がクリックされると、Owner が元の位置に戻ってきます。

**4. Target/parent transformed**

Target_1 がXY平面に変換され、Z軸周りに回転され、そしてその local X軸に沿って拡大されています。

**5. Offset cleared**

Clear Inverse がクリックされている – Owner は再び Target_1によって完全に制御される。

Set Offset がもう一度クリックされている。見てのとおり、(Target/parent transformed)と同じ結果をもたらすことはありません。上記に述べたように、Set Inverse は一度だけ、ターゲット/親を変換する前に使用しましょう。

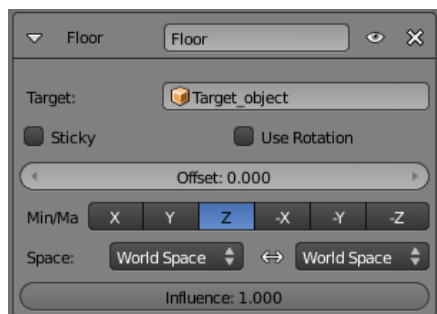
6. Offset set again

床拘束

概要

Floor 拘束は、そのターゲットの位置 (および必要に応じて回転も) を使用して、平面をつかって"侵入禁止サイド"を指定し、オーナーが通り抜けられなくできます。この平面はお望みの方向にすることができます。つまりは、床 (あるいは天井、あるいは壁...)を作り出します!

オプション



Floor panel

Targets

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

Bone

Targetがアーマチュアの場合は、ボーンのための新しいフィールドが表示されます。

Vertex Group

Targetがメッシュである場合、新しいフィールドに、どこの頂点グループを選択できるかが表示されます。

Sticky

このボタンをクリックすると、所有者を、“floor” 平面に触れたときに、動かないようにできます (さらに平面の表面上のを周りにスライドすることすらできません)。これは歩いたり走ったりするアニメーションを作るときには素晴らしいものです!

Use Rotation

このボタンをクリックすると、ターゲットの回転を考慮に入れるよう拘束に強制します。これは、あなたの望む向きの“floor” 平面にすることができ、グローバルなXY、XZ、YZではありません...

Offset

この数値フィールドでは、ターゲットの中心から、Blender単位で指定する数値だけ“floor” 平面からオフセットをとることができます。例えば、足のボーンから足のメッシュの表面への距離を考慮するときに使用します。

Max/Min

このボタンのセット (相互に排他的) は、平面のどちらが“floor”になるかを制御します。ボタンの名前はこの平面の *法線* にちょうど対応します (例えば、Zを有効化すると、“XY平面”といった意味です)

デフォルトでは、これらの法線は *グローバル軸* に揃えられています。しかし、Use Rotationを有効にすると (上記参照)、ターゲットのローカル軸に整列されます。

この拘束は通り抜けられない平面を定義するだけでなく、どちらの側を所有者に侵入禁止するかも定義できます。どちら側にするかは垂直軸の正または負のいずれかを有効にすることで選ぶことができます... 例えば、デフォルト (Z)では、オーナーは正のZ座標に固定されています。

Space

この拘束は、変形プロパティの評価を、オーナーとターゲットのどちらのスペースで行うのかを選択できます。

Follow Path Constraint

概要

Follow Path 拘束はそのオーナーを *曲線の* ターゲットオブジェクトに配置し、この曲線(またはパス)に沿って移動するようにします。また、Follow Curve オプションが有効になっている場合、オーナーの回転に影響を与えて、カーブの曲がりに従うようにすることもできます。

オーナーは、常にグローバル (world) の空間で評価されます:

- その位置は (Transform Properties パネルで示されているように、N)、パス上の通常の位置からのオフセットとして使用されます。例えばオーナーが (1.0, 1.0, 0.0) の位置にある場合、X軸とY軸に沿って、曲線上の通常の位置から1 BU離れます。したがって、オーナーをそのターゲット・パスに くっつけたい場合は、その位置をクリアしましょう (AltG)!
- このオフセット位置も、ターゲット曲線のスケールによって比例的に影響されます。上記と同じようにオフセット (1.0, 1.0, 0.0) をとって、曲線が (2.0, 1.0, 1.0) のスケールをもっている場合、オーナーはX軸に沿って 2 BU オフセットされます (そしてY軸に沿って 1)...
- Curve Follow オプションが有効になっている場合、その回転もまた、曲線によるものからオフセットされます (すなわち、オブジェクトのY軸を曲線の方向と一致させたい場合は、それは必ず、restの、非拘束の状態において、グローバルなY軸に沿っていなければなりません)。ここで再び、オーナーの回転のクリア (AltR) が役に立つかもしれません...

ターゲットカーブパスに沿ったオーナーの動きは、2つの異なる方法で制御されることがあります:

- 最も簡単なのは、動きのフレーム数を定義することであり、曲線のCurveメニューにある Path Animation パネルで、数値フィールドのFrames を *経由*して、およびその開始フレームでの拘束のOffsetオプション *経由*で行えます (デフォルトでは、start frame: 1 [= offset of 0]), 持続時間: 100)。
- 第二の方法 – はるかに正確で強力な – は Speed interpolation curve を Target pathに対して定義することです (Graph Editorで、⇧ ShiftF6)。パスに沿っての開始位置は、IPOの値 0.0に、および終了位置は、IPOの値 1.0に対応します。したがって、開始フレーム、移動速度、終了フレームを制御でき、さらにオブジェクトをパスに沿って強制的に行ったり来たりさせることもできます! IPOカーブの詳細を学ぶためには、[animation chapter](#) を参照してください。
- オーナーをパスに沿って移動させたくない場合は、ターゲット・カーブに flat Speed lpoを与えることもできます (その値がパスに沿ったオーナーの位置を制御します)。

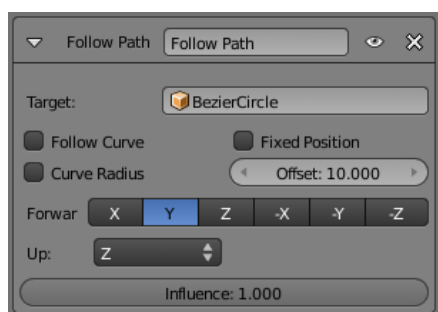
Follow Path は、[Locked Track 拘束](#)とうまく併用できる拘束のひとつです。一つの例は、パス上に飛ぶカメラです。カメラのロール角を制御するために、Locked Track とターゲットオブジェクトを使用して上となる方向を定め、カメラがパスに沿って飛ぶようにできます。

Follow Path と Clamp To

これら2つの拘束を混同しないようにしましょう。それらは両方とも曲線に沿ってオーナーの位置を拘束しますが、しかし Follow Path は “アニメーション-only” の拘束であり、曲線に沿ったそのオーナーの位置が、時間 (すなわち、現在のフレーム)によって決定され、それに対して [Clamp To](#) constraint は、その位置プロパティの値のひとつを使用して曲線に沿ったそのオーナーの位置を決定します。

また、パスはキーフレームのEvaluation Timeを必要とすることにも注意しましょう。パスを選択し、パスのプロパティに移動し、フレーム全体をパスの最初のフレームに設定し (例えばフレーム1)、パスの最初のフレームに評価時間の値を設定し (例えば1)、評価時間を右クリックし、create keyframeを選択し、フレーム全体をパスの最後のフレームに設定し (例えば、フレーム100)、パスの最後のフレームに評価時間の値を設定し (例えば100)、評価時間を右クリックし、create keyframeを選択します。

オプション



Follow Path panel

Target

この拘束は1つのターゲットを使用し、これは *曲線のオブジェクトでなければならず、それを持っていないときは機能しません (red state)*。

Curve Radius

曲線半径によるオブジェクトのスケール

Fixed Position

オブジェクトは、時間に関わらず、曲線の伸びる方向に沿ってどこか一点にロックされたままになります

Offset

パスで定義されている “アニメーション” からオフセットするフレームの数 (デフォルトでは、フレーム 1から)。

Follow Curve

このオプションが有効化されていない場合、オーナーの回転は曲線によって変更されません。それ以外の場合は、次のオプションに応じて影響を受けます:

Forward

パスの順方向に必ず揃えられるオブジェクトの軸(すなわち、オーナー位置における曲線の接線)。

Up

worldのZ軸に(できるだけ) 整列される必要があるオブジェクトの軸。

実際に、このオプションを有効にすると、オーナーの振る舞いは該当するプロパティにおいて [Locked Track constraint](#)が引き起こすものと

共通するものになり、pathを“芯軸”として、worldのZ軸が“磁石”となります。

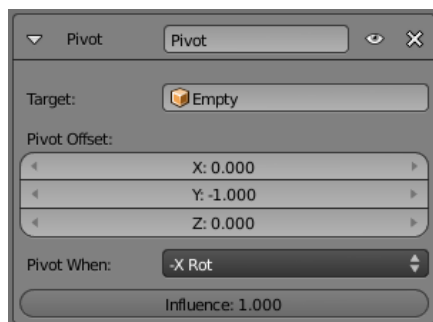
Pivot Constraint

概要

Pivot 拘束は、オーナーがターゲットオブジェクトの周りを回転するようにできます。

もともとは足のリグを対象としたものでした。

オプション



Pivot panel

Target

ピボットポイントとして使用されるオブジェクト

Bone

Target がアーマチュアの場合は、ボーンのための新しいフィールドが表示されます。

Head/Tail

ボーンをターゲットとして使用するとき、このボーンに沿ってどこに目標点を置くのを選ぶことができます。

Vertex Group

Target がメッシュである場合、新しいフィールドに、どの頂点グループが選択できるかが表示されます。

Pivot Offset

ターゲットからのピボットのオフセット

Pivot When

Always, Z Rot, Y Rot, ...

Example

[\[video link\]](#)

See also

- [Blender Artists Forum: Head-Tail pivot Constrain proposal \(with Video and .blend\)](#), the thread where the constraint was first proposed

Rigid Body Joint Constraint

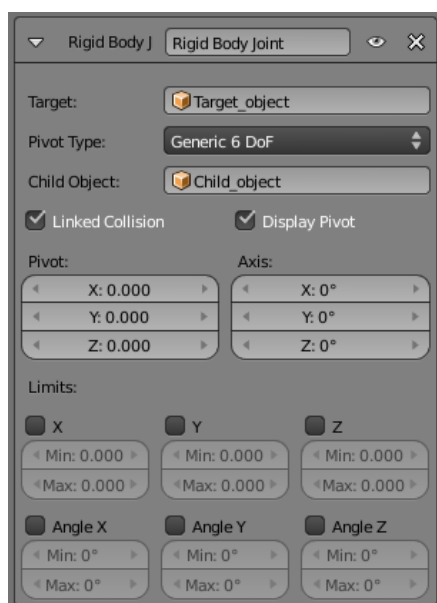
Description

Rigid Body Joint 拘束は非常に特殊です。基本的に、そのオーナーとそのターゲット間の接合をシミュレートするために、Blender Game Engineの物理的な部分で使用されます。次の3つのジョイントタイプが用意されています: hinge type, ball-and-socket type, or generic six-DoF (degrees of freedom) type.

関節点と軸は、オーナーに対して定義され固定されます。ターゲットの移動は、棒の中心点にくっついたようになり、棒のもう一方の端は関節点/ピボット点まわりに回転します...

この拘束は、最も“標準的”な静止画やアニメーションのプロジェクトでは役に立ちません。ただし、BGEの外でその結果を使用することはでき、Game » Record Animation メニューエントリを通じて行えます (User Preferences ウィンドウのメインメニューから。このトピックに関する詳細情報については [Rigid Bodies](#) 参照)。

オプション



Rigid Body Joint panel

Target

この拘束は、一つのターゲットを使用し、それを持っていないときは機能しません (赤の状態)。

Joint Type

Ball

典型的なボールソケット ジョイントのように動作する、すなわち、肩関節のようにすべての軸の周りに回転することができます。

Hinge

一つの平面で動作する、肘のように: オーナーとターゲットは、ピボット (joint point)のX軸まわりにだけ回転させることができます。

Limits

X軸の角度制限

Cone Twist

Ballに似ていますが、これはポイントツーポイントの関節で、円錐とねじれ軸への制限が追加されています

Limits

角度制限

Generic 6DOF

Ball オプションとして動作しますが、ターゲットはもはや、ピボットポイントから一定の距離には拘束されません、デフォルトでは (hence the six degrees of freedom: rotation and translation around/along the three axes).

実際には、デフォルトではもはやジョイントではなく、このオプションを使用すると、ただし追加の設定を有効にしてこれらの自由度のいくつかを制限することはできません:

Limits

Blenderの単位と角度で指定された軸 (of the pivot) における、それぞれの線形および角度の制限。

Child Object

通常は、空白のままにします。右クリックしてReset to Default Valueを選択すると、デフォルト値にリセットして空白にすることができます。

Linked Collision

有効にすると、オーナーとターゲットとの間の衝突検出を無効にします (BGEの物理エンジンで)。

Display Pivot

有効にすると、3Dビューで、ジョイントのピボットを描画します。最も役立つのは、特に Generic 6DOF 関節タイプをつかうときです!

Pivot

これらの3つの数値フィールドをつかうと、*オーナーの空間*で、ピボットポイントを再配置することができます。

Axis

これらの3つの数値フィールドをつかうと、*オーナーの空間*で、ピボットポイントを回転させることができます。

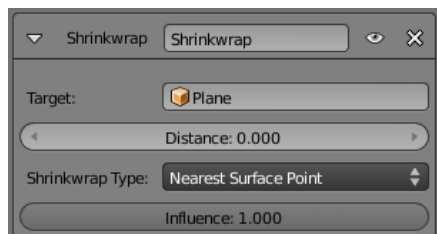
Shrinkwrap Constraint

概要

Shrinkwrap 拘束は、[Shrinkwrap modifier](#)の“オブジェクト版”です。そのオーナーの位置を、ターゲットの表面に縮めます。

これは、ターゲットが表面を持っている必要があることを意味します。実際には、拘束はさらに相手を選び、ターゲットとしてメッシュしか使用することができません。したがって、*Shrinkwrap* オプションは、*Add Constraint to Active Object* メニューにしか表示されず、CtrlAltC、(またはそのボーンに相当)、選択された非アクティブなオブジェクトがメッシュであるときだけのものです。

オプション



Shrinkwrap panel

Target

この拘束は1つのターゲットを使用し、これはメッシュオブジェクトでなければならず、それを持っていないときは機能しません(赤の状態)。

Distance

この数値フィールドは、ターゲットの表面に縮められる計算位置からのオーナーのオフセットを制御します。

正の値は、ターゲットの“外側”にオーナーが配置され、負の値はターゲットの“内側”に配置されます。

このオフセットは、オーナーの元の(すなわち、拘束される前の)位置で定義された直線に沿って適用され、そしてターゲット表面上の位置が計算されます。

Shrinkwrap Type

このドロップダウンリストでは、オーナーの中心を解釈するためのターゲット表面上の点を、どの方法を使って計算するのかを選択することができます。3つのオプションがあります:

Nearest Surface Point

選択したターゲット表面のポイントは、元のオーナーの位置から最短距離のものになります。これがデフォルトで、一般的に最も便利なオプションです。

Projection

ターゲットの表面のポイントを、与えられた軸に沿ってオーナーの中心を投影することで決定します。

この軸は、このタイプを選択したときに現れる X, Y and Z の3つのトグルボタンによって制御されます。これが意味するのは、投影軸の整列は、グローバル軸のいずれか、それら2つ中点 (XY, XZ or YZ)、または3つの中点 (XYZ) を使ってのみ行われるということです。

選択した方向に沿ったオーナーの中心の投影がターゲットの表面にヒットしない場合、オーナーの位置は変更されません。

Nearest Vertex

このメソッドは *Nearest Surface Point* と非常によく似ていて、ただしオーナーの収縮可能な位置がターゲットの頂点に限定されるところが違います。

アニメーション

アニメーションでは、時の経過にそって、オブジェクトを移動させたり形を変えたりしていきます。オブジェクトは、さまざまな方法でアニメーション化できます。1つかそれ以上のオブジェクトを同時にアニメーション化できる方法もあります:

オブジェクト全体としての移動

位置、向き、大きさを一度に変更する。

オブジェクトを変形させる

オブジェクトの頂点またはコントロールポイントをアニメーション化。

アーマチュアを介したキャラクタアニメーション

メッシュ内のボーンの動きによる変形でアニメーションにし、非常に複雑で柔軟な相互作用によって、キャラクターの形をしたオブジェクトの歩行やジャンプを表現します。

この章では最初の2つをカバーしますが、ここで提示する基本は、実際に以下の章を理解するためにも同様に不可欠です。

3つのメソッドは通常、3Dオブジェクトを動かすアニメーションソフトウェアで使用されます:

キーフレーム

全ての位置は、時間の単位(フレーム)ごとに保存されます。アニメーションは、流動的にフレームのオブジェクトを補間することで作成されます。この方法の利点は、はっきりと可視化された単位で作業できることです。アニメーターは、ひとつのポジションから次へというように作業することも、以前に作成したポジションを変更することも、それらをいちどに移動させたりもできます。

アニメーションカーブ

曲線はキーフレームから補間され、位置、回転、およびサイズについてのXYZそれぞれの成分だけでなく、Blender内のその他の属性についても描画することができます。これらのフォームの動きのグラフは、水平方向には時間が設定され、垂直方向には値が設定されています。この方法の利点は、運動の結果を正確に制御できることです。

パス

曲線は3D空間に描画され、オブジェクトはパスに沿う位置の時間関数に従うように拘束されます。

Blenderで最初は二つのシステムだったものは完全に統合されて一つになり、[F-Curve system](#)となりました。

ブレンダー2.5xで、私でもアニメーションのすべてをできるようにになりました。以前は、唯一確かなデータブロックはkeframedする能力でした。今ではユーザーが複数の値に変更することができるデータのほぼすべての種類をアニメーション化する機能があります。

Chapters

一般的な原則とツール

[Keyframes](#)
[Animation Editors](#)
[Using The Timeline](#)
[Markers](#)

グラフエディタ

[Using the Graph Editor](#)
[F-Curves](#)
[Editing Curves](#)
[F-Curve Modifiers](#)

アクションエディタ

[Actions](#)
[Creating Actions](#)

アニメーションのテクニック

[Constraints](#)
[Moving objects on a Path](#)
[Changing Object Layers](#)
[Game Engine Physics Recording](#)

変形によるアニメーション

[Methods of deformation](#)
[Shape Keys](#)
[Absolute Shape Keys](#)
[Deforming by a Lattice](#)
[Deforming with Hooks](#)

See also [Hooks](#) - Uses a modifier as a way to change the shape of a mesh. Sorta like sticking a fish hook in a mesh and pulling. Uses the principles discussed in Shape Keys.

Drivers

[Drivers](#)
[Driven Shape Keys](#)

The [BSoD Introduction to Character Animation tutorial](#) is a good starting point for learning character animation. Even if you never used Blender before.

キーフレーム

キーフレームはアニメーションの基本となるものです。キーフレームは特定のフレームにおけるデータの値を定義します。

これは明らかなことですが、何かがアニメートされているとき、それは時間に応じて変化しているわけです。Blenderでは、オブジェクトをアニメートすることは、ある時間の範囲にわたってそのプロパティを変化させることです。たとえばX方向の位置や、マテリアルの拡散色のRed値等です。

以前触れたように、Blenderにおける基本的な時間の単位は「フレーム」です。1フレームの時間的長さはフレームレートによって決まり、通常、非常に短い時間です。

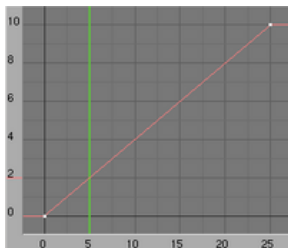
アニメーションは複数のフレームにわたる少しずつの変化によってつくられるので、通常は変化するプロパティを1フレームごとに手作業で調整したりはしません。その理由は、

- そんなことをしていたら何年もかかる！
- 滑らかな変化をさせるのが難しい(フレームごとに数値を正確に計算するなら別ですが、それは大変すぎます)。

これが、ほとんどのアニメーションが補間(interpolation)を使って作られる理由です。

概念としては簡単です。いくつかの「制御点」、キーフレームと呼ばれるものを定義します。これはフレームの間隔をあけて設定します。

キーフレーム同士の間フレームについては、プロパティの数値はBlenderによって計算(補間)されて埋められます。これにより、製作者の負担は大幅に減ります。このカーブはグラフエディタで編集できます。



Example of interpolation

たとえば、以下のような状態のとき、

- フレーム0では数値が0。
- フレーム25では数値が10。
- 線形補間である。

この場合、フレーム5における数値は2になります。

このようなことが途中のフレームのすべてで行われます。最初と最後の二つの点を設定すれば、25フレームにわたって数値0から10まで滑らかに変化します。そしてフレーム15で数値が9になるようにしたければ、あたらな制御点(キーフレーム)を追加すればいいのです。

キーフレームの作成

3Dビューでの作成

3Dビューでキーフレームをセットできます。以下のようなメニューが現れます。

- Location
- Rotation
- Scale
- LocRot
- LocScale
- LocRotScale
- RotScale
- Visual Location
- Visual Rotation
- Visual Scale
- Delta Location
- Delta Rotation
- Delta Scale

プロパティパネルでの作成

変更可能な属性はほぼすべてキーフレームとして設定できます。以下のどちらかの操作をしてください。

- 属性の項目にポインタを重ね、を押す。するとその項目は黄色く変化し、キーフレームが追加されたことが示されます。
- その項目上でRMBをクリックし、Insert Keyframes、ベクトルに対してはInsert Single Keyframeを選ぶ。

アニメーション用エディタでの作成

すでに何らかのキーフレームを持っているチャンネルに対しては、いくつかのアニメーション用エディタのウィンドウ上で新たなキーフレームを追加できます。それぞれのページを参照。

- [タイムライン](#)

- [グラフエディタ](#)
- [ドープシート](#)

キーフレームの編集

キーフレームを編集する方法はいくつかあります。上記のページで説明します。

Actions

(アクション)

Actions

(アクション)

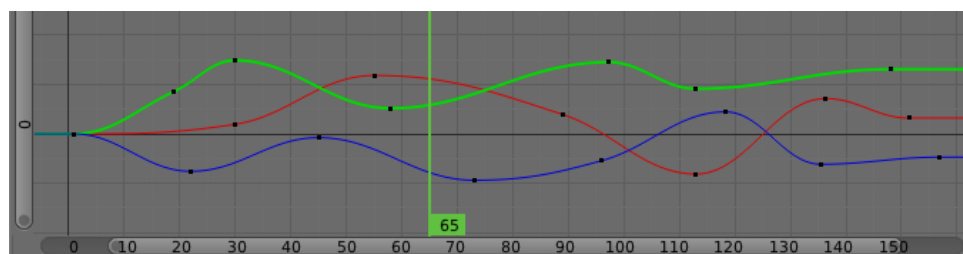
blender でオブジェクトやプロパティをアニメーションさせると、アクションがデータを記録し、収納します。

Actions are added to the Scene or Object.	
Action	Cube CubeAction
Group	Location X Location
Channel	F-Curve Keyframe 0 Keyframe 1 Keyframe 2
F-Curve	Y Location F-Curve
Keyframe	Keyframe 0 Keyframe 1 Keyframe 2

アクション

ですからキーフレームを使ってオブジェクトの位置を変えるアニメーションさせると、このアニメーションはアクションに保存されます。

各プロパティは記録用チャンネルをひとつ持ちます。たとえば Cube.location.x は Channel X Location に記録されます。



グラフエディタ。各チャンネルには、キーフレーム間を結ぶ線で表される、F-カーブがあります。

アクション (Actions)

アニメーションのデータを記録し、収容します。

グループ (Groups)

チャンネルの集まりです。

チャンネル (Channels)

プロパティを記録します。

F-カーブ (F-Curves)

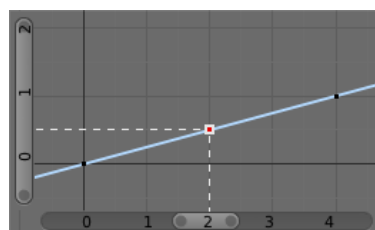
キーフレーム間の差を補間するために使われます。

キーフレーム (Keyframes)

プロパティの値を設定するのに使われます。

F-Curve Interpolation

(F-カーブ補間)



グラフエディタ: チャンネル F-カーブ

キーフレームは、ユーザーにより値をセットされています。

F-カーブはキーフレーム間の差異を補間するために使われます。

F-カーブには、さまざまな補間方式と [F-カーブモディファイア\(F-Curve Modifiers\)](#) があります。

[F-カーブ\(F-Curve\)](#) の設定内容のほとんどは [グラフエディタ\(Graph Editor\)](#) で見つかります。

Basic Animation

(基本的なアニメーション)

以下はオブジェクトをアニメートさせる、一般的な方法です。

この方法は、ポーズモードでのアーマチュアのボーンなど、別のオブジェクトに使うことができます。

Insert Keyframes

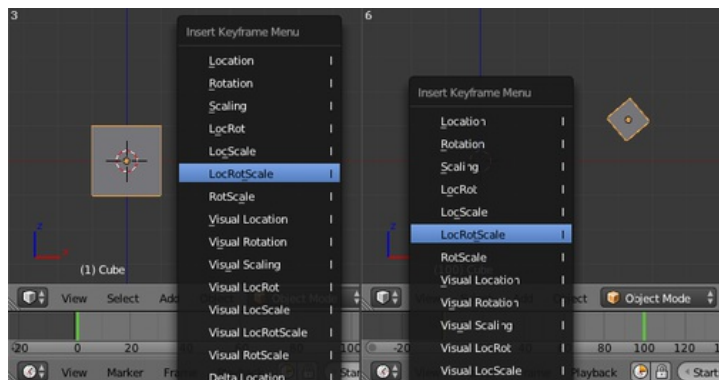
(キーフレームの挿入)

この例では立方体の位置、回転、拡大縮小をアニメートする方法をお見せします。

1. はじめに、タイムライン(Timeline) か他のアニメーションエディタ内で、フレームを 1 に合わせます。
2. 立方体を オブジェクトモードで選択した状態で、3D ビュー内で I を押します。
3. キーフレーム挿入 (Insert Keyframe) メニューから 位置回転拡大縮小 (LocRotScale) を選びます。

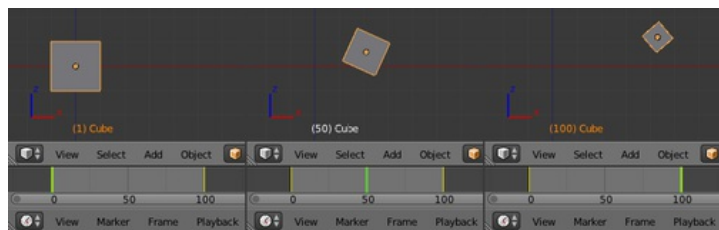
これでフレーム 1 の 立方体の位置、回転、拡大縮小が記録されます。

4. フレームを 100 に合わせます。
5. つかむ/移動 (Grab/Move) G、回転 (Rotate) R、拡大縮小 (Scale) S により、立方体をトランスフォームします。
6. 3D ビューで I を押します。キーフレーム挿入 (Insert Keyframe Menu) メニューから 位置回転拡大縮小 (LocRotScale) を選びます。



3. 6. キーフレームの挿入

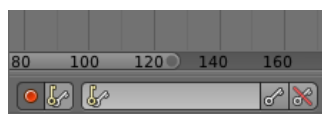
アニメーションを試すには AltA を押して再生します。



フレーム 1、50、100 でのアニメーション

Auto Keyframe

(自動キーフレーム)



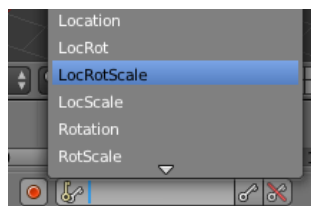
タイムラインの自動キーフレーム

自動キーフレームは、タイムラインヘッダーにある赤い記録ボタンです。自動キーフレームは、トランスフォーム型のプロパティの値が変化したときにだけ、自動的に所定のフレームにキーフレームを追加します。

詳しくは [タイムライン](#) の「ヘッダ (IV) Automation」をご覧ください。

Keying Sets

(キーイングセット)



タイムライン キーイングセット

キーイングセット (Keying Sets) はキーフレームチャンネルの集合で、複数のプロパティを同時に記録するために使われます。'LocRotScale' のような組み込みのキーイングセットがいくつかあり、独自にキーイングセットを作ることができます。

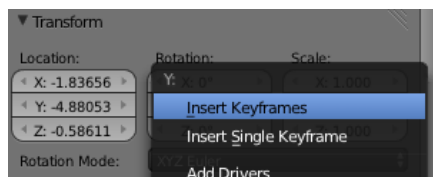
キーイングセットを使うには、はじめに **タイムラインヘッダー**か **'キーイングセットパネル'** からキーイングセットを選びます。

これで、3DビューでIを押したときに、blender はアクティブなキーイングセット内のすべてのプロパティに対して、キーフレームを追加します。

詳しくは [キーイングセット \(eng\)](#) をご覧ください。

Properties

(プロパティ)



キーフレームのプロパティ

キーフレームは blender の色々な種類のプロパティをアニメートさせるのに使えます。

あるプロパティに UI からキーフレームを追加するには、プロパティを RMB して、次に「ひとつのキーフレームを挿入 (Insert Single Keyframe)」または「キーフレームを挿入 (Insert Keyframes)」を選びます。

「キーフレームを挿入 (Insert Keyframes)」はプロパティの集合に対してひとつのキーフレームを追加します。



プロパティ、ドライバ、キーフレーム

プロパティは、その状態によって違った色とメニューアイテムを持ちます (訳注: メニューにある「単一 (Single)」とは、XYZのような組になったプロパティについて、Xだけ、Yだけ、Zだけを対象にするものです)。

灰色 - プロパティはキーフレームやドライバでアニメートされていません。

- キーフレームを挿入 (Insert Keyframes) I
- 単一のキーフレームを挿入 (Insert Single Keyframe)
- ドライバを追加 (Add Drivers)
- 単一のドライバを追加 (Add Single Driver)
- ドライバをペースト (Paste Driver)

紫色 - プロパティの値はドライバによりコントロールされます。

- ドライバを削除 (Delete Drivers)
- 単一のドライバを削除 (Delete Single Driver)
- ドライバをコピー (Copy Driver)
- ドライバをペースト (Paste Driver)

緑色 - プロパティにはキーフレームのあるチャンネルがあります。

- キーフレームを挿入 (Insert Keyframes) I
- 単一のキーフレームを挿入 (Insert Single Keyframe)
- キーフレームを消去 (Clear Keyframes) Alt+ Shift
- 単一のキーフレームを消去 (Clear Single Keyframes)

黄色 - 現在のフレーム上で、プロパティがキーフレームを持ちます。

- キーフレームを置き換え (Replace Keyframes) I
- 単一のキーフレームを置き換え (Replace Single Keyframe)
- キーフレームを削除 (Delete Keyframes) AltI
- 単一のキーフレームを削除 (Delete Single Keyframe)
- キーフレームをクリア (Clear Keyframes) Alt+ ShiftI
- 単一のキーフレームをクリア (Clear Single Keyframes)

各プロパティにはいくつか、キーイングセットオプションがあります。

- キーイングセットにすべて追加 (Add All to Keying Set) K
- キーイングセットに単一で追加 (Add Single to Keying Set)
- キーイングセットから除去 (Remove from Keying Set)

Editing

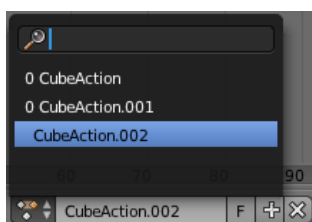
(編集)

3Dビューで:

現在のフレームにキーフレームを挿入 (Insert Keyframes) I
現在のフレームでキーフレームを削除 (Delete Keyframes) AltI

Working with Actions

(アクションでの作業)



アクションブラウザ

キーフレームを追加して最初にオブジェクトをアニメートさせると、blender はデータ記録のため *アクション* を作ります。

アクションの管理には [ドープシート\(DopeSheet\)](#) のアクションエディタヘッダーにある *アクションブラウザ(Action Browser)* を使うか、[NLA Editor](#) のプロパティ領域を使います。

同一オブジェクトに対して複数のアクションを作るつもりなら、各アクションに対して **F** ボタンを押してください。この操作でアクションに *Fake User* が与えられ、blender はリンクしていないアクションを保存します。

オブジェクトが同時に編集に使える *アクション* はひとつだけです。複数のアクションを混ぜるには、[NLAエディタ](#) を使います。

ドライバ

ドライバは、あるプロパティのアニメーションを他のプロパティの値によってコントロールするために使います。つまり、そのアニメートされる値はフレーム番号でコントロールされるのではなく、特定のチャンネルにコントロールされます。ドライバが使えるのは、何らかの単一のプロパティや、回転の差や、Blender上でも編集できるPythonスクリプト等です。


たとえば、ドライバのX位置を使ってマテリアルの色 (RGBカーブ) をコントロールしたり、ドライバの回転を使ってオブジェクトの拡大縮小をコントロールしたり、ドライバの拡大縮小を使ってメッシュやカーブの形状を (シェイプキーを介して) コントロールしたり、Pythonの関数でコンストレイントの影響をコントロールしたり... もっともっとうろんなことができます。

一つの効果的な使い方は、キャラクターアニメーションでの使用です。たとえば顔のシェイプキーに対してドライバを追加し、そのドライバオブジェクト (もちろんこれはレンダリングされないように設定します) を動かしてキャラクターの表情を操作することができます。NLAエディタと少し似ていますが、よりわかりやすいでしょう。[シェイプキーをドライブする](#)を参照。

ドライバのインターフェースや機能はF-カーブに似ていますが、ドライバはアニメーションシステムのヒエラルキーにおいてアクションの外側に位置しています。ドライバはアクションの後で評価されるので、アクションに関わるどんなF-カーブもドライバの効果に上書きされます。


マッピング

ドライバで使われるカーブは時間に支配されているわけではありません (これは力説しておきます。非常に重要なことなのでよく理解してください)。ドライバのプロパティの変更 (X軸上) に応じて、カーブのどの位置の値 (Y軸上) が適用されるかが決まります。

デフォルトでは、入力と出力はGenerator F-カーブモディファイアによって一対一にマッピングされています。このモディファイアを削除して、グラフエリアにCtrl + LMB  でキーフレームを追加すれば、好きなマッピングを作れます。

ドライバの作成

ドライバを作成するには、まずドライブする属性をプロパティパネルで見つけます。これが、他の属性によってドライブされる属性となります。

チャンネル上でRMB  フリックして、いかのオプションの一つを選択してください。

Add Drivers

選択したチャンネルに関連するチャンネルすべてにドライバが作られます。たとえば、X、Y、Z方向の移動に対してそれぞれドライバが作られます。

Add Single Driver

選択したチャンネルのみに対してドライバが作られます。

ドライバの編集

ドライバを作成したらそのチャンネルが何によってどのようにドライブされるかを決めなくてはなりません。

ドライバエディタはグラフエディタの中にあります。ヘッダのメニューで、F-カーブエディタビューからドライバビューに切り替えられます。

ドライバエディタは現在のシーン内にあるすべてのドライバを表示します。左側のチャンネル欄に表示されます。

シーンの中に複数のドライバがある場合、グラフエリアでの表示 / 非表示を切り替えられます。また、特定のドライバを編集するにはそれを選択する必要があります。

ドライバプロパティ

ドライバのプロパティは右側のプロパティパネル (デフォルトでは隠されています。Nで表示されます) にあります。

Update Dependencies

ドライバの接続を更新します。

ドライバの種類

ドライバにはいくつかの種類があります。ドライバの入力値がどのように計算されるかが異なります。

Maximum Value

ユーザ変数の最高値をとります。

Minimum Value

ユーザ変数の最低値をとります。

Sum Values

ユーザ変数の合計値をとります。

Averaged Values

ユーザ変数の平均値をとります。

Scripted Expression

Pythonによる記述にしたがって値を計算します。これはユーザ変数を自在に使える方法でもあります (後述)。

変数

ユーザ変数はドライバの入力値を計算するものです。スクリプトを使う場合以外は、少なくとも一つの変数が必要です。一つの変数を使うと、スクリプト以外のすべての種類のドライバは同じ値を生み出します。

単一プロパティ

シーン内のあらゆるものの中からデータブロックを呼び出すにはSingle Propertyを使います。これはランプの強度でもいいし、テクスチャのカラーでもいいです。

座標変換チャンネル

このタイプはオブジェクトの座標変換プロパティを通じてカーブをコントロールすることができます。つまり、一つのドライバで9つのカーブを別々にコントロールできます(XYZそれぞれの位置、回転、拡大縮小)。

オブジェクトドライバを設定するには、

- ドライブしたいチャンネルを選択する。
- OB欄が現れるので、ドライバオブジェクトの名前を入力する。
- ドライバがアーマチュア以外の場合、Driver typeのドロップダウンリストからはObjectのみが選べる。
- 最後に、二つ目のドロップダウンリストでカーブをドライブするためのプロパティを選択する(Loc X、Scale Zなど)。

これでカーブは時間にコントロールされるのではなく、選んだプロパティによってコントロールされます。値はBlender単位です。

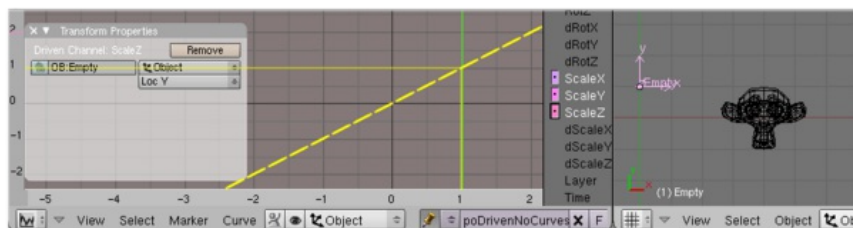
ドライバのプロパティはローカル座標で評価されることに注意してください(親オブジェクトの座標変換には影響されません)。

例

これからの例では、よく知られた猿の頭(Susanne)のサイズを、Emptyをドライバとしてコントロールしてみます。まず、SusanneオブジェクトのScaleX、ScaleY、ScaleZチャンネルにAdd Driverでドライバを追加します(カーブがまだ存在しない場合は、自動的に作られます)。今の時点では、まだカーブがないので、Blenderは直線的なカーブがあるかのように一対一のマッピングをします(下の画像での破線)。これはまた同じドライバのプロパティ(この例ではY位置)をいくつかの違ったドライバに使えることを示しています。

Object drivers.

下図ではカーブエディタウィンドウはSuzanneのカーブに表示固定されているので、Emptyオブジェクトを移動しながらドライバカーブを見ることができます)。



The empty driver at Y = 1.0 gives Suzanne a scale of 1.0...



With driver at Y = 2.0, driver's scale at 2.0...



And driver at Y = -2.0, driver's scale at... Oh, I think you got it!

では、ドライバのカーブを基本的な一対一のマッピングより複雑にしてみましょう。下図に示されているのは上と同じシーンですが、今回はScaleYチャンネルが1.0から-1.0の間で反復する擬似サインカーブになっています。これは三つの制御点とCyclic補外で作られています。SuzanneのX方向とZ方向の大きさは依然としてEmptyオブジェクトのY位置を直接的に反映していますが、Y方向の大きさは下図のようにEmptyオブジェクトの移動とともに反復的に変わるということです。

Object driver with curve.



The empty driver at Y = 1.0 gives Suzanne a scale of (1.0, 1.0, 1.0)...



With driver at Y = 1.25, driven's scale at (1.25, 0.0, 1.25)...



And driver at Y = -2.0, driven's scale at (-2.0, 1.0, -2.0)!

回転の差

回転の差はボーンの場合だけ使えますが、オブジェクトの場合とほぼ同様です。

ボーンをドライバとして使うには、OB欄にアーマチュアの名前を入力し、Driver typeのリストからPoseを選択します。そして現れたBO欄にボーンの名前を入力し、使用するプロパティを選択します。

以下の特徴に注意してください。

- 座標変換はそのボーンの空間で評価されます(親ボーンやオブジェクトそのものの座標には影響されません)。
- 回転をドライバに使う場合、制限に気づくでしょう。ボーンのリニア回転は常に $[-180^\circ, 180^\circ]$ の範囲にマッピングされます(つまり、 180° を超えて回転させると、 -180° に値が飛び移ります)。
- 特別なドライバプロパティが使えます。{Literal|Rotation Difference}は他のボーンの名前を入力すると、そのボーンとドライバボーンとの角度の差をドライバに使用します。

距離

距離モードは二つのオブジェクトまたはボーン間の距離を使います。

式ドライバ

Python式ドライバ(略してパイドライバ)は、他のオブジェクトのプロパティを使う代わりに、一行のPythonによる式を入力に使用します。明確な値が計算できるならどんな記号でも使えます。



パイドライバによってコントロールするのは、カーブそのものではありません。他のタイプのドライバと同じように、カーブ上のどこの値を使うかということをコントロールします。

パイドライバを使うには、ドライバタイプをScripted Expressionに変更します。すると標準的なテキスト欄が現れるので、そこにPython式を入力します。

このドライバには面白い可能性がたくさんあります。数式やプログラミングを使ってオブジェクトをアニメートすることができるのです。唯一のルールは、パイドライバの表現は最終的には実数値を生み出さなければいけないということです。たとえば文字列などではいけません。

オブジェクトが特定の点の周りで往復するようにサインまたはコサイン関数を使うこともできるし、拡散、RGB、アルファなどをランダムな数値で変えることもできます。

パイドライバの評価は、内蔵のeval()関数の結果と同じです。ただしパイドライバでは結果として実数を返さなければならないという制約があります(あらゆる種類の数値は自動的に浮動小数点数にキャストされます)。

Valid Expressions

(訳注: Blender 2.6現在ではAPが大幅に変更されているので、以下の式は動かないものが多いです。最新の情報を参照してください)

We've already told the basics: there is a text field where you can type an expression in Python. Here are some examples of valid expressions:

- Any real value:

1. 1.0

- Expressions with numbers and operators:

1. $4.5 + 8.9 * 7.0 - (2 / 3.0)$

- Expressions also with variables:

1. `math.pi * 2 + 5.0`

- Available data:

1. `Blender.Get("curframe")` # the current animation frame

- A little math:

1. `math.sin(Blender.Get("curframe"))` # the sine of the current frame

- Using the animated (driven) object, available as `self`:

1. `self.LocX * 10`

Builtin resources and aliases

Pydrivers use their own global dictionary that is cached and only gets recreated when any expression fails. In this dictionary, a few modules are pre-imported, so that they can be used inside pydriver expressions.

Note

To save typing and keep expressions smaller, we've added aliases for each module: Blender can be referenced as "Blender" or simply as "b". Below, each module is followed by its available aliases.

- All from `builtin` (the default `builtin` module)
- `Blender`: `blender`, `b`
- `Blender.Noise`: `noise`, `n`
- `math`: `math`, `m`

Example expression:

1. `m.cos(m.pi * b.Get("curframe") / n.random())`

Aliases were also added for a few commonly needed data:

- `ob(name)` is equivalent to `Blender.Object.Get(name)`
- `me(name)` is equivalent to `Blender.Mesh.Get(name)`
- `ma(name)` is equivalent to `Blender.Material.Get(name)`

Example expression:

1. `ob("Cube").LocX + ob("Lamp").RotZ`

The pydrivers.py Blender text

Besides the above modules, if there's a Blender text buffer called "pydrivers.py" loaded in the Text Editor, it's also imported:

- `pydrivers`: `pydrivers`, `p`

This allows users to create their own functions and add their own variables without the restriction of the one-line Python expression. For example, if your `pydrivers.py` text looks like this:

1. `myvar = 10.0`
- 2.
3. `def myfunction(arg):`
4. `# do something fancy here`
5. `return float_val`

You can access both `myvar` and `myfunction` inside any expression:

1. `p.myvar * p.myfunction(2) - 1`

Note: if you make updates to the `pydrivers.py` text, go to the Curve Editor window and click in any pydriver's text input box (in the Transform Properties panel), then click out of it or press ↵ Enter, to force reloading the `pydrivers.py` module and to update all pydrivers at once.

Example

Let's say we want to give our Suzanne a perfect two BU radius circle movement in the (XY) plane. This is not easy to achieve with standard curves, as you have to create perfect sinusoidal ones (for those who lost their math, a perfect circle in a Cartesian plane is obtained with $x = \sin(t)$ and $y = \cos(t)$...). So let's use Python drivers:

The `sin()` and `cos()` math functions take as input a *radian* value (i.e. a cycle takes 2π frames to complete), and output a value in range $[-1.0, 1.0]$. So, to get a **2BU** radius circle fully walked in **50** frames (two seconds in PAL videos), we have to use the following Python expression:

- For LocX:

1. `2.0 * math.sin(Blender.Get("curframe")*math.pi/25)`

- For LocY:

```
1. 2.0 * math.cos(Blender.Get("curframe")*math.pi/25)
```

Now, Suzanne will go in endless circles around the world's origin (if you want to make it turn around another point, just add an offset to the above expressions, or drive dLoc... curves...).

Note that:

- In the above example, we did not use F-curves.
- You can also get a perfect circular movement by using a [NURBS circle curve](#) as a [path](#)...

Links

- The [Blender 2.49 Python API reference](#) and the chapter on [extending Blender with Python](#).
- [Python](#) and its [documentation](#).
- This might be a good hunting ground for those looking for functions to try with pydrivers: <http://functions.wolfram.com/> (newcomers are recommended to start with elementary ones, specially trigonometric).
- Finally, the patch tracker entry, with patch and sample .blend files for pydrivers: [right here](#).

マーカー

マーカーは何か重要なことが起こるフレームに印をつけるために使います。キャラクターのアニメーションが始まるフレーム、カメラの位置が変わるフレーム、ドアが開くフレーム、などです。マーカーには一目で意味が分かるように名前を付けられます。マーカーはBlenderの様々なウィンドウで使えます。キーフレームと違ってマーカーは常に整数番号のフレームに付けられます。たとえば、フレーム**2.5**にマーカーをセットすることはできません。

マーカーは以下のすべてのエディタ(およびその中のあらゆるモード)で使えます。

- [グラフィックエディタウィンドウ](#)
- [アクションエディタウィンドウ](#)
- [ドープシート](#)
- [NLAエディタウィンドウ](#)
- [ビデオシーケンスエディタウィンドウ](#)
- [タイムラインウィンドウ](#)

上記のウィンドウのどれかでマーカーが作成されると、マーカーをサポートしているすべてのウィンドウにそれが現れます。上記に加えて:

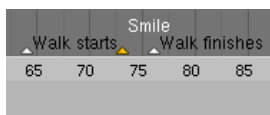
- [3Dビューウィンドウ](#).

ポーズマーカー

別の種類の、「ポーズマーカー」と呼ばれるものもあります。これはアーマチュアとアクションエディタウィンドウに特有のもので、ポーズライブラリに関連します。詳しくは[ここ\(eng\)](#)を参照。

視覚化

標準

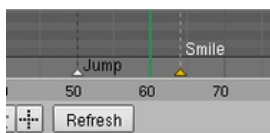


Markers: small but useful.

ほとんどのウィンドウはマーカーを同じ方法で視覚化します。下部に表示される小さな三角形です。白いものは選択されていないもの、黄色いものは選択されているものです。

名前が付けられている場合は、マーカーの右側に表示されます。選択されている場合は白い文字になります。図"Markers: small but useful"を見てください。

シーケンサ



Markers in the Sequencer

*ビデオシーケンスエディタ*では個々のマーカーの位置に点線が描かれます(選択されていないものはグレー、選択されているものは白)。

3Dビュー



Marker in a 3D View.

*3Dビュー*ではマーカーの追加/削除をすることはできませんが、そのフレームに存在しているマーカーの名前が<>に囲まれて左下に表示されます。図"Marker in a 3D view"を見てください。

マーカーの作成と編集

残念ながら、マーカーをサポートしているすべてのウィンドウに共通したショートカットやメニューはありません。よって、以下の囲みには最も多い操作法を載せ、例外はカッコの中に書いてあります。

マーカーの作成

Mode: all modes

Hotkey: M (CtrlAltM in a VSE)

Menu: Marker » Add Marker (Frame » Add Marker in a timeline)



マーカーを追加する最も単純な方法は、目的のフレームに移動し、M(ビデオシーケンスエディタではCtrlAltM)を押すことです。



あるいは、AltA (またはタイムラインウィンドウで“playback”ボタン)を押してアニメーションを再生し、タイミングよくM (またはビデオシーケンスエディタではCtrlAltM)を押せばマーカーが追加されます。これは音楽のリズムに合わせてマークするのに便利です。

マーカーの選択

Mode: all modes

Hotkey: RMB , ⇧ Shift RMB , A/CtrlA, B/CtrlB

マーカーの三角形をRMB をクリックすると選択できます。複数のマーカーを選択または選択解除するには⇧ Shift RMB です。

lpoカーブエディタ、アクションエディタ、NLAエディタ、ビデオシーケンスエディタでは、CtrlAですべてのマーカーを選択または選択解除できます。また、CtrlBで矩形選択もできます (LMB で選択、RMB で選択解除)。これらのウィンドウのSelectメニューには関連するオプションが見つかるでしょう。

タイムラインウィンドウとオーディオウィンドウでは、Aですべてのマーカーを選択または選択解除、Bで矩形選択または選択解除できます。

マーカーの命名

Mode: all modes

Hotkey: CtrlM

Menu: Marker » (Re)Name Marker (Frame » Name Marker in a timeline)



何十ものマーカーがシーンの時間の中に散らばっていたら、その意味を覚えていない限り役に立ちません。マーカーを選択し、CtrlM押し、名前をタイプしてOKを押すと名前を付けられます。

マーカーの移動

Mode: all modes

Hotkey: CtrlG (G in a timeline or audio)

Menu: Marker » Grab/Move Marker (Frame » Grab/Move Marker in a timeline)

一つまたは複数のマーカーを選択し、CtrlG (タイムラインウィンドウとオーディオウィンドウではG)を押すと移動でき、LMB または⇐ Enterで決定します (RMB またはEscでキャンセル)。

デフォルトでは、マーカーの移動は1フレーム単位ですが、Ctrlを押しているとき、一秒単位で移動できます。フレームレートが25 fpsだったら、25フレーム単位での移動になります。

マーカーの複製

Mode: all modes

Hotkey: Ctrl⇧ ShiftD (⇧ ShiftD in a timeline or audio)

Menu: Marker » Duplicate Marker (Frame » Duplicate Marker in a timeline)

Ctrl⇧ ShiftD (タイムラインウィンドウとオーディオウィンドウでは⇧ ShiftD)を押すと、選択したマーカーを複製できます。複製をすると自動的に移動モードになるので、目的の位置 (時間) に移動してください。


Blenderにおける他の複製コマンドとは違い、複製されたマーカーは名前が変更されません (".001"などは付きません)。

マーカーの削除

Mode: all modes

Hotkey: ⇧ ShiftX (X in a timeline or audio)

Menu: Marker » Delete Marker (Frame » Delete Marker in a timeline)

選択したマーカーを削除するには⇧ ShiftX (タイムラインウィンドウとオーディオウィンドウではX)を押し、確認メッセージをLMB をクリックします。

Keyframe Visualization

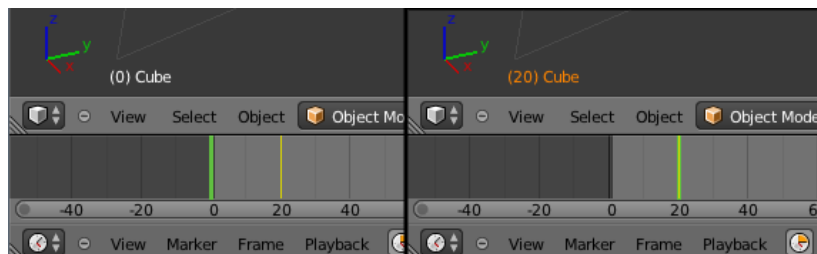
(キーフレームの視覚表現)

3Dビューには、アニメーション制作を助ける重要な視覚化機能がいくつかあります。

Keyframe Visualization

(キーフレームの視覚表現)

現在のフレームが、現在アクティブなオブジェクトのキーフレームであるとき、このオブジェクトの名前が黄色に変わります。オブジェクト名は3Dビューの左下隅に表示されます。



左: 現在のフレームが 0 のとき。右: 現在のフレームが Cube のキーフレームであるとき

Motion Paths

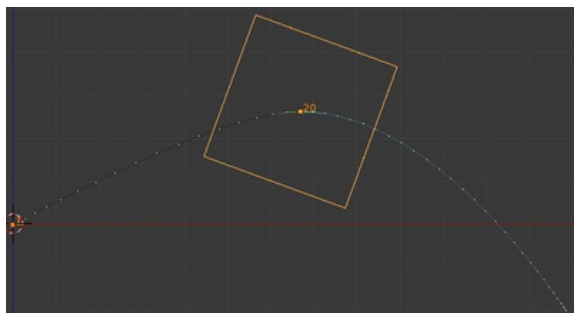
(モーションパス)

モード: オブジェクトモード

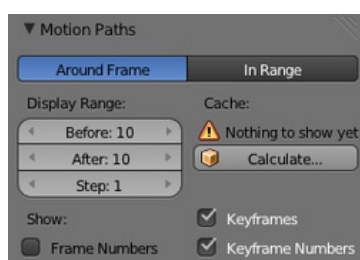
パネル: オブジェクト(Object)

(訳注: 本文中にある「ゴースト(Ghosts)」はアマチュアの機能です。アマチュアのモーションパス、およびゴーストについては、[ポージング視覚表現](#)をご覧ください。)

この機能は、一連のフレームすべてでオブジェクト位置を表示し、アニメーションを視覚化するものです。



アニメーションするキューブのモーションパスを表示したところ



モーションパスパネル

オプションを見る前に、まずはパスの表示／非表示の方法を見てみましょう(オプションはすべて、オブジェクト(Object) パネル内の「モーションパス」にあります)。表示／非表示操作は自分で行う必要があります。はじめに、モーションパスを表示／非表示にするオブジェクトを選択しなければなりません。次に、

- パスを表示するには(もしくはパスの更新が必要ななら)パスの計算(Calculate Path) ボタンをクリックします。
- パスを非表示にするには、その横の×印をクリックします。

忘れてください:これらの操作が影響するのは、選択中のオブジェクトのパスだけです。

パスは黒く描かれます。白い点はフレームを示します。現在のフレーム付近は青く光ります。

Options

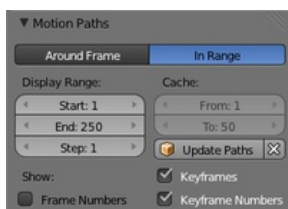
(オプション)



「フレーム周辺 (Around Frame)」設定のモーションパスパネル

フレーム周辺 (Around Frame)

現在のフレームの前後、指定数分のフレームのポーズについて、パスを表示します (ゴーストと同様です)。



「指定範囲 (In Range)」設定のモーションパスパネル

指定範囲 (In Range)

特定範囲内のポーズについて、パスを表示します。

表示範囲 (Display Range)

前/後 (Before/After)

現在のフレームの前後何フレームを表示させるのか指定します (「フレーム周辺 (Around Frame)」用)

開始/終了 (Start/End)

表示/計算させるパス範囲の、開始および終了フレーム (「指定範囲 (In Range)」用)

ステップ (Step)

ゴーストにおける GStep と同種のもので、 n フレームおきにパスを表示します。もっとも役立つのは、フレーム番号を表示しているとき (後述) です。3Dビューが雑然とするのを防ぎます。

フレーム番号 (Frame Numbers)

有効化すると、パス上にある各フレーム点の隣に、小さな数字が現れます。これはもちろん、そのフレームの番号です。

キーフレーム (Keyframes)

有効化すると、モーションパス上にボーンのキーフレームを表す、大きな黄色の四角が描かれます (これは、指定フレームの、キーづけされたボーンのパスだけが黄色い四角になるということです)。

キーフレーム番号 (Keyframe Numbers)

有効化すると、フレーム番号のうちキーフレームであるものに色がつきます。上の項目「キーフレーム」を有効にしないと有効になりません。

キャッシュ (Cache)

先頭 (From)/末尾 (To)

モーションパスを描く範囲の先頭/末尾フレームです。先にモーションパスを削除しないとこの範囲は変更できません。

パスの計算/更新 (Calculate Paths/ Update Paths)

パスが一度も計算されていなければ、パスの計算は新しいモーションパスをキャッシュ上に作ります。ポップアップボックスでは、計算するフレーム範囲を選んでください。

パスがすでに計算されていれば、パスは現在のアニメーションにあわせた形に更新されます。計算済みパスのフレーム範囲を変更するには、パスを削除して再計算する必要があります。

アニメーション用エディタ

アニメーション制作は大掛かりな仕事です。反復的な作業もよくあります。そのため、作業を速め、量を減らすための様々なテクニックが何年にもわたって開発されてきました。

それらの機能を実現するために、Blenderにはいくつかのアニメーション用エディタがあります。

タイムライン

タイムラインはアニメーションデータの編集ができるわけではありませんが、シーンのすべてのキーフレームを表示します。[アニメーションマーカー](#)の編集はすることができます。

グラフエディタ

これはもっとも下層レベルのエディタで、カーブを使って個々のプロパティ/セッティング/チャンネルを操作します。

ドープシート

ドープシート(旧称アクションエディタ)は4つのモードを持っています。それぞれは非常に似ていますが、違った種類のデータをあつかいます。

ドープシートモード

基本的なキーフレームエディタ。シーン内のすべてのオブジェクトの、キーフレームを持ったすべてのチャンネルを表示します。

アクションエディタモード

ドープシートモードに似ていますが、オブジェクトごとのキーフレームが表示され、アクションを作成するのに特化しています。

シェイプキーエディタモード

選択されたオブジェクトのシェイプキーアニメーションを表示します。

グリースペンシルモード

グリースペンシルによるアニメーションが表示されます。

NLAエディタ

この上層レベルエディタの背後にあるアイデアは、ノンリニアビデオ編集から借りています。個々の要素がアクションを表し、それを移動、複製、縮めたり引き伸ばしたり(つまりスピードアップしたりスローダウンしたり)、などができます。キーがメッシュの面のようなものだとすると、「NLAアクション」はそれをカプセル化しているオブジェクトのようなものです。



理解しなければならない重要なことは、アニメーションデータの変化がそれぞれのエディタにどう反映されるかということです。

- キーフレームは、アニメーションカーブを形作ります。
- そのアニメーションカーブは、アクションを生成するのに使われます。
- そのアクションは、NLAで使われます。

タイムライン

タイムラインウィンドウは時計のアイコンが目印で、デフォルトではスクリーンの下部に位置しています。


タイムラインは「エディタ」というよりも、アウトライナーウィンドウのような情報取得のためのウィンドウに多少の編集機能がついたものです。

このウィンドウでシーンのアニメーションの概要を得ることができます。現在の時間（フレームまたは秒）、アクティブなオブジェクトのキーフレーム、開始フレームと終了フレーム、マーカー、などなど…。

このウィンドウにはVTRのようなコントローラーがあり、アニメーションの再生、現在のフレームと再生範囲、キーフレーム間のスクロールができます。

タイムラインの要素

カレントフレーム

カレントフレーム（現在のフレーム）は緑色の太い縦線で表示されています（タイムカーソルと呼ばれます。図"Timeline window"ではフレーム100にあります）。ウィンドウ内の好きなところをLMB  クリックするとタイムカーソルを動かせ、またドラッグすれば前後にアニメーションさせることができます。現在のフレーム番号（または秒数）がクリックしているポイントの近くおよびcurrent frame欄に表示されます。

キーフレーム

アクティブなオブジェクトのキーフレーム（またはアクティブな**キーフレーム**）は色のついた縦線で表示されます。Ipoタイプによって三種類あります。

- **黄**

アクションではないオブジェクトキー（位置、回転など）

- **橙**

マテリアルキー（拡散、光沢、反射など）

- **青**

[アクション](#)キー（アーマチュアのポーズ、オブジェクトや形状のIpo）

注意

タイムラインに表示されないキーもあります（テキストやコンストレイント）。

図"Timeline window"では二つのオブジェクトキーフレーム（フレーム10と150）、一つのマテリアルキーフレーム（フレーム40）が表示されており、アクションキーフレームはありません。

マーカー

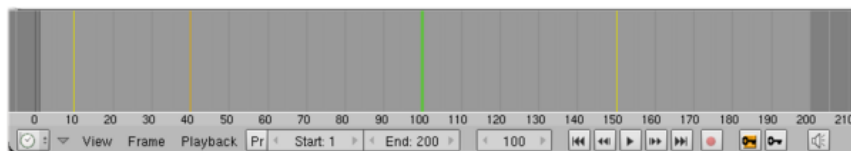
[マーカー](#)マーカーは小さな三角形として視覚化され、そばに名前が表示されます。

色の意味は：

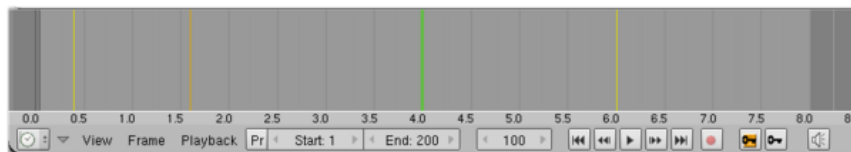
- 白い線, 黒い文字: 選択されていないマーカー
- 黄色い線, 白い文字: 選択されているマーカー

ビューの調整

Timeline window.






Frames display.



Seconds display.

このウィンドウは1次元で、横軸によってシーンにおける時間のみを表現しています。Blenderの基本的な時間の単位はフレームなので、デフォルトではボタンの数値はフレーム数を表しています。

アニメーション範囲は明るいグレーの領域として表されます（図"Timeline window"ではフレーム1から200）。

このウィンドウはBlenderのインターフェースにおける他の領域と同じように動作します。MMB  ドラッグで移動したり（1次元なので左右だけが）、Ctrl MMB  や Wheel  で拡大縮小したりできます。

Viewメニュー

このメニューでは表示するものとしなないもの、またどのように表示するかを選べます。

Maximize Window (CtrlArrowup)

この標準的なコマンドは、このウィンドウをフルスクリーンにします。最大化されるとこの項目は *Tile Window* に変化し、選択すると以前のサイズに戻ります ((CtrlArrowdown))。

Lock Time to Other Windows

これは「時間」関連のウィンドウ(横軸によって時間の流れを表現する種類のウィンドウ)すべてに共通する機能です。このようなウィンドウにはこのオプションがあり、有効にすると常に同じ時間の範囲を表示します。(移動や拡大縮小によって)一つのウィンドウで表示する範囲を変えると、他のウィンドウも直ちに更新されます。

ViewAll (^ Home)

この標準的なコマンドは、開始時間と終了時間によって決められたアニメーション範囲全体が見えるように表示を拡大縮小します。

Center View (C)

この標準的なコマンドは、カレントフレームが中央に来るように表示を移動します。

Jump to Prev Key (CtrlPageDown)

Jump to Next Key (CtrlPageUp)

これらのコマンドは、アクティブなオブジェクトの、直前または直後のキーフレームに移動します。

Jump to Prev Marker (PageDown)

Jump to Next Marker (PageUp)

これらのコマンドは、直前または直後の [マーカー](#) に移動します。

Only Selected Data Keys

私がおもうに、これを有効にすると選択したキーフレームだけが表示されるはずなのですが... 実際は常にアクティブな要素のキーフレームがすべて表示されるようです！

Show Seconds/Literal/Show Frames (T)

デフォルトでは時間の流れはBlenderの内部で扱われているのと同様にフレームを単位として表示されますが、この項目を有効にすると秒による表示になります (fpsの設定から計算されます)。ストーリーボードにはよく秒が単位として使われています。秒の表示に設定すると、いちいち掛け算をするよりも少しだけ楽でしょう。

Play Back Animation

現在のフレームからアニメーションを再生し、終了フレームまで達すると開始フレームからまた再生されます。pauseボタン(またはEsc)を押すまで再生は続きます。

Playbackメニューで設定された条件に当てはまるすべてのウィンドウでアニメーションが表示されます。

編集

Frameメニュー

このメニューは以下の二つ以外は主に [マーカー](#) に関連します。

Set as End (E)

Set as Start (S)

現在のフレームを開始フレームまたは終了フレームに設定します。

再生

Playbackメニュー

このメニューはアニメーションがどこでどのように再生されるかをコントロールします。

Continue Physics

ツールチップには「再生中、フレーム番号にかかわらず物理シミュレーションを続ける」とありますが、私(訳注: 英語版を書いた人...)には意味がわかりません。

Set Frames/Sec

クリックすると新しいfpsを入力できます。

注意: [はじめに](#) に書かれている警告を覚えておいてください。この設定を変えるとすでに作られているアニメーションがスピードアップ/スピードダウンします。

Sync Playback to Frames/Sec

指定されたFPSになるように強制します。

Blenderは、指定のFPSよりも多くのフレームを計算するパワーがあるときは次のフレームまで待ちます。よってこのセッティングはアニメーションがリアルタイムで再生するには重過ぎるときだけ意味があります。デフォルトではBlenderはすべてのフレームを計算するので、場合によっては再生スピードが遅くなります。この項目を有効にすると、必要に応じてフレームが飛ばれて(つまり計算されない)、本来の再生スピードになります。

[後述](#) の“sync audio”ボタンも参照。

その他のオプションは、どのような種類のウィンドウがアニメーションの再生に同期するかに関係します。当然、多くのウィンドウがかかわるほどCPUのパワーが必要になります。

Sequencer Windows

有効にすると、すべてのビデオシーケンスエディタウィンドウが再生に同期します(表示モードにかかわらず)。
[後述](#)の“sync audio”ボタンも参照。

Image Windows

有効にすると、すべてのUV/イメージエディタが再生に同期します。

Buttons Windows

有効にすると、すべてのボタンウィンドウが再生に同期します(これにより、設定した数値が変化の様子がわかります)。

Animation Windows

有効にすると、すべてのアニメーションウィンドウ(Ipoカーブエディタ、アクションエディタ、NLAエディタ)が再生に同期します。

All 3D Windows

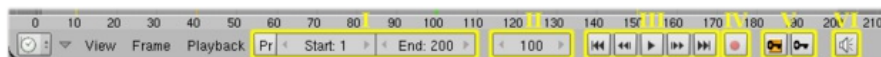
有効にすると、すべての3Dビューウィンドウがアニメーションに同期します。

Top-Left 3D Window

有効にすると、左上の3Dビューウィンドウのみがアニメーションに同期します。

ヘッダ

ヘッダのコントローラーはおおよそビデオ再生のコントローラーを模しています。



The header of the *Timeline* window.

(I) The animation range

最初の三つのコントロールはアニメーションの開始および終了フレームに関わります。

Start/End

開始と終了のフレーム。前述[Frameメニュー](#)のSet as Start/Set as Endを参照。

Pr

再生範囲(Playback Range)の略。通常はStart/Endや、シーンコンテキスト→レンダーサブコンテキスト(F10)のAnimパネルと同じ数値が表示されます。しかしこのボタンを有効にしていると、リアルタイム再生(タイムラインのPlayボタン、またはAltAで開始する再生)時のみの一時的なアニメーション範囲を設定できます。これは長いアニメーションの一部に対して作業しているときに便利です。

(II) The current frame

三番目の数値欄は、カレントフレーム(緑色の線として表示されています)を表示および指定できます。

(III) VTR buttons

これら5つのボタンはアニメーションをナビゲートするのに使います。

The center “play” button

アニメーションを再生します。再生中は「停止」ボタンになります。

The first and last buttons

それぞれ開始フレームと終了フレームに移動します。

The other two

直前/直後に表示されているキーフレームに移動します。

(IV) Automation

「録画」のような赤い丸のマークのボタンは「自動化」とでも呼べるような機能を有効にします。3Dビュー内でオブジェクトを動かすと、それに応じてキーフレームが追加/削除されます。これは再生中にも有効であることに注意してください。オブジェクトをの操作中は再生が止まり、決定すると再開します。

このオプションを有効にすると、そのモードを決めるAuto-Keying Modeのドロップダウンリストが右に現れます。

Replace Keys

存在しているキーフレームのみを置き換えます。新しく追加されることはありません。よってすでにキーフレームが存在している場合だけオブジェクトの変更が記録されます。

Add/Replace Keys

もしキーフレームが存在していれば置き換え、なければ新しく追加します。

User PreferencesウィンドウのEdit Methodsに同じようなオプションがあるのがわかるでしょう。しかし、これらは効果がないように見えます！しかしそれらのすぐ下に、どの曲線が自動的にキーを追加されるかを定める三つのトグルボタンがあります。

Available

すでに存在するすべてのIpoカーブにキーを追加します。

Needed

必要なときだけキーを追加します(変更があったプロパティのカーブにだけ追加します)。

Use Visual Keying

キーの値に影響を与えるようなコンストレイントを持ったオブジェクトに使います。たとえば、Copy Locationコンストレイントを持ったオブジェクトにキーを追加すると、通常はコンストレイントが適用されない場合の位置に対するキーが追加されます。このオプション有効にすると、コンストレイントされた位置へのキーが追加されます。

自動化は3Dビューでの座標変換プロパティに対してのみ動作します (ボタンウィンドウにおけるマテリアルカラーの変更などには使えません)。

(V) Inserting and deleting keys

二つの「キー」ボタンで、カレントフレームでアクティブになっているオブジェクトのキーを挿入 (I) または削除 (AltI) できます。これは理解するのも使うのもやや難しいです！ この操作には最も広いエリア (Blenderのウィンドウ) のコンテキストが使われます。たとえば一番広いのはマテリアルコンテキストのIpoエディタだったとすると、キーはアクティブなマテリアルIpoカーブに対して追加／削除されます。それが3Dビューであれば、IAltを押したのと同じようにキーが追加／削除されます。

(VI) Synchronize with the VSE sound

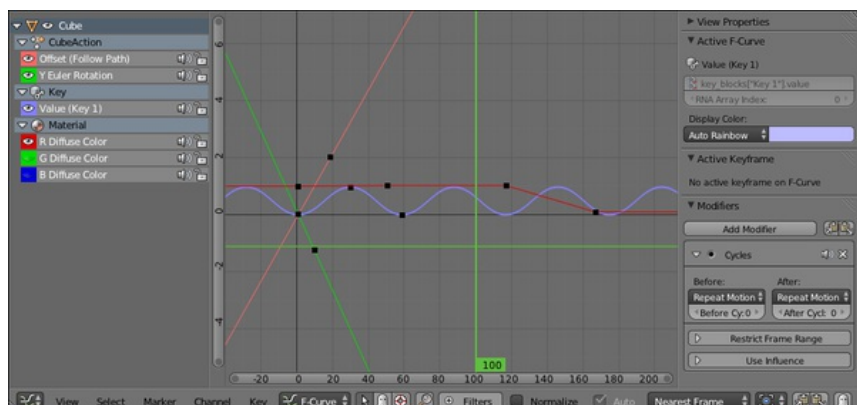
スピーカーのボタンを有効にすると、基本的にPlaybackメニューのSync Playback to Frames/SecとSequencer Windowsを両方とも有効にしたのと同じ効果があります。再生はシーケンサと同期するようになり、そのオーディオが時間の基準として使われます。ビデオの編集作業に便利です。

Graph Editor

(グラフエディタ)

グラフエディタは重要なアニメーションエディタです。これを使って [F-カーブ](#) を使っているあらゆるプロパティのアニメーションを編集できます。

グラフエディタには二つのモードがあります。[アクション\(eng\)](#) 用の「F-カーブ(F-Curves)」と、[ドライバ](#) 用の「ドライバ(Drivers)」です。この二つは機能的によく似ています。

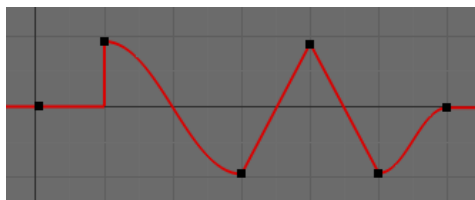


グラフエディタ

Curve Editor Area

(カーブエディタ領域)

ここでカーブとキーフレームを閲覧、編集できます。



さまざまな補間方式のカーブ

詳しくは [F-カーブ](#) と [編集\(eng\)](#) をご覧ください。

Navigation

(操作)

他のウィンドウと同様に、次のことができます:

パン(Pan) MMB

クリックしてドラッグすると、ビューを縦(値)または横(時間)方向にパンします。

ズーム(Zoom) Wheel

マウスホイールでズームイン、ズームアウトします。

ビューの拡縮(Scale View) Ctrl MMB

ビューを縦または横方向に拡縮します。

以下はその他の便利なツールです。

すべてを表示(View All) \backslash Home

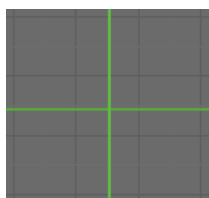
すべてのキーフレームが見えるように表示領域を再設定します。

選択中のものを表示(View Selected) Numpad.

選択状態のキーフレームが見えるように表示領域をリセットします。

2D Cursor

(2Dカーソル)



グラフエディタの2Dカー

ソル

現在のフレームは「タイムカーソル(Time Cursor)」と呼ばれる縦方向の緑の線で表されます。

[タイムライン](#)にあるように、現在のフレームは LMB  を押すか、押し続けることで変えられます。

横方向の緑の線は「カーソル(Cursor)」と呼ばれます。これは [ビュー\(View\)メニュー](#) または [ビュー\(View\)プロパティパネル](#) を使って無効にできます。

「タイムカーソル」と「カーソル」の交差が「2Dカーソル」になります。「2Dカーソル」は主に編集ツールとして使われます。

View Axes

(ビューの軸)

アクションでは X軸は時間を表し、Y軸がそのプロパティに設定される値を表します。

ドライバでは X軸は [ドライバの値](#) を、Y軸はそのプロパティに設定される値を表します。

選択したカーブによって、値の意味はさまざまです: 例えば回転(rotation)プロパティの単位は度(degrees)、位置(location)プロパティの単位はBlender Units です。ドライバは回転プロパティの単位にラジアン(radians)を使うことに注意してください。

Markers

(マーカー)

他のアニメーションエディタと同様、マーカーはエディタの下部に表示されます。



グラフエディタのマーカー

マーカーは [グラフエディタ](#) 内で編集できますが、通常は [タイムライン](#) を使うのが一番です。

詳しくは [マーカーメニュー\(eng\)](#) や [マーカー](#) をご覧ください。

Header

(ヘッダー)

次の項目があります。

- [メニュー](#)
- [グラフエディタのモード](#)
- [ビュー\(View\)のコントロール](#)
- [カーブ\(Curve\)のコントロール](#)

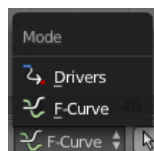
Menus

(メニュー)

詳しくは [Header Menus\(eng\)](#) をご覧ください。

Header Controls

(ヘッダーコントロール)



グラフモード

モード(Mode)

[Actions\(eng\)](#) 用の F-カーブと、[ドライバ](#) 用のドライバ



ビューコントロール

ビュー(View)コントロール

[選択中のもので表示\(Show Only Selected\)](#)

選択中のオブジェクトおよびデータに関係するカーブだけを含めます。

[非表示を表示\(Show Hidden\)](#)

可視状態ではないオブジェクト、ボーンのカーブを含めます。

エラーのみ表示 (Show Only Errors)

無効にされているかエラーのあるカーブだけを含めます。

検索フィルター (Search Filter)

検索語に含まれるキーワードを持つカーブだけを含めます。

タイプフィルター (Type Filter)

プロパティのデータタイプを条件に、カーブを抽出します。

正規化 (Normalize)

値が 1.0 から -1.0 の範囲になるようにカーブを正規化します。

自動 (Auto)

カーブ編集のたびに自動的にカーブの正規化を再計算します。



カーブのコントロール

カーブコントロール

自動スナップ (Auto Snap)

トランスフォーム時にキーフレームを自動的にスナップします。

自動スナップなし (No Auto-Snap)

タイムステップ (Time Step)

近接するフレーム (Nearest Frame)

近接するマーカー (Nearest Marker)

ピボットポイント (Pivot Point)

回転のピボットポイント。

バウンディングボックスの中心 (Bounding Box Center)

選択中のキーフレームの中心

2D カーソル

2Dカーソルの中心。2Dカーソルとは、タイムカーソルとカーソルの位置です。

それぞれの中心 (Individual Centers)

キーフレームそれぞれの「ベジエ」ハンドルを回転します。

キーフレームをコピー (Copy Keyframes) CtrlC

選択したキーフレームをメモリにコピーします。

キーフレームをペースト (Paste Keyframes) CtrlV

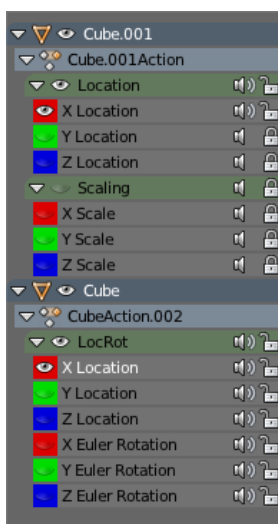
選択したカーブの現在のフレームに、メモリからキーフレームをペーストします。

スナップショットの作成 (Create Snapshot)

カーブの現在の形を表す絵(ゴースト)を、グラフィエディタの背景に表示します。

Channels Region

(チャンネル領域)



チャンネル領域

チャンネル領域はグラフィエディタのカーブを選んだり管理したりするのに使います。

カーブを隠す (*Hide curve*)

目のアイコンで表されます。

カーブの非アクティブ化／ミュート (*Deactive/Mute curve*)

スピーカーのアイコンで表されます。

カーブ編集のロック (*Lock curve from editing*)

南京錠のアイコンで表されます。

Channel Editing

(チャンネルの編集)

チャンネルの選択 LMB

複数選択／選択解除 ⇧ Shift LMB

すべてを選択／すべてを選択解除 A

境界選択 (*Border Select*) Drag LMB または B Drag LMB

境界選択解除 (*Border Deselect*) ⇧ ShiftDrag LMB または B ⇧ ShiftDrag LMB

選択したものを削除 (*Delete Selected*) X または Delete

選択したものをロック (*Lock selected*) ⇔ Tab

選択したものをだけ可視化 (*Make only selected visible*) V

選択したもののミュート (*Mute*) とロック (*Lock*) の有効化 ⇧ ShiftCtrlW

選択したもののミュート (*Mute*) とロック (*Lock*) の無効化 AltW

選択したもののミュート (*Mute*) とロック (*Lock*) の切り替え ⇧ ShiftW

詳しくは [Channel Menu\(eng\)](#) をご覧ください。

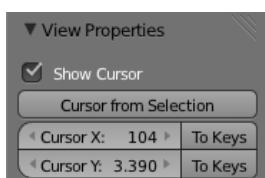
Properties Region

(プロパティ領域)

プロパティ領域(Nキー)にあるパネル

View Properties Panel

(ビュープロパティパネル)



ビュープロパティパネル

カーソルを表示 (Show Cursor)

縦方向のカーソル (Cursor) を表示します。

選択からカーソル (Cursor from Selection)

2Dカーソルを、選択中のキーフレームの中央に合わせます。

カーソル X (Cursor X)

タイムカーソルの X 位置。

キーに (To Keys)

選択中のキーフレームをタイムカーソルに合わせます。

カーソル Y (Cursor Y)

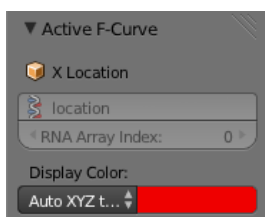
縦方向のカーソルの Y 位置。

キーに (To Keys)

選択中のキーフレームをカーソルに合わせます。

Active F-Curve Panel

(アクティブ F-カーブパネル)



アクティブ F-カーブパネル

アクティブな F-カーブのプロパティを表示するパネルです。

チャンネル名 (Channel Name) (X Location)

ID Type + チャンネル名。

RNA Path

プロパティの RNA Path + 配列添字 (Array index)。

カラーモード (Color Mode)

アクティブな F-カーブの彩色設定

自動で虹色に (Auto Rainbow)

F-カーブの HUE をチャンネルインデックスに基いて増やします。

自動でXYZをRGB色に (Auto XYZ to RGB)

xyz位置のようなプロパティセットに対して、自動的に赤、緑、青の色セットを設定します。

ユーザー定義 (User Defined)

アクティブな F-カーブに独自色を定義します。

Active Keyframe Panel

(アクティブキーフレームパネル)



アクティブキーフレームパネル

補間 (Interpolation)

アクティブなキーフレームに、順方向の補間を設定します。

一定 (Constant)

次のキーフレームまで同じ値を維持します。

直線/ニア (Linear)

次のキーフレームとの差分を取ります。

ベジエ (Bezier)

次のキーフレームまでの、ベジエ曲線による補間です。

キー (Key)

フレーム (Frame)

アクティブなキーフレームのフレームを設定します。

値 (Value)

アクティブなキーフレームの値を設定します。

左ハンドル (Left Handle)

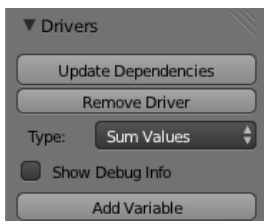
アクティブなキーフレームの、左補間ハンドルの位置を設定します。

右ハンドル (Right Handle)

アクティブなキーフレームの、右補間ハンドルの位置を設定します。

Drivers Panel

(ドライバーパネル)



ドライバーパネル

詳しくは [Drivers Panel \(eng\)](#) をご覧ください。

Modifiers Panel

(モディファイアパネル)



モディファイアパネル

詳しくは [F-カーブモディファイア](#) をご覧ください。

参考

- [グラフエディタ - F-カーブ](#)
- [グラフエディタ - F-カーブモディファイア](#)
- [Graph Editor - Editing \(eng\)](#)
- [Actions \(eng\)](#)
- [ドライバ](#)

F-カーブ

何かに対してキーフレームを作ったら、それに対応するカーブを編集することができます。Blender 2.5では、F-カーブというものがIPOカーブに代わって登場しました。しかし、カーブの編集は基本的には同じです。

補間のコンセプト

何かアニメートされているとき、それは時間に応じて変化しているわけです。Blenderでは、オブジェクトをアニメートということは、ある時間の範囲にわたってそのプロパティを変化させることです。たとえばX方向の位置や、マテリアルの拡散色のRed値等です。

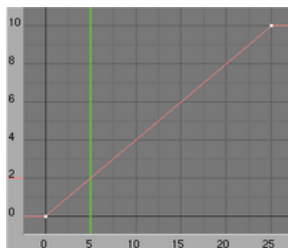
以前触れたように、Blenderにおける基本的な時間の単位は「フレーム」です。1フレームの時間的長さはフレームレートによって決まり、通常、非常に短い時間です。

アニメーションは複数のフレームにわたる少しずつの変化によってつくられるので、通常は変化するプロパティを1フレームごとに手作業で調整したりはしません。その理由は、

- そんなことをしていたら何年もかかる！
- 滑らかな変化をさせるのが難しい(フレームごとに数値を正確に計算するなら別ですが、それは大変すぎます)。

これが、ほとんどのアニメーションが補間(interpolation)を使って作られる理由です。

概念としては簡単です。いくつかの「制御点」、キーフレームと呼ばれるものを定義します。これはフレームの間隔をあけて設定します。キーフレーム同士の間フレームについては、プロパティの数値はBlenderによって計算(補間)されて埋められます。これにより、製作者の負担は大幅に減ります。



Example of interpolation

たとえば、以下のような状態のとき、

- フレーム0では数値が0。
- フレーム25では数値が10。
- 線形補間である。

この場合、フレーム5における数値は2になります。

このようなことが途中のフレームのすべてで行われます。最初と最後の二つの点を設定すれば、25フレームにわたって数値0から10まで滑らかに変化します。そしてフレーム15で数値が9になるようにしたければ、あたらな制御点(キーフレーム)を追加すればいいのです。

補間の種類

「カーブ」という名前ですが、実際には直線の組み合わせでもありえるし、非連続的な値の集合でもありえます(「階段状のカーブ」として表されます)。これは補間モードによって変わります。

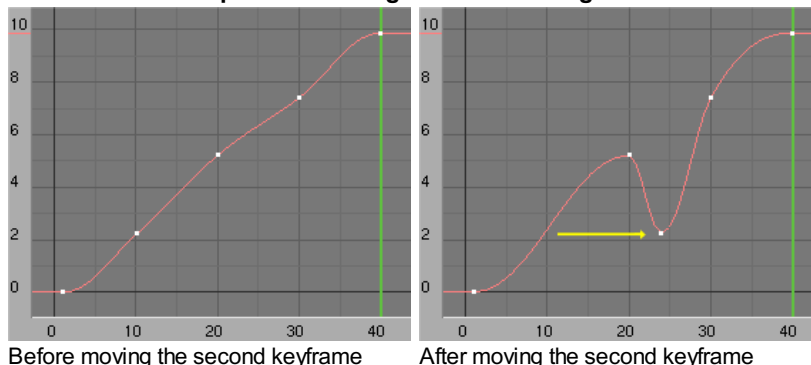
時間の進む方向

F-カーブはベジェカーブによく似ていますが、補間の仕方が少し違います。

明らかな理由から、カーブによって示されるプロパティは個々の時点につき一つだけです。よって、

- 制御点を動かして、先の時点にある別の制御点を飛び越えようとすると、カーブ上でその二つの点の順序が入れ替わります。カーブが時間をさかのぼって進むのを防ぐためです。
- 上記と同じ理由で、閉じたカーブは作れません。

Two control points switching: the curve can't go back in time!



カーブの選択

デフォルトでは、チャンネルにキーフレームを追加すると、グラフエディタはエディットモードでそれを表示します。3Dビューと同じように、チャンネルの編集可能状態は⇄ Tabで切り替えられます。チャンネルボックスのlockアイコンでも切り替えられます。

選択ツール

カーブハンドルを選択するには LMB (複数選択には⇄ Shift LMB) でクリックします。中心点を選択してキーそのものを選択することも、ハンドルだけを選択することもできます。

Aを押して(またはSelect » Select/Deselect All)すべてを選択(または選択解除)することもできます。またB(またはSelect » Border Select)で矩形選択です。LMBドラッグで選択を追加、RMB(またはCtrlAlt LMB)で選択から除外できます。

Select All A

Invert Selection CtrlI

Border Select B

Border Axis Range AltB

Border (include Handles CtrlB

Columns on Selected Keys K

Column on current Frame CtrlK

Columns on selected Markers ⇄ ShiftK

Between Selected Markers AltK

Before Current Frame [

After Current Frame]

Select More Ctrl+ NumPad

Select Less Ctrl- NumPad

Select Linked L

カーブの編集

座標変換

選択されたカーブ(白い制御点またはキーフレーム)は、移動(G)、回転(R)、拡大縮小(S)することができます。Curve » Transformメニューでも可能です。

さらに、移動と拡大縮小の場合、X軸(時間)またはY軸(値)の方向のみに制限することができます。変換中にXまたはYを押してください。

また、おなじみの「スナップ」機能も使えます。座標変換の際にCtrlを押しながらコントロールすると、1フレーム/1数値ごとのステップで調整できます。

⇄ Shiftを押しながらだとゆっくりした動きになり細かい調整ができます。ただし、⇄ ShiftCtrlを押していても、動きがゆっくりになるだけでステップは細かくなりません。

コピー

ひとつまたは複数のキーフレームをバッファにコピーすることができます。そして、同じカーブ上にペーストできます。これはヘッダにある二つのボタンによって操作します。

- The “copy” CtrlCボタン(下向きの矢印)で、選択かつ表示されているカーブをコピーします。
- The “paste” CtrlVボタン(上向きの矢印)で、表示されているカーブにバッファの内容をペーストします。

このツールは“keyframe”モードでは使えないので注意してください。また、エディットモードでは(カーブ全体ではなく)選択した点のみコピーします。

Keyメニューでキーを複製することもできます。

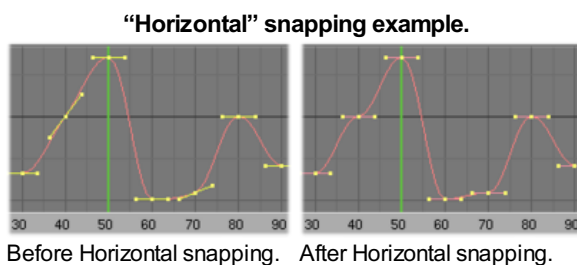
削除

キーフレームを削除するには、「(デフォルト)」のモードで)選択し、Xを押します(またはCurve » Delete)。

スナップ

いくつかのスナップコマンドが指定できます(⇄ ShiftSまたはCurve » Snap)。

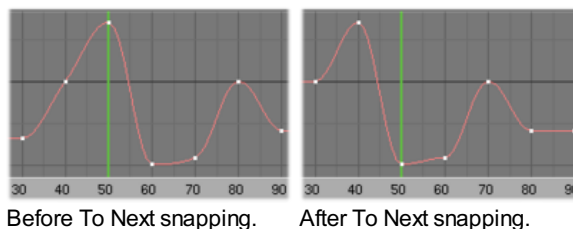
- Horizontal : これはデフォルトのF-カーブの場合だけ有用です。選択したカーブのすべての制御点を水平に回転します。



- To Next : 選択したカーブのすべての制御点(キーフレーム)を隣の制御点

“To Next” snapping example.

の値に変更します(カーブを横にずらすような効果になります)。



- To Frame : 選択したカーブのすべての制御点を、それぞれ最も近い整数フレームに移動します(フレーム**23.2**に設定されているキーフレームはフレーム**23**に移動します)。
- To Current Frame : これは「デフォルト」モードでは何の効果もありません。

反転

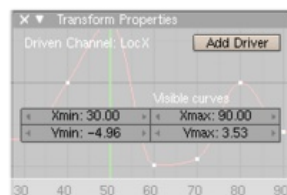
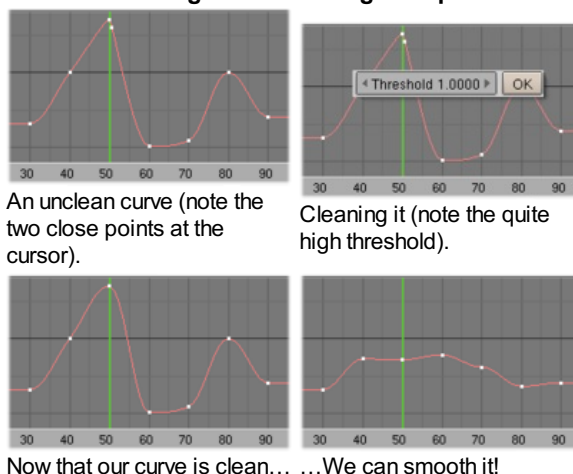
選択したカーブを反転することができます(⇧ ShiftMまたはCurve » Mirror)。反転の軸は、カレントフレーム、縦軸(フレーム**0**)、横軸(値**0**)から選べます。

クリーニングとスムージング

選択したカーブをクリーニングすることができます。たとえば時間的にも数値的にも非常に近い位置にある制御点を除去できます。Oを押して(またはCurve » Clean Keyframesメニュー)、現れたダイアログでしきい値を指定し、OKをクリックします。

選択したカーブを滑らかにすることもできます(AltOまたはCurve » Smooth Keys)。ただし、このスムージングアルゴリズムはそれぞれのキーフレーム間の距離を2で割って線形的に数値を平均するようで、何の設定項目もなく非常に滑らかになってしまいます! また最初と最後のキーは動かされません。

Cleaning and Smoothing examples.



The Transform Properties panel of curves in “default” mode.

Transform Properties panel

そう、3Dビューと同じです。N(またはView » Channel Properties)で表示/非表示もできます。デフォルトのモードではこれはそれほど有用ではありません。カーブ全体のXおよびY方向の最大値・最小値を表示、編集できるだけです。この値を変更するとカーブを拡大縮小できます。

Sample Keyframes ⇧ ShiftO

それぞれのフレームにおいて補間された値をキーフレームに置き換えます。

Bake Curves AltC

カーブをサンプリングした点に置き換え、さらに編集をできないようにします。

補間と補外

F-カーブは三つの補助プロパティを持っています。二点間の補間のしかた、延長の仕方、ハンドルのタイプ、をコントロールします。

補間

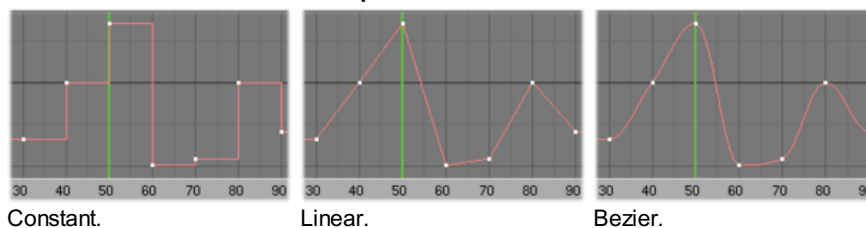
3種類あります(TまたはCurve » Interpolation Mode)。

- Constant まったく補間されません。カーブは直前のキーフレームでの値を、次のフレームの直前まで保持しつづけ、不連続な(階段状の)「カーブ」になります。つかいどころは多くないでしょう。
- Linear 二点の間を直線的に補間します。カーブは折れ線のようになります。これは二つだけのキーフレームとExtrapolation延長モードを使って、無限に続く直線を生み出すのに便利です。
- Bezier より強力で便利な補間です。またこれがデフォルトの設定です。これはすばらしく滑らかなカーブ、つまり滑らかなアニメーションを作

ります。

ある種のプロパティは不連続な値しか取れないことに注意してください。その場合は設定に関わらずConstantのように見えます。

The three interpolations available for curves.



補外

補外(または外挿)とは、最初のキーフレームよりも前と、最後のキーフレームよりも後のカーブの続け方です。

二つの基本的な補外モードがあります(⇧ ShiftEまたはChannel » Extrapolation Mode)。

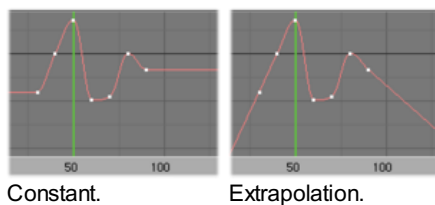
Constant

デフォルトの設定です。最初のキーフレームの値と、最後のキーフレームの値をそれぞれそのまま延長します。

Linear

カーブの端は最初の二点(または最後の二点)を通る直線として延長されます。

The two extensions available for lpo curves.



そのほかの補外ツールは[F-カーブモディファイア](#)の中にあります。

ハンドルの種類

Bezier補間の際に便利なオプションです。カーブを制御するハンドルの種類を設定できます(V)。

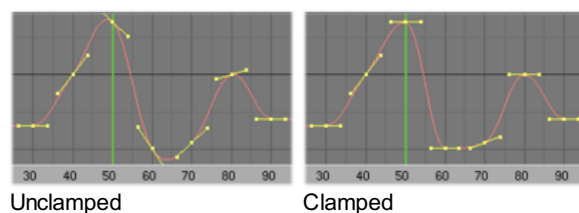
Automatic

Vector

Aligned

Free

Auto Clamped



F-カーブモディファイア

F-カーブモディファイアはオブジェクトモディファイアと似ていて、非破壊的に効果を追加し、それはいつでも調整できます。またレイヤー化して複雑な効果を生み出すこともできます。

モディファイアの追加

F-カーブモディファイアパネルはプロパティパネルにあります。まずカーブ上のどこか一点を選択するか、チャンネルリストから選んで、カーブを選択状態にします。そして **Add Modifier** ボタンをクリックし、モディファイアの種類を選びます。

モディファイアの種類

Generator

Generatorは因数分解または展開された多項式です。これは値によって直線、放物線、またはより複雑なカーブを表せる数学的な式です。

Additive

このオプションは、デフォルトのようにカーブを置き換えるのではなく、モディファイアの効果をカーブに足します。

Poly Order

多項式の次数を指定します。つまりxの指数の中で一番大きなものです(係数の個数よりも一つ少ない数になります)。

係数を変えるとカーブの形が変わります。多項式については[ウィキペディアの当該ページ](#)を参照してください。

Built-in Function

上記に加えて以下のような数式があります。形状をコントロールするためのオプションは共通しています。それぞれの関数の詳細については数学の文献を参照してください。

- Sine
- Cosine
- Tangent
- Square Root
- Natural Logarithm
- Normalized Sine (sin(x)/x)

Amplitude

Y方向のスケールを調整します。

Phase Multiplier

方向のスケールを調整します。

Phase Offset

X方向のオフセットを調整します。

Value Offset

Y方向のオフセットを調整します。

Envelope

制御点を使ってカーブの全体的な形を調整します。

Reference Value

エンベロープの中心となるY座標を指定します。

Min

Reference Valueからマイナス方向へのエンベロープの大きさ。

Max

Reference Valueからプラス方向へのエンベロープの大きさ。

Add Point

制御点を現在のフレーム位置に追加します。

Fra:

制御点のフレーム位置。

Min

制御点のマイナス方向のY位置。

Max

制御点のプラス方向のY位置。

Cycles

二つ以上の制御点を持った循環的なモーションをカーブに追加します。オプションはカーブの前後の部分に関わる設定です。

Cycle Mode

Repeat Motion

まったく同じようにカーブを繰り返します。

Repeat with Offset

カーブの最初の点と最後の点がつながるように繰り返します。

Repeat Mirrored

カーブはX軸方向に反転されてつなげられます。

Before/After Cycles

カーブを繰り返す回数。0にすると無限に繰り返します。

Noise

カーブにノイズを付け加えます。繊細または極端なランダムさをアニメーションの動きに与えます。たとえばカメラの揺れなどです。

Blend Type

Replace

カーブに対してプラスマイナス0.5の範囲でノイズを加えます。

Add

カーブに対して0から1の範囲でノイズを加えます。

Subtract

カーブに対して0から1の範囲でノイズを引き算します。

Multiply

カーブに対して0から1の範囲でノイズを掛け算します。

Scale

ノイズの全体的なスケールを調整します。数値が0から遠くなるほどゆるい乱れになります。

Strength

ノイズのY方向のスケールを調整します。

Phase

乱数の種を調整します。

Depth

ノイズの細部の緻密さを調整します。

Python

Limits

カーブの値を指定したX方向およびY方向の範囲に制限します。

Minimum/Maximum X

この範囲外のカーブはカットされ、それらの境界点での値にセットされます。

Minimum/Maximum Y

この範囲外のカーブは範囲内に収められます。

Stepped

指定したフレーム間隔ごとに値を丸めることによって階段状のカーブを作ります。

Step Size

一つのステップのフレーム数。

Offset

値を維持しはじめるフレームを決めます。

Use Start Frame

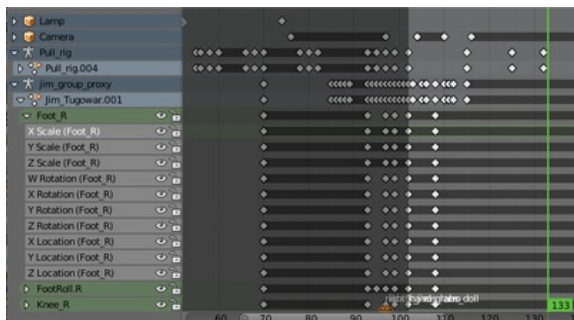
指定したStart Frame以降のみにモディファイアを適用します。

Use End Frame

指定したEnd Frame以前のみにモディファイアを適用します。

The Dopesheet

(ドープシート)

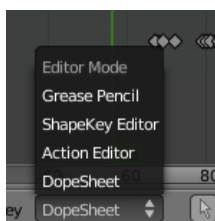


The DopeSheet

古くは手描きのアニメーターはよく、絵や音、カメラがいつ、どれだけ動くのか、正確に示す表を作りました。この表のニックネームは「ドープシート(競馬予想紙)」でした。CGの根本は手描きのアニメーションと大きく違いますが、Blenderのドープシートは類似した指示書を受け継いでいます。アニメーターはシーン内で起きるすべてのことを「鳥瞰図」のように見ることができます。

Dope Sheet Modes

(ドープシートのモード)



DopeSheet modes

ドープシートには4つの基本的なビューがあります。アニメーションを違った局面から見るができます:

[ドープシート\(DopeSheet\)\(eng\)](#)

ドープシートでは複数のアクションを一度に編集できます。

[アクションエディタ\(Action Editor\)\(eng\)](#)

アクションエディタはデフォルトの、もっとも便利なモードです。アクションの定義や調節はここで行えます。

[シェイプキーエディタ\(Shape Key Editor\)\(eng\)](#)

シェイプキーエディタはシェイプ(Shape) lpo データブロック専用です。直前のモードと同じアクションデータブロックを使用・編集します。ただ、アクションエディタモードがシェイプチャンネルをとてうまく扱え、それ以上の要素もないので、いまでは役に立たなくなっているようです。

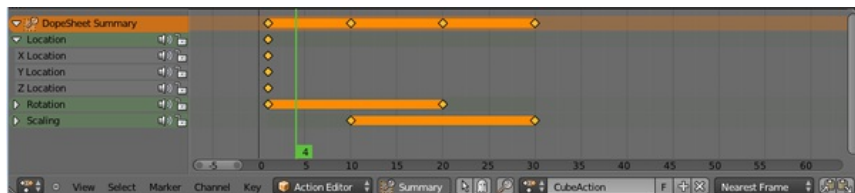
[グリースペンシル\(Grease Pencil\)\(eng\)](#)

グリースペンシルは [グリースペンシルツール](#) のキーフレーム専用です。グリースペンシルレイヤーごとにストリップがあり、これでキーをつかむことができるので、スケッチのアニメーションタイミングを簡単に割り当て直せます。このモードは、グリースペンシルのデータを閲覧し、編集する別の方法にすぎないので、データブロックを使用しません(このためアクションとは無関係です)。スケッチしたウィンドウの数だけ(3Dビュー、UV/画像エディタ等)、上層のグリースペンシルチャンネルができることに注意してください。

Interface

(インターフェイス)

アクションエディタ(Action Editor) のインターフェイスはどこか FCurve エディタのものに似ており、3つの領域に分かれています:



ヘッダーバー

メニュー、エディタのモードに関連するコントロール群、アクションデータブロックに関連するコントロール群、それに2、3の他のツール(コピー・ペーストボタンやスナップの種類など)があります。

メイン領域

この中には、すべての可視のアクションチャンネルのキーフレームがあります。

他の「時系列」ウィンドウと同様、X軸が時間を表します。Y軸は FCurve エディタと違ってそれ自身に意味を持たず、単にアクションチャンネルが積み重なった状態です。それぞれが水平方向に伸びる色のついたストリップとして表示されます(動いている/キーを打たれた区間は濃い色になります)。

チャンネルストリップ上にキーフレームが置かれ、薄い灰色(非選択状態)か黄色(選択状態)のダイヤモンドで表現されます。

このウィンドウの重要な機能のひとつは、どのチャンネル(例えば lpo曲線)が「実際に」影響を受けるのか、ただちに知られることです。二つの隣り合うキーフレーム間で指定チャンネルの値がまったく変化しない場合、この間には、灰色(非選択)か黄色(選択)の線が引かれます。

左の「リストツリー」


この部分はアクションチャンネルの「ヘッダー」およびその階層構造を表しています。基本的に、次の項目を含みます:


- 「上層」のチャンネル。FCurve データブロック全体を表しています(したがって Object 用のもの、Shape 用のもの...等があります)。元になる FCurve データブロックで定義された、すべてのキーフレームの集合です。
- 「中層」のチャンネル。現在は使われていないようです(上層チャンネルごとに存在しており、すべて FCurves という名前で、オプションはひとつもありません)。
- 「下層」のチャンネル。個々の FCurve と、そのキーフレームを表します(幸いにも、キーを打たれた lpo だけが表示されます)。

各層は親のチャンネルの左にある小さな矢印で、広げたり折り畳んだりできます。

チャンネルのヘッダーの右には、チャンネルの設定用コントロールがあります:

- 小さな「スピーカー」をクリックすると、そのチャンネルをミュートできます(子チャンネルがあれば、それもすべてミュートされます)。
- 小さな「南京錠」をクリックすると、このチャンネルと子チャンネルを編集できないようにします(NLA内でも動作しますが、元になるFCurveの編集はできることに注意してください)。

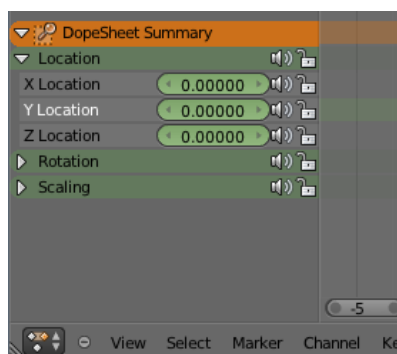
チャンネルは選択状態(文字が白く、ストリップが灰-青色)か非選択状態(文字が黒く、ストリップがピンク-茶色)にできます。切り替えには LMB  クリックを使ってください。

ヘッダーを Ctrl LMB  すると、チャンネルのプロパティにアクセスできます。

最後に、この他に、現在のキーフレームの値を変更したり、新規に追加したりできる、値のスライダーを持つ列があります。低層のチャンネル(例: 個々のFCurve)に対してのみ利用できます。スライダーの表示のさせ方は [後述](#) します。

View Menu

(ビューメニュー)



スライダーを表示させたアクションエディタ

リアルタイム更新 (Realtime Updates)

キーフレームをトランスフォームしたときに、アニメーションへの変更が他のビューに反映されます

フレーム数インジケータを表示 (Show Frame Number Indicator)

現在のフレームを表す線の近くに、フレーム番号を表示します

スライダーを表示 (Show Sliders)

チャンネルに値のスライダーを表示させるかどうかの切り替えです(図「Object と Shape チャンネルを持つアクションエディタモードのアクションエディタ ウィンドウ」をご覧ください)。

グループ色を使う (Use Group Colors)

グループとチャンネルを、対応するグループとマッチする色で描画します。

AutoMerge Keyframes

近接するキーフレームを自動的に結合します

Sync Markers

マーカーをキーフレームの編集に同期します

Show Seconds

X軸に時間をフレームと秒のどちらで表示するか選びます。

Set Preview RangeP

再生に使われるフレーム範囲をインタラクティブに決めます。AltA のリアルタイム再生で使われる、一時的なプレビュー範囲を決めることができます(これは [Timeline ウィンドウヘッダー](#) の Pr オプションと同じものです)。

Clear Preview RangeAltP

プレビュー範囲を消去します。

Auto-Set Preview Range

アクション全体が再生されるように、プレビュー範囲を自動的に設定します。

Select Menu

(選択メニュー) [F-カーブ](#) をご覧ください。

Marker Menu

(マーカーメニュー) [マーカー](#) をご覧ください。

ノンリアアニメーション(NAL)エディタ

NLAエディタは退屈なキーフレーム作業をせずにアクションを操作、再構成できます。これは製作中のアニメーションに対して広範で重大な変更を、比較的簡単に行うためによく使われます。またこれはアクションを再構成し、レイヤー化することもできます。アニメーションを構成、バージョン管理するのが楽になります。

トラック

トラックはNLAにおけるレイヤーシステムです。基本的な使い方ではストリップを構成するのに使いますが、モーションを階層化することもできます。グラフィックソフトがピクセルを階層化すると似ています。下のレイヤーが先で、上のレイヤーが後です。



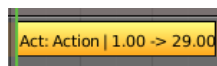
ストリップ

三種類のストリップ(細長く続いたもの)があります。アクション、トランジション(移行)、メタです。アクションは実際のキーフレームデータを内包しています。トランジションはアクション同士の間動きを計算します。メタは複数のストリップをグループ化できます。

アクションストリップの作成

アクションをNLAで使うには、まずアクションストリップにする必要があります。それには、NLAでリストにあるアクションの隣の✳️マークをクリックします。

Menu: Add → Action



Action Strip.

トランジションストリップの作成

同じトラックにある二つのストリップを選択し、

Menu: Add → Transition



Transition Strip.

メタストリップの作成

たくさんのストリップを同じように動かすことが多いと感じたら、それをメタストリップにまとめると便利でしょう。これは複数のストリップをグループにして、一つのストリップとして扱えます。

Menu: Add → Meta Strips



Shift-select two or more strips..

Combine them into a meta strip.

ストリップの編集

アクションストリップの内容は編集できます。そのためには調整(Tweak)モードに入ります。

Hotkey: ⇄ Tab

Menu: View → Enter Tweak Mode



Strip in NLA mode..

Strip in Tweak mode.

調整モードでストリップを動かすと、キーも一緒に移動することに気づくでしょう。場合によってはキーがオリジナルの位置にとどまってほしいこともあります。そうするにはストリップの横のunpinアイコンを押してください。



Nla strip with pinned keys.



Strip moved, notice the keys move with it.



The unpinned keys return to their original frames.

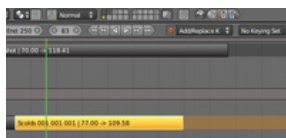
編集が終わったら、View > Exitまたはtabで調整モードを抜けます。

ストリップの再実体化

アクションストリップの内容は何度も使うことができます。ストリップを追加するには、ストリップを選択し、

Menu: Edit→ Duplicate Strips

一つのストリップの内容を調整すると、複製されたストリップも同様に変更されます。オリジナルのストリップは赤い色になります。



Original strip.



Duplicated strip.



Duplicated strip being edited.

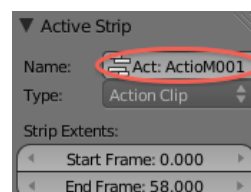
ストリップのプロパティ

ストリップのプロパティはヘッダからアクセスできます。

Menu: View→ Properties

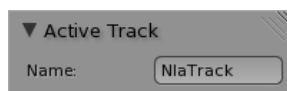
ストリップの名前変更

すべてのストリップは名前が変更できます。ストリッププロパティの"Active Track"セクションで変更してください。



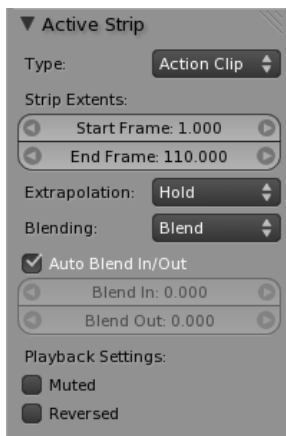
Active Track

ストリップが現在属しているトラックです。



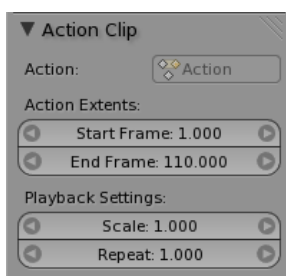
Active Strip

ストリップそのものの属性です。typeはアクションクリップまたはトランジションクリップです。'Strip Extents'欄はストリップの属性であって、アクションの属性ではないことに注意してください。また、Extrapolationのリストにある"Hold"はクリップの前後両方に対して作用します。これは前のクリップが上手く働かなくなる可能性があります。

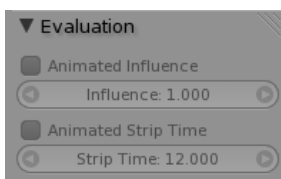


Active Action

これはストリップの「オブジェクト」としてのデータを表しています。オブジェクトの座標変換値に似ています。

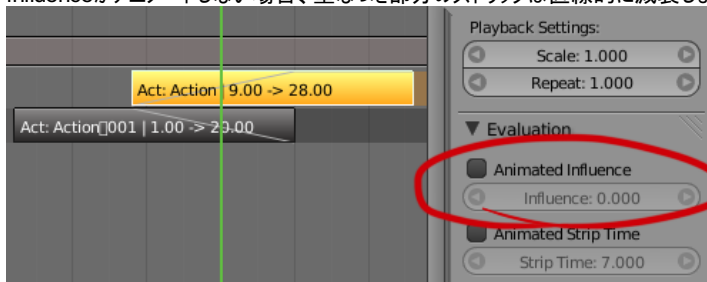


Evaluation



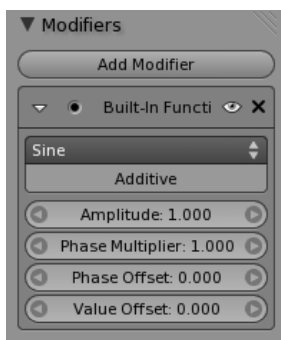
これはストリップの影響力およびその時間範囲を指定します。

influenceがアニメートしない場合、重なった部分のストリップは直線的に減衰します。



Strip Modifiers

メッシュやグラフの場合と同様に、モディファイアは様々な効果をスタックとして組み合わせられます。写真はごく一例です。



コンストレイントを使ったアニメーション

コンストレイントはオブジェクトのプロパティ(位置、回転、拡大縮小)をコントロールする方法の一つで、それには単純な数値(“**limit**”**コンストレイント**など)かターゲットと呼ばれるオブジェクト(“**copy**”**コンストレイント**など)を用います。

コンストレイントは静止画の制作にも便利ですが、あきらかに、主な用途はアニメーションの制作です。コンストレイントのアニメーションには二つの側面があります。

- コンストレイントのターゲットを通じてオブジェクトのアニメーションを制御できる(間接的アニメーション)。
- コンストレイントの設定をアニメートできる。

コンストレイントによるアニメーションの制御

これはターゲットを用いるコンストレイントの場合です。ターゲットはコンストレイントのオーナーのプロパティをコントロールでき、したがってターゲットをアニメートすることは間接的にオーナーをアニメートすることになります。

コンストレイントによる間接的なアニメーションは非常にシンプルに行うこともできます。たとえば**Copy Locationコンストレイント**を使うと、オーナーオブジェクトは単純にターゲットの位置を(指定したオフセットをともなって)コピーします。しかし、たとえば**Actionコンストレイント**のようなものを使えば、非常に複雑なふるまいをさせることもできます。Actionコンストレイントはアクションに対する**アニメーションドライバ**の一種です。

またクラシックな**Child Ofコンストレイント**にも触れなくてはならないでしょう。これはオブジェクトの親子関係を生成します。この関係には暗黙のうちに間接的アニメーションが含まれます(親オブジェクトを移動すると子オブジェクトにも影響します)。しかし、Child Ofコンストレイントのもうひとつの重要な点は、ボーンとオブジェクトを親子にすることによって、**アーマチュア**を使ったアニメートができるということです。

シンプルなCopy Locationの例に戻りましょう。このコンストレイントは、二つの異なった振る舞いをします。

- Offsetボタンを無効(デフォルト)にしていると、オーナーの位置はターゲットによって「絶対的に」コントロールされます。つまり、ターゲット(および下層にあるコンストレイント)以外のどんなものもオーナーの位置に影響を与えることはできません。オブジェクトのアニメーションカーブであってもです。
- しかし、Offsetボタンを有効にすると、オーナーの位置は「相対的に」コントロールされます。つまりオーナーの位置は、ターゲットの位置を上乗せされる形で変更されます。そして、もとの位置はたとえばLoc...カーブ(またはアクションやNLA)でコントロールできます。

例

Copy LocationコンストレイントとOffsetボタンを使ってみましょう。まず、オーナーのオブジェクト(moonと呼びましょう)に(0.0, 0.0, 0.0)を中心とした円を描かせる(LocX/LocYのアニメーションカーブにパイドライバを使います。**ドライバ**を参照)。そしてターゲット(earthと呼びましょう)の位置をコピーするように設定し、Offsetボタンを有効にします。これで惑星の周りを回る衛星ができました。それから同じようにして、この惑星がさらに恒星オブジェクト(名前はsunがいいでしょう)の周りを回るようにすることもできるし、その恒星が銀河を回るようにすることもできます。

これは上記(およびいくつかの他の)テクニックを用いて作った太陽系のアニメーションです。

[\[video link\]](#)

ただしこの「太陽系」はリアリティではありません(縮尺も回転の方向も間違っています...)

このアニメーションを作った**File:ManAnimationTechsUsingConstraintsExSolarSys.blend**をダウンロードできます。

コンストレイントの影響をアニメートする

より「クラシック」に、アニメーションカーブを使ってコンストレイントのプロパティをアニメートすることもできます。

使えるのは二つのカーブだけです(**グラフィティ**を参照)。

- コンストレイントのInfluence(影響力)をアニメートできます。上の太陽系の場合、カメラがmoonとearthについていき、それから止まるようにしています。それにはOffsetを有効にした二つのCopy Locationコンストレイントを使い、Influenceをクロスフェードさせました。
- より間接的に、アーマチュアのボーンをターゲットとしたコンストレイントを使って、そのボーン上(rootとtipの間)のどの点を真のターゲットポイントとするかをアニメートすることができます(**0.0**はrootの位置、**1.0**はtipの位置です)。

パスに沿ってオブジェクトを動かす

アニメーションでは、パス(軌道)に沿ってオブジェクトを動かすことがよく必要になります。複雑なカメラの動きや、レールの上の列車、「見えない路線」に従って動いている多くの乗り物、それから自転車のチェーンなどを考えてみてください。これらの動きはすべて、通常のlpoカーブを使って作ることももちろん可能です。しかしそれをやるのは悪夢になるでしょう！ それよりも、目的の動きを表現したパスを作り、オブジェクトをそれに従わせるほうが簡単で直感的です。

Blenderには、オブジェクトをパスに従わせるコンストレイントが二種類あります。それらはパス上でのオブジェクトの位置を決定・アニメートする方法が異なっています。

Blenderではどんな**カーブオブジェクト**でもパスにできます。CurveデータパネルでPath Animationボタンを有効にすると、カーブがパスになります。しかし、わざわざその操作をする必要はありません。パスコンストレイントのターゲットとして選択したカーブは自動的にこのオプションが有効になります。

また、最初からパスとしてオブジェクトを追加することもできます。3DビューでAdd » Curve » Pathメニューを選んでください。3次元のNURBSカーブが追加されます。デフォルトでは、Blenderのカーブは2次元です。つまり平面状に張り付いているようなカーブです。これはパスとしては望ましくないことも多いでしょう。通所のカーブを3次元のカーブにするには、Curve and Surfaceパネルで3Dボタンを押してください。

パスのプロパティとしてもうひとつ重要なのは、方向です。これは3次元のカーブの場合、矢印で表示されます。Curve » Segments » Switch Direction(またはW2 NumPad)で方向を切り替えられます。

パスやカーブの編集については[モデリングの章](#)を参照してください。

カーブ上のオブジェクト形状

オブジェクトの形状そのものがパスに沿ってほしい場合(たとえばプリンタの内部の紙など)は、カーブモディファイアを使ってください。

Parenting Method

古いバージョンのBlenderでは、オブジェクトをパスに従わせるコンストレイントはなく、別の方法(推奨はされませんが使用できます)を使っていました。親子の関連付けをベースにした方法です。

この方法を使うには、まずパスに沿わせるオブジェクトを選択し、◊ Shiftを押しながらカーブも選択し、CtrlPのメニューからFollow Pathを選びます。これでオブジェクトはパスに沿ってアニメーションします。

パスアニメーションの設定はカーブプロパティのPath Animationパネルにあります。

Frames

パス上を動くのに要するフレーム数を定義します。

Evaluation Time

カレントフレームを定義します。デフォルトではアニメーション全体のフレーム番号とリンクしていますが、キーフレームを追加してより細かくコントロールすることもできます。

Follow

カーブの子オブジェクトがパスの方向に従うようにします。

Radius

カーブの子オブジェクトがカーブの半径に従って拡大縮小されます。[カーブの押し出し\(eng\)](#)を参照。

Offset Children

カーブの子オブジェクトのtime offset値に従ってアニメーションがオフセットされます。time offsetの設定はObject Panel/パネルのAnimation Hacksセクションにあります。

The Follow Pathコンストレイント

Follow Pathコンストレイントが実現するのはもっとも「クラシックな」テクニックです。デフォルトでは、オーナーオブジェクトはパス全体を一度だけ、フレーム1から100フレームかけて移動します。コンストレイントパネルのOffset欄でスタートするフレームを指定したり、移動にかけるフレーム数を(Curve and Surfaceパネルの)Framesプロパティで指定できます。

しかし、より正確にパス上のオブジェクトの動きをコントロールするには、キーフレームを使うか、パスのEvaluation Time属性に対してSpeedアニメーションカーブを使います。このカーブはパス上の**0.0**(開始地点)から**1.0**(終了地点)までカレントフレームの位置を動かします。

さらなる詳細や例については[Follow Pathコンストレイント](#)のページを参照してください。

The Clamp Toコンストレイント

オブジェクトをパスに沿わせるもう一つの方法は、Clamp Toコンストレイントを使うことです。これはより高度なテクニックを実現しています。パス上のオーナーオブジェクトの位置を決めるために、このコンストレイントは指定した軸方向におけるオーナーの位置を使います。したがって、ターゲットパスに沿った動きをアニメートするためにはオブジェクトの位置を(lpoカーブやその他の間接的アニメーションで)アニメートすることになります

このため、オブジェクトの位置がパスの長さに影響されることはなく、またデフォルトではオブジェクトはパス上のどこかに静止しているということです。

さらなる詳細や例については[Clamp Toコンストレイント](#)のページを参照してください。

Introduction to Physics Simulation

(物理シミュレーションの紹介)

この章では物理現象をシミュレートするために使われることの多い、次のようなさまざまな高度な Blender の効果を扱います:

- 煙
- 雨
- ほこり
- 布地
- 水
- ゼリー

[Particle Systems/パーティクル](#) は多くの事象をシミュレートするのに使われます: 髪の毛、草、煙、毛くず

[Hair \(eng\)](#) はパーティクルシステムの部分集合で、細かな繊維が集まったような物体、例えば髪、毛皮、草、羽柄のようなものに使えます。

[Soft Bodies/柔体](#) は重力や風、他との衝突などの力に対して曲がったり変形したりしようとするもの全般に使えます。皮膚や革の表現の他、布のようなオブジェクトには [Cloth/布地 シミュレーション](#) がありますが、布にも使えます。

[Rigid Bodies/剛体](#) は十分な堅さを持った運動する物体のシミュレートに使えます。

液体と気体を含む [Fluids/流体](#) のシミュレートが可能です。[Smoke/煙](#) のシミュレータもあります。

[Force Fields/力場](#) を使うと他のシミュレーションの挙動を変更できます。

Gravity

(重力)

重力はシーン内にあるすべての物理システムに等しく用いられるグローバルな設定で、シーン/Scene タブで設定できます。デフォルトの Z 軸方向に -9.810 という値は現実世界の重力で、通常はそのまま問題ありません。値を下げる/上げるほど、大きな/小さな重力をシミュレートします。重力は g で表され、大きさは $[m \times s^{-2}]$ で求めます。

重力は **地球** のどこでもほとんど同じです。南極と北極では $g = 9.832 m \times s^{-2}$ で、赤道上では $g = 9.780 m \times s^{-2}$ です。プラハ や ボストン での落下の詳しい計算には 9.81373 を使ってください。月 のシーンのレンダリングでは、6倍小さな値、つまり $1.622 m \times s^{-2}$ を使います。一番人気のおそらく植民地化されていない **火星** では $g = 3.69$ です。

なお Field Weights (剛体フィールドの重み) パネルでは、物理システムごとの重力値の拡大率を設定できます。(訳注: Scene (シーン) コンテキストで Rigid Body World (剛体ワールド) を追加したときの Rigid Body Field Weights (剛体フィールドの重み) を指しているようです。グループごとに重力を決められるようになります)

フォースフィールド (Force Fields)

力場は動的システムに特別な動きを加える方法を提供します。[パーティクル\(Particle\)](#)と[ソフトボディ\(Soft Body\)](#)と[クロス\(Cloth\)オブジェクト](#)は全て力場によって影響されることができます。力場は自動的に全てに影響を与えます。シミュレーションやパーティクルシステムをそれらの作用から取り除くには、単に各々のフォースウェイト(Force Weight)パネルにあるその力場のタイプの作用を下げてください。

- 全てのタイプのオブジェクトとパーティクルが場を生成できますが、カーブオブジェクトだけがカーブガイド(Curve Guides)場を生むことができます。
- 力場はパーティクルからも生成できます。[パーティクル物理\(Particle Physics\)](#)を参照してください。
- 効果を持つために、オブジェクトは少なくとも一つの通常レイヤを共有する必要があります。

あなたはパーティクルでの効果をオブジェクトのグループに制限することができます。[\(パーティクル物理\(Particle Physics\)のページを参照してください\)](#)。

フォースフィールドの作成

モード: オブジェクト (Object) モード

パネル: オブジェクト (Object) コンテキスト → 物理演算 (Physics) サブコンテキスト → フィールド (Fields)

ホットキー: F7

単体の力場を作るには、追加 (Add) » フォースフィールド (Force Field)を選択して、使いたい力場を選択してください。この方法は力場をアタッチしたEmptyを作ります。

既存のオブジェクトから場を作るためには、オブジェクトを選択して物理演算 (Physics)サブコンテキストを変更してください。そして、フィールド (Fields)メニューから場のタイプを選択してください。

場には共通して多くのオプションがあります。これらの一般オプションは球形 (Spherical)場のために規定されます。

ノート

場(フィールド (Fields)パネル)やひずみ(deflection)コリジョン (Collision)の設定を変更した後で、あなたはパーティクル、ソフトボディ、クロスシステムを再計算する必要があります(キャッシュ開放 (Free Cache))。これは自動的にには行われません。あなたはCtrlB → 選択されたキャッシュを開放 (Free cache selected)で全ての選択されたオブジェクトのキャッシュを消すことができます。

パーティクル (Particles)は全ての種類のフォースフィールド (Force Fields)に反応でき、ソフトボディ (Soft Body)は球形 (Spherical)風 (Wind)渦 (Vortex)にのみ反応できます (調和 (Harmonic)場にも反応できますが、役に立つ方法ではありません)。

共通の場設定

大部分の場は、それらがとても異なった行いをするにせよ、同じ設定を持っています。場のタイプ特有の設定は、より下の方で述べます。カーブガイドとテクスチャ場はとても異なったオプションを持っています。

形状 (Shape)

場は、全方向作用のポイント (Point)か、XY面が固定でZ方向だけが変化する平面 (Plane)のどちらかです。

強さ (Strength)

場の効果の強さ。これは力が働く方向を変更するために、正か負にできます。力場の強さはカオブジェクトのスケールによってスケールされます。同じ効果を保って、シーンをスケールアップ・スケールダウンすることが可能です。

フロー (Flow)

効果器の力を気流の速度に変換します。

ノイズ (Noise)

力の強さにノイズを加えます。

シード (Seed)

ランダムノイズの種を変更します。

エフェクトポイント (Effect Point)

パーティクルの位置 (Location)と回転 (Rotation)に場の効果が及ぶかを切り替えられます。

コリジョン吸収 (Collision Absorption)

衝突物によって力が吸収されます。

減衰 (Falloff)

ここでは力場の形状を指定することができます (もし、減衰 (Fall-off)のパワーが0よりも大きいならば)。

球 (Sphere)

球のように、減衰は全ての方向で一様です。

チューブ (Tube)

減衰はチューブ形状の力場に帰着します。

場の最小 (Minimum)と最大 (Maximum)の距離と同様に、場の放射減衰 (Radial falloff)も調節されえます。

円錐 (Cone)

減衰は円錐形状の力場に帰着します。追加のオプションはチューブ (Tube)のオプションと同じです。

Z方向 (Z Direction)

減衰 (Fall-off)はZ軸の正、Z軸の負、またはその両方の方向にのみ適用できます。

パワー (Power)

どのように力場からの距離で力場のパワーが変わるかです。 r がオブジェクトの中心からの距離として、力は $1/r^{\text{Power}}$ で変化します。2の減衰 (Fall-off)は力場を、引力の引き寄せの減衰である $1/r^2$ に変えます。

最大距離 (Max Distance)

力場を、指定した最大半径(オブジェクト周辺の追加円形で表示される)にのみ効果を生じるようにします。

最小距離 (Min Distance)

最大限の強さの効力がある力場の場所までの、オブジェクトの中心からの距離。もし0の減衰 (Fall-off)ならば、このパラメータは何もしません。なぜなら、場は最大距離 (Max Dist) (もしくは無限大)まで、最大限の強さの効力があるからです。オブジェクト周りの追加円形によって表示されます。

場のタイプ

力 (Force)

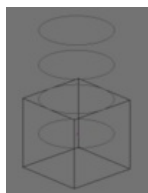
力 (Force)場は場の一番簡単なものです。これはオブジェクトの方へ向かう (正の強さ)、もしくは遠ざかる (負の強さ) 一定の力をもたらします。ニュートン粒子は負の強さの場で引きつけ、正の強さの場で吹き飛ばします。

[ポイド \(Boids\)](#)のために、正の強さの場は *ゴール (Goal)* として使うことができ、負の強さの場は *捕食者 (Predator)* として使うことができます。ポイド (Boids)がゴールや捕食者を探したり逃げたりするかどうかは、ポイド(Boids)の物理演算 (Physics)設定に依ります。



画像 2b: 球形場の表示器

風 (Wind)



画像 3a: 風 (Wind)場の表示器

風は、カオブジェクトのローカルZ軸に沿って、一方向での一定の力をもたらします。力の強さは、表示される複数の円形の間隔によって視覚化されます。

渦 (Vortex Field)

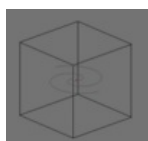


Image 3b: 渦 (Vortex)場の表示器

渦場はカオブジェクトのローカルZ軸の周辺の複数点の方向をよじった、螺旋状の力をもたらします。これは渦巻いた流し台や竜巻、パーティクルヘアでの捻転毛(kinks)を作るのに便利です。

磁石 (Magnetic)

この場はパーティクルの早さに依存します。これは磁化オブジェクトにおける磁力をシミュレートします。

調和 (Harmonic)

力場の源泉が調和振動子(バネ、振り子)のゼロポイントです。減衰 (Damping)パラメータを1に設定した場合、オブジェクトが達した瞬間に動きが止められます。この力場をパーティクルに割り当てた場合、それは本当に独特です。

制止時の長さ (Rest Length)

調和力の制止時の長さをコントロールします。

複数のばね (Multiple Springs)

複数のバネによって影響される全てのポイントで起こります。

普通は場のシステムの全てのパーティクルはターゲットシステムの全てのパーティクルに作用します。調和 (Harmonic)は違います! ここにある全てのターゲットパーティクルは場のパーティクルに割り当てられています。したがってパーティクルは、このように形作りながら、他のパーティクルの場所に動くでしょう。[チュートリアル: 形作るパーティクル \(英文\)](#)。

チャージ (Charge)

これは、電荷(charge)を持つ現実の粒子のように、効果を受けたパーティクルの電荷場(負/正)を元に振る舞い(引き合い/反発)を変えることを除いて、球形場に似ています。これは、この場が、同様にチャージ (Charge)場を持つパーティクルに対する効果のみを持っているということを意味します!(そうでなければ、それらは"電荷"を持っておらず、それゆえ影響されません。)

レナードジョーンズ (Lennard-Jones)

この場は、効果生みと効果受けパーティクルのサイズによって決定される振る舞いを持つ、とても小さな範囲の力です。組み合わせられたサイズより小さい距離では、場はともしりぞけあい、その後引き合う距離となります。それはパーティクルをお互いから平衡距離となるように保とうと試みます。パーティクルは全て、この場によって影響を受けるために、お互いに極めて接近している必要があります。

パーティクルは例えば電荷とレナード・ジョーンズ・ポテンシャルの両方を持つことができます。それは多分、私たちの中の原子核物理学者のためになります。

テクスチャ (Texture field)

複雑な力場を任意に作るために、テクスチャ力場を使うことができます。三方向の力は色で符号化されています。赤はX軸を、緑はY軸を、青はZ軸を符号化します(3Dウィンドウでの座標軸の色と同じように)。0.5の値は力が無いことを意味し、0.5よりも大きい値で負の軸方向(-Zのような)に加速し、0.5よりも小さい値で正の軸方向(+Zのような)に加速します。

テクスチャモード (Texture mode)

これはテクスチャに由来するカベクトルの手法を設定します。

RGB

色成分を直接、方向が符号化された色のカベクトルの成分として使用します。あなたはこのために、画像やカラーバンドなどのRGBテクスチャが必要です。そのため、カラーバンドの無いブレンド (Blend)テクスチャは十分ではないでしょう。

グラデーション (Gradient)

テクスチャの輝度 (グレースケール)の3Dグラデーションとしてカベクトルを計算します。グラデーションベクトルは常に明るさを増加する方向へ向きます。

カール (Curl)

3D RGBテクスチャのカール(RGBベクトルの回転)から、カベクトルを計算します。これもカラーテクスチャと共にのみ働きます。例えばパーリンノイズ(Perlin noise)を使ったカラークラウドテクスチャ(color clouds texture)を使い、見た目の良い乱気流の力を作るのに使用することができます。

ナブラ (Nabla)

グラデーション (Gradient)とカール (Curl)テクスチャモードのために必要とされる部分的な派生を計算するために使われるオフセットです。

オブジェクト座標を使用 (Use Object Coordinates)

パーティクルが使用するテクスチャ空間として放出オブジェクトの座標(及び回転とスケール)を使用します。力場を動かすことが可能となり、オブジェクトの位置座標をそれらの座標に拘束しておけます。

テクスチャ座標のルート (Root Texture Coordinates)

これは、ヘアストランドの全ての部分にパーティクルのルートポジションから計算されるテクスチャ力を使うので、ヘアに便利です。

2D

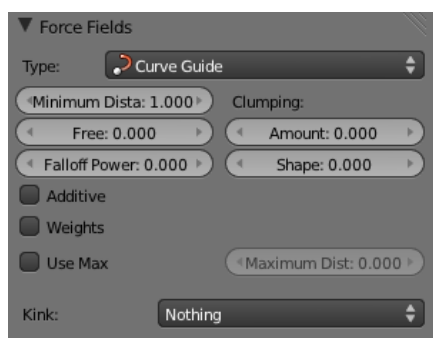
2DボタンはパーティクルのZ座標を無視して、パーティクルのXとYをテクスチャ座標として使用します。

プロシージャテクスチャのみが真の3Dであることを覚えておいてください。

例

- 単色テクスチャの0.5/0.0/0.5はY軸の正の方向への力を作ります。例えば、ヘアはY軸に向きを合わせます。
- カラーバンドを持ったブレンド(blend)テクスチャは力の"平面"を作るのに使われます。例えば、左側が0.5/0.5/0.5で右側が1.0/0.5/0.5なら、XYに対して垂直(すなわちZと平行)に力の平面を持ちます。座標にオブジェクトを使うならば、あなたはパーティクルを周りに押すためにオブジェクトを使うことができます。
- 動作するwood textureはモーションのようなウェーブを作ることができます。

カーブガイド (Curve Guide)



画像 4a: カーブガイド (Curve Guide)場

カーブ (Curve)オブジェクトはカーブガイド (Curve Guide)場の源泉になることができます。あなたは一定のパスに沿ってパーティクルを動かすことができま

す。それらはソフトボディ (Softbody)には影響を与えません。代表的なシナリオは静脈内の赤血球の移動やモーター内での粒子流動の動作付けでしょう。形状固定へアストランドにもカーブガイド (Curve Guide)を使うことができます。もっとも今は**パーティクルモード (Particle Mode)**を持っているため、これは既にそんなに頻繁には使われなくなっているでしょうけども。ソフトボディ(Softbody)としてや他の通常の手法でカーブを動作付けすることができるので、大きな制御を行ないながら、シミュレーション時間を最小に維持し、とても複雑なアニメーションを構築することができます。

カーブ流 (Curve Follow)オプションはパーティクルのためには働きません。代わりに、(パーティクル (Particle)サブコンテキストの物理演算 (Physics)パネルにある)角速度 (Angular Velocity)を設定して、回転を一定のままにする(すなわち、ダイナミック (Dynamic)を有効にしない)必要があります。

カーブガイド (Curve Guide)はカーブからの距離とは独立に、同じレイヤの全てのパーティクルに影響します。複数のガイドを同一レイヤに持っていた場合、それらの場は互いに合計されます (あなたが物理課程で習っているであろう方法)。しかしあなたは作用範囲を制限することができます:

最小距離 (Minimum Distance)

最大限の強さの効力がある力場の場所までの、カーブからの距離。もし0の減衰 (Fall-off)ならば、このパラメータは何もしません。なぜなら、場は最大距離 (MaxDist) (もしくは無限大)まで、最大限の強さの効力があるからです。最小距離 (MinDist)は3Dウィンドウでカーブの端点の円形によって表示されます。

自由 (Free)

パーティクル寿命のfraction。これはカーブには使われていません。

減衰 (Fall-off)

この設定は最小距離 (MinDist)と最大距離 (MaxDist)の間のガイドの強さを管理します。1の減衰 (Fall-off)は線形進行を意味します。

パーティクルはその寿命の間、カーブガイド (Curve Guide)に従います。速度は寿命とパスの長さに依存します。

付加 (Additive)

付加 (Additive)を使った場合、パーティクルの早さも減衰 (Fall-off)に依存して評価されます。

ウェイト (Weights)

カーブのウェイトをカーブに沿ったパーティクル作用への作用に使用します。

最大距離 (Maximum Distance)/最大を使用 (Use Max)

最大の作用半径。カーブオブジェクト周辺の追加円形で表示します。

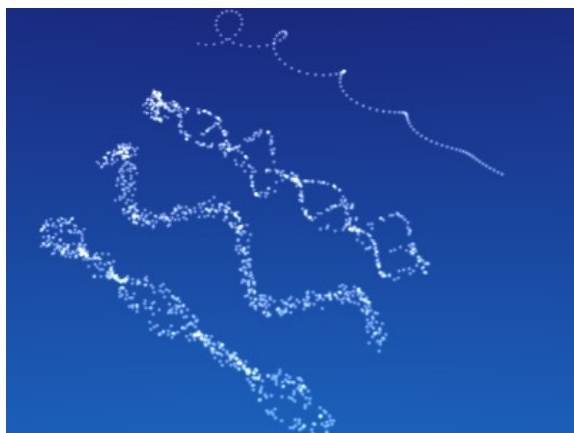
他の設定はカーブに沿った力場の形態を管理します。

集結量 (Clumping Amount)

パーティクルがカーブの最後から一緒に来るか (1)、それらがばらばらに散っていくか (-1)。

シェイプ (Shape)

一緒に来るパーティクルの形態を定義します。+0.99: パーティクルはカーブの最後に接触します。0: カーブに沿って線形進行します。-0.99: パーティクルはカーブの最初に接触します。



画像 4b: カーブガイドのねじれ (Kink)オプション。左から右に: 放射 (Radial)、波 (Wave)、三つ編み (Braid)、ロール (Roll)。

[アニメーション](#)

ねじれ (Kink)のドロップダウンボックスで、力場の形態を変更することができます。

カール (Curl)

放出するカーブの距離に依存した作用の半径。

放射 (Radial)

三次元、定常波。

波 (Wave)

二次元、定常波。

三つ編み (Braid)

編み。

ロール (Roll)

一次元、定常波。

結果の形状を決めることはそんなに簡単ではありませんので、私は十分明確に表示されることを望みます (画像 4b)。

頻度 (Frequency)

オフセットの周波数。

シェイプ (Shape)

開始/終了のオフセットの調整。

振幅 (Amplitude)

オフセットの振幅。

ボイド (Boid)

...

乱流 (Turbulence)

3Dノイズを使って、ランダムな乱流効果を作ります。

サイズ (Size)

ノイズのスケールを指示します。

グローバル (Global)

アタッチしているオブジェクトに代わって、ワールドに相対的なノイズの大きさと強さを作ります。

空気抵抗 (Drag)

抗力(drag)は、パーティクル動作を遅くしそれを阻害するように働く力です。

リニア (Linear)

抗力の成分は速度に比例します。

二次式 (Quadratic)

抗力の成分は速度の二乗に比例します。

リンク

- [Wind & Deflector force update 2.48](#)
- [Particle options and guides \(v2.40\)](#)

衝突 (Collision)

パーティクル、**ソフトボディ**および**クロス**はメッシュオブジェクトと衝突させることができます。また**Boids**はCollisionオブジェクトを避けようとします。

- オブジェクト同士が影響しあうには少なくとも1つのレイヤーを共有している必要があります。
- パーティクルに対する影響を特定のオブジェクトグループに制限することもできます ([Field Weights panel](#))。
- ソフトボディに対するdeflection(ゆがませる)は難しいです。しばしば突き抜けてしまいます。
- Hairパーティクルはdeflectionを無視します(しかしソフトボディとしてアニメーションさせることはできます)

オブジェクトのdeflectionの設定を変えると、パーティクル、ソフトボディ、クロスなどのシミュレーションを(Free Cacheでキャッシュを消去して)再計算しなくてはなりません。これは自動的にには行われません。選択されたすべてのオブジェクトのキャッシュを消去するにはCtrlB → Free cache selectedです。

Mode: Object Mode

Panel: Object context → Physics sub-context → Collision

オプション

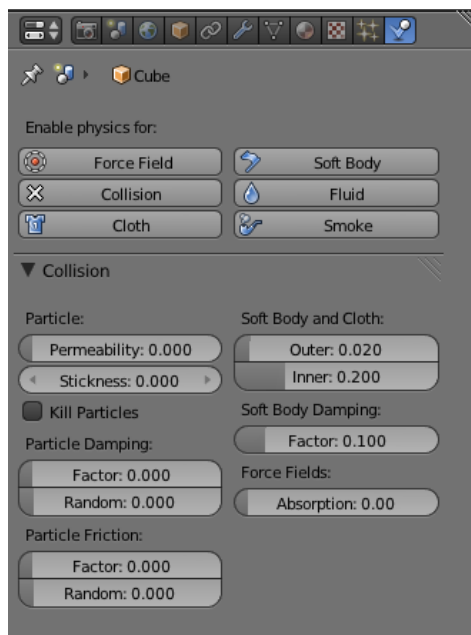


Image 1: Collision panel in the Physics sub-context.

Permeability

メッシュを通過するパーティクルのかけらの割合。IpoのPermチャンネルでアニメーションさせることもできます。

Stickiness

オブジェクトにくっつくパーティクルの量

Kill Particles

衝突時にパーティクルを消します。

Damping Factor

衝突時の減衰(パーティクルの速度とは独立しています)。

Random damping

減衰率をランダムにします。

Friction Factor

オブジェクトの表面を動くときの摩擦。

Random friction

摩擦をランダムにします。

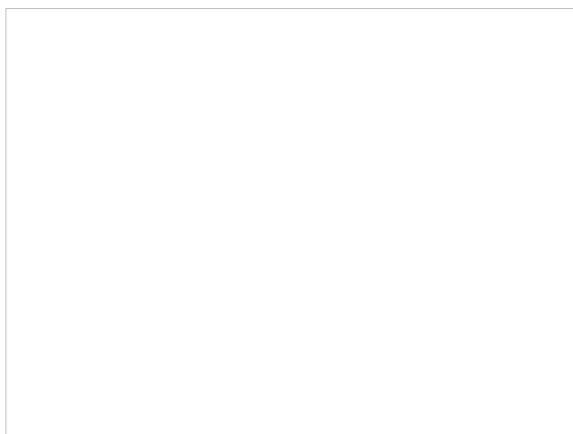


Image 1b: A softbody vertex colliding with a plane.

ソフトボディおよびクロスの衝突

Outer

外側の衝突領域の大きさ。

Inner

内側の衝突領域の大きさ (詰め物の厚み)。

外側と内側は法線で決定されます。図Image 1bの水色の矢印です。

Damping Factor

衝突時の減衰。

ソフトボディの衝突を完璧に行うのは難しいです。ぶつかるオブジェクトが速すぎれば、ソフトボディを突き抜けてしまいます。[ソフトボディ](#)のセクションも見てください。

力場の相互作用

Absorption

deflectorはdeflectエフェクターに作用させることもできます。衝突またはdeflectorオブジェクトを設定して特定のエフェクターの力場をゆがませることができます。Absorptionの値を調整してください。100%にすると力はそのオブジェクトをまったく通り抜けなくなります。たとえばこの値を10%、43%、3%に設定したオブジェクトが縦に並んでいた場合、最終的に通り抜ける力は約50%($100 \times (1-0.1) \times (1-0.43) \times (1-0.03)$)になります。

例



Image 2: Deflected Particles.

これは下向きに放射しているパーティクルシステムにduplivertされたMetaオブジェクトで、立方体メッシュによってゆがめられています。

ヒント

- メッシュの法線が正しい側に向いているように気をつけてください。
- Hairパーティクルは力場に直接的に反応します。よって近い距離で力場を使うなら必ずしも衝突シミュレーションは必要ありません。
- Particle modelにおいて、Hairパーティクルはそれ自体を放射しているメッシュを避けようとしています。よって、少なくともモデリングに関しては衝突のような作用を利用できるということです。

パーティクル(Particles)

パーティクルはオブジェクトメッシュから放射される多量の粒子です。典型的には数千という数になります。個々のパーティクルは光の点であったりメッシュであったり、つながれていたり自由に飛び跳ねたりします。パーティクルは様々な影響や力に反応し、また寿命という概念があります。自由に動き回るパーティクルは炎、煙、霧、そして塵や魔法の力などの表現に使えます。

[ヘア](#) (Hair)タイプのパーティクルは通常のパーティクルを特殊化したものです。ヘアは繊維 (strands) から成り立っていて、髪や動物の毛、草などを表現できます。

パーティクルはモディファイアのパネルとしても存在しますが、すべての設定はObjectコンテキストのParticleサブコンテキストで行います。

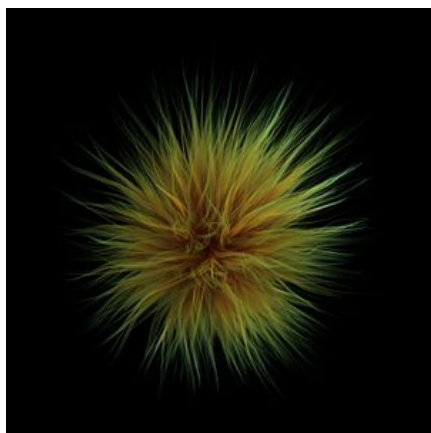


Image 1: Some fur made from particles ([Blend file](#)).

パーティクルは基本的にメッシュ空間に放射されます。その動きは以下のような様々な要素に左右されます。

- メッシュから放出されるとき初期速度。
- エミッター (パーティクルを放射するオブジェクト) の動き。
- 重力や空気抵抗。
- 風、渦、カーブに沿ったガイドなどの力場。
- 他のオブジェクトとの衝突などの相互作用。
- 他のパーティクルとの引力や反発力。
- ソフトボディによる動き (ヘアパーティクルのみ)。
- [ラティス](#) による手動操作。

パーティクルは以下のような形態でレンダリングできます。

- [Halos](#) (炎、煙、雲などに)。
- [メッシュ](#)。そのメッシュそのものをアニメーションすることもできます (魚、蜂など)。この場合、パーティクルはオブジェクトを運ぶ役目ということになります。
- [繊維 \(strands\)](#) (髪、毛、草)。パーティクルの動く道筋が繊維として見えます。この繊維は3D Windowで操作できます (櫛でとかしたり、追加、カット、移動したり)。

どのオブジェクトも、いくつものパーティクルシステムを持つことができます。そしてそれぞれのパーティクルシステムは最大100,000個のパーティクルを含むことができます。パーティクルのタイプによっては (ヘアおよびkeyed) それぞれが最大10,000個の子パーティクルを持つことができます (そのそれぞれが親と似たような動き、放射をします)。具体的な限界はコンピュータのメモリとあなたの忍耐によって決まります。

以前のバージョンとの非互換性

Blender 2.45までで使われていたパーティクルシステムと新しいシステムでは多くの違いがあります。旧システムではできなかったいろいろな事ができるようになりました。新システムは旧システムとは互換性がなく、Blenderは旧システムを変換しようとはしますが完全に上手くいくわけではありません。旧システムは新システムでのエミッターシステム (後述) に似ています。もしもBlender 2.45またはそれ以前のバージョンを使っている場合は、[旧ドキュメントを参照してください](#)。

作業の流れ

通常のパーティクルを使うには以下のような手順を踏みます。

1. パーティクルを放射するメッシュを作成する。
2. そのメッシュに対してパーティクルシステムを1つまたはそれ以上設定する。望みの効果を得るために、複数のパーティクルシステムを相互作用または合成しなければならないことはよくあります。
3. それぞれのパーティクルシステムの設定を整える。
4. ベースメッシュや、シーンに含まれるほかのパーティクルメッシュをアニメートする。
5. パーティクルの軌跡や流れを成形する。
6. ヘアパーティクルの場合はカットしたり櫛でとかしたりして整える。
7. 最終的なレンダリングやシミュレーションを行い、必要に応じて手直します。

パーティクルシステムの作成

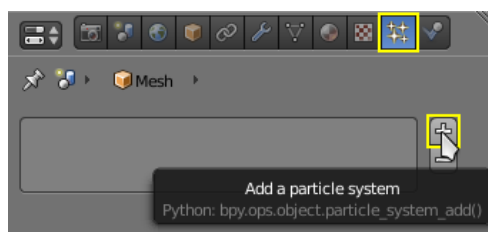


Image 2: Adding a particle system.

オブジェクトに新しいパーティクルシステムを追加するには、Object Settings EditorのParticlesタブで +ボタンを押します。一つのオブジェクトは複数のパーティクルシステムを持つことができます。

それぞれのパーティクルシステムは互いに独立した設定を持っています。それを共通させることもできるので、手作業で設定値をコピーしなくてもすみます。また、複数のオブジェクト間で設定値を共有することもできます。Randomの項目を使ってそれぞれの見た目を少しずつ変えれば、単純なコピーには見えなくなります。

パーティクルシステムの種類

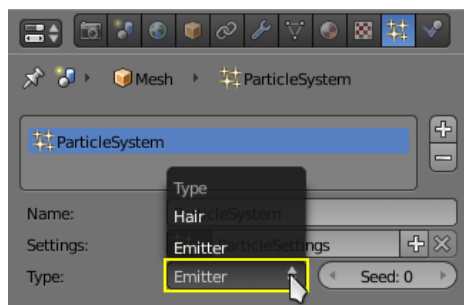


Image 3: Particle system types.

パーティクルシステムを作成したら、Propertyウィンドウにたくさんのパネルやボタンが現れます。しかしあわてないように！ パーティクルシステムは2つのタイプがあり、Typeのドロップダウンリストで切り替えられます。

エミッター (Emitter)

旧パーティクルシステムをさらに拡張したようなものです。設定した開始フレームから終了フレームまでの間、オブジェクトの表面からパーティクルが放射され、それぞれのパーティクルは寿命を持っています。

ヘア (Hair)

このタイプのパーティクルは繊維 (strand) としてレンダリングされ、非常に独特の特徴を持っています。3D Windowでリアルタイムに編集したり、クロスシミュレーションを併用してアニメートしたりできます。

パーティクルシステムのタイプによって設定項目は部分的に異なります。たとえばImage 3ではエミッターだけの設定が表示されています。

共通オプション

それぞれのシステムは共通したコントロールセクションを持っています。ただしその中の項目はシステムによって変わります。

放射 (Emission)	パーティクルの初期状態やシーンへの表れ方を設定します。
キャッシュ (Cache)	リアルタイムな反応性を上げるため、また無駄な再計算を避けるため、パーティクルのデータをメモリ内にキャッシュ、またはディスクに蓄積できます。
速度 (Velocity)	パーティクルの初期速度。
回転 (Rotation)	パーティクルの回転に関する設定。
物理 (Physics)	パーティクルの動き方の種類を決めます。
レンダー (Render)	レンダリングの設定。
表示 (Display)	3D Windowでの表示。
子 (Children)	子パーティクルの生成に関するコントロール。
フィールドウェイト (Field Weights)	外部からの力に関する設定。
力場 (Force Field Settings)	パーティクルを力場にします。
頂点グループ (Vertex Groups)	頂点グループによって様々な影響を与えます。

リンク

- [Tutorials](#)
- [Physics Caching and Baking](#)
- [Particle Rewrite Documentation](#)
- [Thoughts about the particle rewrite code](#)
- [Static Particle Fur Library](#)

柔体 (Soft Body) シミュレーション



Image 1a: A softbody cloth uncovering a text. [Animation](#) – [Blend file](#)

柔体 (Soft Body) は、柔らかい、または変形する物体のシミュレーションです。Blenderにおいて、このシステムは布でできた物および閉じたメッシュに適しています。べらべらした布切れに対しては、別のエンジンを使った専用の[クロス](#)シミュレーションがより適しています。

シミュレーションはオブジェクトの頂点や制御点に対して力を作用させることで行われます。重力や力場といった外部の力、および頂点同士をつなぎとめようとする内部の力です。このようにして、何かで満たされた物体が力に反応した場合にとる形状をリアルに再現できるのです。

柔体は他のオブジェクトと衝突させることができます (*Collision*)。自分自身と衝突することもできます (*Self Collision*)。

柔体シミュレーションの結果は静的なオブジェクトに変換できます。また、シミュレーションの結果を編集 (*bake edit*) することもできます。たとえば途中のあたりの結果を編集して、そこからまたシミュレーションをすることもできます。

典型的な使い方



Image 1b: A wind cone. The cone is a Soft Body, as the suspension. [Animation](#) – [Blend file](#)

柔体は以下のようなものに適しています。

- 伸び縮みする物体。衝突をする場合もしない場合も。
- 力に反応する旗や布切れ。
- ある種のモデリング作業。たとえばクッションやテーブルクロスを作るとき。
- Blenderには衣服をシミュレートするのに適した別のシステム ([クロス](#)を参照) がありますが、幅の広い袖などの特定の部分には柔体を使うこともできます。
- 髪 (衝突を最小限にすれば)。
- 揺れるロープ、鎖などのアニメーション。

以下のビデオを見れば使い道のアイデアがもっとわくかもしれません: [\[1\]](#), [\[2\]](#)

柔体の作成

柔体シミュレーションは頂点または制御点をもつすべての種類のオブジェクトで行えます。

- メッシュ
- カーブ
- サーフィス
- ラティス

オブジェクトを柔体としてシミュレーションを起動するには:

- PropertiesウィンドウのPhysicsタブ (一番右の、ボールが弾んでいるアイコン) に移動します。
- Soft Bodyのボタンを押します。

するとたくさんの設定項目が現れます。すべての項目のリファレンスは[ここを参照](#)。

- シミュレーションはAltAで始められます。
- Spaceでポーズ、AltAで再開。
- Escで中止します。

シミュレーション品質

Soft Body Solverパネルでシミュレーションの正確さを設定できます。

Min Step

各フレームに対するシミュレーションの最小ステップ数。動きの速い衝突オブジェクトが柔体をすり抜けてしまうようなときは数値を大きくしてください。

Max Step

各フレームに対するシミュレーションの最大ステップ数。通常、シミュレーションのステップ数は(Error Limitに基づいて)動的に変わりますが、この値を変えたほうが良い場合もあります。

Auto-Step

ステップサイズを自動的に変えます。

Error Limit

シミュレーションの全体的な品質。デフォルトは0.1です。これは衝突の作用をどのくらい正確にチェックするかという最も重要な項目です。まずは平均的な辺の長さの半分の値にしてください。そして目に見える不都合、乱れ、大げさすぎる反応などが発生した場合は、値を下げます。Error Limitが同じなら、シミュレーションはどんなときでも大体同じくらいの「ひどさ」で行われます。値を下げると、より最適なステップサイズになります。

Fuzzy

シミュレーションの計算が速くなりますが、正確さは下がります。

Choke

且衝突オブジェクトのメッシュ内に進入した頂点や辺を落ち着かせます(出て行く速度を下げます)。

Diagnostics

Print Performance to Console

シミュレーションの動作をコンソールに出力します。

Estimate Matrix

行列を推定してCOM、ROT、SCALEに分離します。

キャッシュとバイク

柔体や他の物理シミュレーションは統一的なキャッシュおよびバイクのシステムを持っています。詳しくは[パーティクルキャッシュ \(eng\)](#)を参照してください。

アニメーションをプレビューすると、シミュレーションの結果は自動的にディスクにキャッシュされ、次にアニメーションするときには保存したデータを読み取ってすばやく動作します。シミュレーションをバイク(Bake)した場合はそのキャッシュは保護され、再計算が必要になるような変更を加える際はたずねられます。



開始時間と終了時間に注意

シミュレーションはBakeパネルのStartとEndで指定したフレームの範囲だけで行われます。バイク以外のときもです！ よってデフォルトの250フレームよりも長くシミュレーションしたい場合はEndの値を変える必要があります。

- キャッシュ:
 - アニメーションが始まると、それぞれの物理シミュレーションシステムが、その開始時間から終了時間までのフレームをディスクに書き込みます。これらのファイルは.blendファイルと同じ場所に“blendcache”の接頭語をつけて保存されます。
 - 変更が加えられた場合キャッシュは消去されますが、すべての変更の際ではありません。よって手動で開放しなければならない場合もあります。たとえば、カ場の設定を変えたときなどです。再度キャッシュする際には、シミュレーションの開始時間よりも前の(または同じ)フレームからアニメーションさせなければならないことに注意してください。
 - ディスクに対する書き込み権限がないときはキャッシュはされません。
 - キャッシュの開放は各シミュレーションシステムごとのパネルから行えます。またCtrlBで、選択したすべてのオブジェクトのキャッシュを消去できます。
 - .blendファイルの名前がとても長くて、かつオペレーティングシステムでファイル名の長さ制限がある場合、問題が発生する可能性があります。
- バイク:
 - バイクすると、シミュレーションの結果はプロテクトされ、変更はできなくなります。
 - CtrlBで、選択したすべてのオブジェクトのバイク結果を消去できます。また現在のオブジェクトの柔体シミュレーションについてはFree Bakeで開放できます。
 - メッシュ形状を変更してもシミュレーションの再計算はされません。

レンダーファームでレンダリングするときは、すべての物理シミュレーションをバイクして、blendcacheファイルをすべてレンダーファームに送るのが良いでしょう。

リアルタイムなインタラクション

柔体シミュレーションの作業をするときには、Timelineウィンドウが便利だと気づくでしょう。目的のフレームに簡単に移動でき、つねにそのときのフレームのシミュレーションが表示されます。TimelineウィンドウのPlaybackメニューからContinue Physicsのオプションを選択すると、シミュレーション中にオブジェクトにかかわることができます。たとえばオブジェクトを動かして衝突させたり、柔体を揺らしたりできます。とても楽しいですよ！



Continue PhysicsはAltAでアニメーションしているときは働きません。

この機能はTimelineウィンドウでPlayボタンを押したときだけ使えます。

シミュレーションの途中のフレームで柔体オブジェクトを選択し、EditingコンテキストでモディファイアをApplyすると、そのときのオブジェクトの形状が定着します。

ヒント

- 柔体シミュレーションはオブジェクトの頂点が均等に分布していると特に上手く働きます。また衝突を高品質でシミュレートするには十分な数の頂点が必要です。オブジェクトのある部分に頂点を追加すると、その部分の変形のしかたが変わります（硬くなります）。*Image 1b*のアニメーションを参照。
- 衝突の計算は長い時間がかかることがあります。見えない部分は計算しないようにしましょう。
- 衝突の計算を速くするためにしばしば便利なのは、単純で透明で比較的大きな補助オブジェクトを作って、実際にはそれに衝突させることです（*Image 1a*の例を参照）。
- 柔体を使うのは、その必要があるときだけにしましょう。ぴったりした布で表面を覆った物体を単独でアニメーションさせる場合などは意味がありません。柔体の髪に自己衝突を適用することはできますが、これはややこしいことになるでしょう。それについては後の[柔体-衝突 \(eng\)](#)で扱います。
- オブジェクトそのものではなく、それに付加したLatticeやCurve Guideを柔体にすることも試してみましょう。計算が物凄く速くなるかもしれません。

リンク

- [Developer Notes](#)
- [Swinging of a chain](#)
- [Softbodies for Rigged Characters](#)

Simple examples

(簡単な例)

柔体物理のパワーが見られる簡単な例です。

bouncing cube

(はねるキューブ)

開始、終了フレームを 1 と 150 に変えてください。

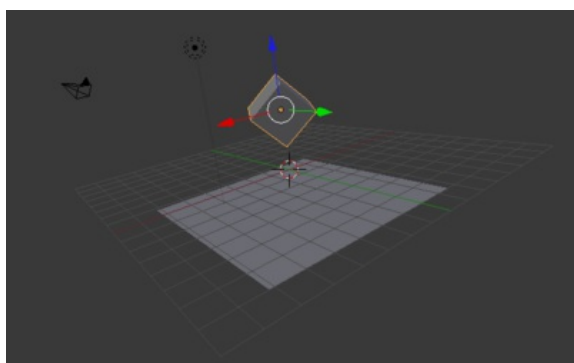


タイムライン

平面を追加して5倍に拡大します。Physics (物理) タブを開き、Collision (衝突) を追加します。今回はデフォルトの設定で構いません。

キューブを追加するか、デフォルトのキューブを使います。Tab で Edit Mode に入って3回 Subdivide (細分化) します。Bevel モディファイアを追加して角を滑らかにします。少し傾けるため、r を2回押してマウスカーソルを少し動かします。

ここまで進めると、シーンは次のようになるでしょう:



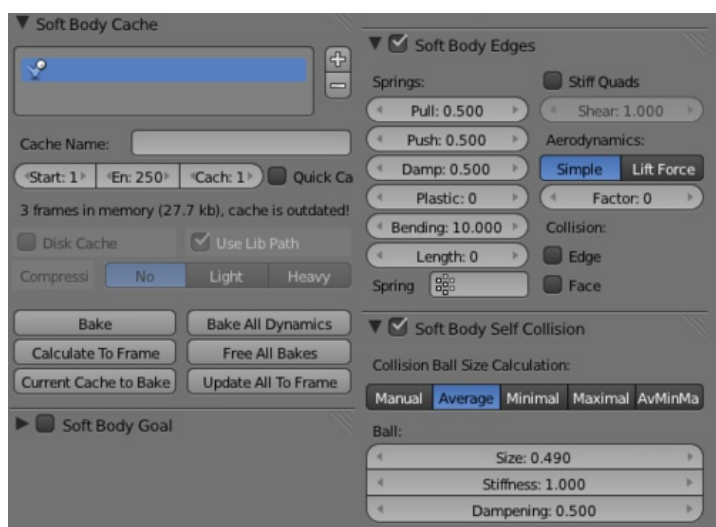
柔体物理の準備ができたシーン

柔体物理を追加する準備はばっちりです。物理タブを開き Soft Body (柔体) を追加します。「Soft Body Goal (ソフトボディゴール)」のチェックを外し、「Soft Body Self Collision (ソフトボディのセルフコリジョン)」にチェックを入れます。「Soft Body Edges (ソフトボディエッジ)」パネル内で Bending (曲げ) を 10 に増やします。

Alt+a でアニメーションを再生すると、はねるキューブがゆっくりアニメーションします。速くするには、柔体物理の Bake (焼き込み) を行う必要があります。

「Soft Body Cache (ソフトボディのキャッシュ)」パネル内の開始/終了をアニメーションの開始/終了フレームにあわせませす。今回は 1 と 150 です。動作テスト用に Cache Step (キャッシュステップ) を 5 か 10 にすることができますが、最終的には 1 にして、すべてをキャッシュさせたほうがよいでしょう。

ここまでできれば、物理パネルが次のようになるはずで:



物理設定

これで、シミュレーションの焼き込みを行なって、キューブに材料とテクスチャを与え、アニメーションをレンダリングすることができるようになりました。

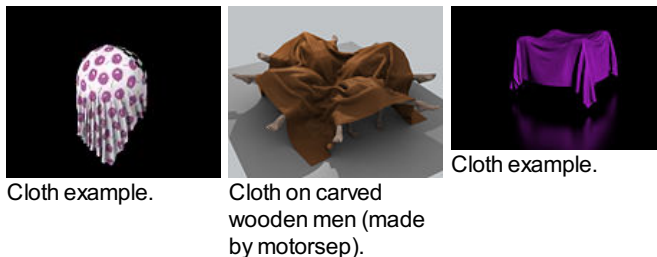
result

(結果)

レンダリングされたはねるキューブです:

[\[video link\]](#)

クロス(Cloth)シミュレーション



Cloth example.

Cloth on carved wooden men (made by motorsep).

Cloth example.

クロス(布)シミュレーションはCGの表現の中でも難関の一つです。布は現実の世界ではありふれた素材のようで、実際はその内部や環境との間で非常に複雑な相互作用が働いているのです。何年もの開発の末にBlenderは非常に強力なクロスシミュレーターを手に入れ、衣服や旗の再現ができるようになりました。クロスはあなたのコントロールするオブジェクトや風や力場および他の空力モデルと関わって、その影響を受けます。

解説

クロスとして設定されれば、どんなメッシュでもクロスとして振舞います。Physicsサブコンテキストに三つのパネルで構成されたClothパネルがあります。クロスは開いたメッシュでも閉じたメッシュでもよく、また質量がゼロでもいいし、単位面積あたりの質量を持たせることもできます。

クロスは一般的にグリッドまたは立方体としてモデル化されますが、たとえばテディベアでもいいのです。しかしながら、閉じたメッシュであれば[ソフトボディ](#)を使うほうがよりうまくシミュレートできます。クロスはべらべらした布地のシミュレーションに向いています。

オブジェクトがクロスとして設定されると、クロス[モディファイア](#)がオブジェクトのモディファイアスタックに追加されます。モディファイアは他のモディファイア、たとえばArmatureやSmoothと相互作用します。この場合、メッシュの最終的な形状はモディファイアスタックにおけるモディファイアの順序に則って計算されます。たとえば、クロスを滑らかにするのならその形状が計算されたあとにしなくてはなりません。

よってクロスは二つの場所で編集することになります。F7のPhysicsボタンでクロスの様々な性質を調整し、モディファイアスタックでは画面での表示や他のモディファイアとの関係を調整します。

クロスモディファイアをApplyして、クロスをそのフレームでの形状に固定することもできます。そうするとモディファイアは削除されます。たとえば、平らなクロスをテーブルにかがせて、シミュレーションを行い、それからモディファイアをApplyします。この場合はモデリングのためにシミュレーターを使ったということになるでしょう。

シミュレーションの結果はキャッシュに保存されます。よって一度計算されたメッシュ形状のアニメーションは再計算する必要はありません。シミュレーションの設定を変えた場合は、キャッシュの消去と再計算を自由にできます。一度目のシミュレーションは完全に自動的に行われ、バイクや余計なステップは含まれません。

各フレームでのクロス形状の計算はバックグラウンドで自動的に行われるので、その計算中も作業を続けられます。しかしこれは非常にCPUを使う仕事で、あなたのコンピュータのパワーとシミュレーションの複雑さにかかっています。CPUのパワーがたくさんいる計算の場合は描画の遅れに気づくでしょう。

先回りしないように

クロスシミュレーションをセットアップし、しかしまだシミュレーションの終わりまで計算されていない場合、先のほうのフレームにジャンプするとクロス形状が正しく計算されないかもしれません。それ以前のフレームがまだ計算されていないからです。

作業の流れ

クロスに関する作業は以下のような手順になります。

1. クロスの初期形状をモデリングします。
2. PropertiesのPhysicsタブで、オブジェクトをクロスとして設定します。
3. クロスと相互作用するdeflectionオブジェクトをモデリングします。Deflectionモディファイアがモディファイアスタックの最後、つまり他のモディファイアによる変形等よりも後になるように注意してください。
4. クロスに照明をあて、マテリアルやテクスチャの設定をします。必要ならUV展開をします。
5. 必要ならパーティクルを追加します。たとえば布の表面から蒸気が出るような場合。
6. シミュレーションを実行し、満足な結果になるように設定を調整します。timelineウインドウのVCRコントローラが役に立つでしょう。
7. 場合によっては、シミュレーションのある時点でのクロス形状を初期形状として再度シミュレーションします。
8. フレームごとに細かな手直しをします。

クロスシミュレーションの生成

このセクションでは望みの結果になるようにどう設定すればいいかを紹介합니다。まず、クロスを有効にして、再現したいクロスの種類に応じてセットアップします。プリセットから手始めとなるものを選びます。

設定してみるとおわかりのように、布地が重くなるほど、硬くなり、伸びにくく、空気の影響が少なくなります。

クロスパネル

Presets

いくつかのクロスの例が用意されています。新しく付け加えることもできます。

Quality

1フレームごとのシミュレーションのステップです。数値を高くすると品質は上がりますが、計算が遅くなります。

素材

Mass

クロス素材の質量。

Structural

全体的な硬さ。

Bending

しわの係数。数値が高いほど大きなひだができます。

減衰

Spring

クロスの速度の減衰。高いほど滑らかで、あまり揺れなくなります。

Air

空気は通常、物が落下する速度を和らげる働きがあります。

ピン止め



Cloth in action.

クロスをピン止めるのにまず必要なのは**頂点グループ**です。これにはいくつか方法があり、ウェイトペイントを使ってピン止めたい部分にペイントするのもやり方の一つです([ウェイトペイント](#)を参照)。

頂点グループを設定したら、あとは簡単です。ClothパネルのPinning of clothボタンを押し、使用する頂点グループとその点での硬さを設定します。

Stiffness

ターゲットの点での硬さ。数値はデフォルトの1のままで良いでしょう。

衝突

大抵の場合、クロスはたんに3D空間に垂れ下がっているだけではなく、環境内の他のオブジェクトと衝突します。適切なシミュレーションをするためにいくつか設定・注意しなければならないことがあります。


1. クロスオブジェクトをCollision(衝突シミュレーション)に参加するように設定します。
2. オプションとして(しかし推奨)、クロス自身とも衝突するように設定します。
3. 他のオブジェクトはメッシュオブジェクトでなければいけません。
4. 他のオブジェクトは移動したり、さらに他のオブジェクトの影響(アーマチュアやシェイプキーなど)で変形したりしてもかまいません。
5. 他のオブジェクトはクロスをdeflect(ゆがめる)する設定をします。
6. .blendファイルを保存してください。そうするとシミュレーションの結果も保存されます。
7. それではシミュレーションをBakeしましょう。アニメーション範囲の各フレームでの形状が計算されます。
8. その後、各フレームでの形状を編集できます。
9. 環境や他のオブジェクトを再調整して、特定のフレーム以降を再シミュレーションすることもできます。

衝突の設定

Cloth Collisions panel.

今度はクロスオブジェクトを衝突シミュレーションに参加させなければいけません。右のようなCloth Collisionパネルを探してください。

Enable Collisions

この項目をLMB クリックすると衝突が有効になります。

Quality

シミュレーションの全体的な品質。数値を高くすると計算に時間がかかりますが、物体がクロスを突き抜けてしまったりすることが少なくなります。

Distance

他のオブジェクトがこの距離よりも近づくと、クロスを押しのけます。

Repel

衝突しそうになったときの斥力。

Repel Distance

斥力が働く最大距離。最小値よりも大きくする必要があります。

Friction

摩擦効果。メッシュオブジェクトとぶつかったときの滑りにくさです。たとえば、絹は綿よりも摩擦が低いです。

クロス自身との衝突

現実の布は自分自身をすり抜けたりはしません。よって通常は自己衝突の設定が必要でしょう。

Enable Self Collisions

これをクリックすると、クロスは自分自身を突き抜けなくなります。これはシミュレーションの時間が増えますが、結果がよりリアルになります。遠くから見た旗などにはこの設定は必要ないでしょうが、人物のケープやブラウスを近くから見るときには有効にするべきでしょう。

Quality

数値を高くすると、衝突の計算がより多く繰り返され、品質が上がります。少なくともCollision Qualityと同じかそれ以上にする必要があります。





Distance

何か問題があったらMin Distanceの数値を変えてみるというかもしれません。たいていは0.75あたりが最適ですが、動きが速い場合は1.0にしたほうがいいでしょう。0.5まで下げるのは非常に危険です(すり抜けてしまうことが多々あるでしょう)が、計算のスピードアップにもなります。

サンプルファイル: [Cloth selfcollisions](#).

共有されたレイヤー

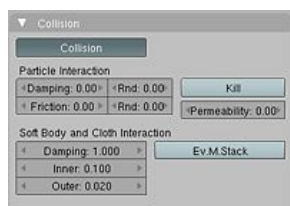
たとえば、レイヤー2と3にズボンが、レイヤー1と2に人物が存在している場合を考えてみてください。ズボンにはすでにクロスの設定がしてあります。クロスにとって人物が「見え」て要る必要があります。そうすれば、人物の足が曲がる時、その動きはクロスを押しのけます。大事なことは、他のシミュレーションと同じく、相互作用は共有されているレイヤーのみで起こるということです。この例の場合、双方のオブジェクトはレイヤー2を共有しています。

オブジェクトのレイヤーを変えるには、3D ViewのObject ModeでオブジェクトをRMB クリックして選択します。そしてMでMove Layersポップアップを表示させると、そのオブジェクトが存在しているレイヤーがわかります。オブジェクトを1つのレイヤーに移動するにはレイヤーのボタンをLMB クリックしてください。複数のレイヤーに移動するには⇧ Shift LMB で選びます。あるレイヤーからオブジェクトを取り除きたい場合は、⇧ Shift LMB で再度クリックします。

メッシュオブジェクトの衝突

衝突に使いたいのがメッシュオブジェクトではないとき、たとえばNURBSサーフィスやテキストオブジェクトの場合、メッシュに変換する必要があります。Object Modeでオブジェクトを選択し、3D viewのヘッダからObject → Convert Object Type (AltC)を選び、ポップアップメニューからMeshを選びます。

クロスとオブジェクトの衝突



Collision settings.

クロスは他のオブジェクトによってゆがまないといけません。クロスをゆがめるには、オブジェクトをクロスと衝突させるように設定する必要があります。衝突するオブジェクト(クロスオブジェクトのほうではありません)のdeflectionを有効にしてください。

ButtonsウィンドウのObjectコンテキスト、その中のPhysicsサブコンテキストで、右図のようなCollisionパネルを見つけてください。このパネルで設定をすると、共有しているレイヤー内の他のオブジェクト(パーティクル、ソフトボディ、クロス)に対しても衝突やdeflectionが起こることに注意してください。

注意

CollisionパネルはPhysicsサブコンテキスト内に三種類あります。一つ目はFieldsパネルのそばにあるもので、これがいまここで必要なパネルです。二つ目はSoft Bodyグループにあるタブで、これはソフトボディに関係するものです(よってクロスに関する限り見る必要はありません)。そして最後にCloth panelのそばのタブです。これは、さきほど設定に使いました。

メッシュオブジェクトのモディファイアスタック



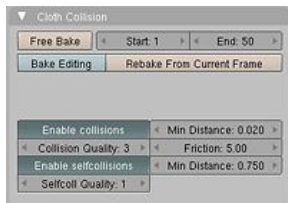
Collision stack.

オブジェクトの形状がクロスを変形させるためには、そのフレームでのオブジェクトの「本当の」形状がわからなければシミュレーションできません。この本当の形状とは基本となる形状がシェイプキーやアーマチュアによって変形された状態のもので、したがって、Collisionモディファイアはこれらのモディファイアよりもあとに位置しなければいけません。右図は人物のメッシュオブジェクトのモディファイアです(クロスオブジェクトではありません)。

クロスのキャッシュ

キャッシュの設定は他の物理シミュレーションと同様です。詳しくは[パーティクルキャッシュ](#)を見てください。

衝突をバイクする



シミュレーションにかかわるフレームでの他のオブジェクトの設定(アーマチュアによるメッシュの変形も含めて)が終わったら、衝突の計算を始められます。クロスを選択し、ObjectコンテキストのPhysicsサブコンテキストで、StartとEndを設定してシミュレーションするフレーム範囲を決め、Bakeボタンを押してください。

バイクしたシミュレーションを消去するまではStartやEndは変えられません。シミュレーションが終わると、開放、編集、再バイクのボタンが現れるでしょう。

ここですぐに気づくことがいくつかあるでしょう。まず、以前よりもバイクが非常に遅いということ、それからおそらく右図のように箱を突き抜けてしまう部分があるでしょう。

シミュレーションのキャッシュを編集する

キャッシュには各フレームでのメッシュ形状が含まれています。シミュレーションをバイクしたあとBake Editingボタンを押すと、キャッシュされた形状を編集できます。編集したいフレームに移動し、⇨ TabでEdit modeに移動します。そしてメッシュ編集ツールをなんでも使って頂点を動かしてください。Edit modeから出ると、アニメーションのフレームとしてそのメッシュ形状が記録されます。そのフレーム以降を再シミュレートしたい場合はRebake from next Frameを押してアニメーションを再生してください。Blenderは編集後のメッシュ形状を使って続きをシミュレートしてくれます。

メッシュを編集して、オブジェクトが突き抜けてしまった部分などを手直ししましょう。

メッシュの頂点を追加、消去、押し出し、取り外し、などとすると、1フレーム目から現在のフレームまでメッシュが置き換えられます。よって、Edit modeから出るときにはキャッシュの保護を解除して消去してください。そうすればちゃんとつながりのあるシミュレーションになります。

トラブルシューティング

衝突検出に関して問題にぶつかったときは、修正する方法は二つあります。

- 手っ取り早いのはCloth CollisionパネルのMin Distanceを大きくすることです。これが一番手軽に「つきぬけ」を修正できます。しかしながら、正確さは低下するので見た目はあまりよくないでしょう。このやり方をすると、クロスは空気の上にふんわりと乗っているような感じになります。
- もう一つの方法はQuality (Clothパネルにあります)を上げることです。そうするとより細かいステップでシミュレーションが行われ、すばやい動きの衝突も高い精度で再現できるようになります。また、Collision Qualityを上げることもできます。衝突の計算を反復する回数が上がります。
- これらの方法で解決できないときは、キャッシュまたはバイクされた結果をEdit modeで編集できます。
- クロスがオブジェクトに切り裂かれてしまうという場合は、ClothパネルのStructStiffを非常に高く、1000くらいにしてみてください。

Subsurfモディファイア

バイクやキャッシュは異なるsubsurfレベルに応じてそれぞれ行われるので、3D Viewとレンダリング時のsubsurfレベルは同じにしておいてください。

例

クロスシミュレーションをするには、もちろん布が必要です。よってデフォルトの立方体を消去して平面を追加してください。私はY軸に沿って拡大してみました、そうしなくてもかまいません。きれいにべらべらふにやふにやとさせるにはメッシュを何度か再分割する必要があります。私は8回再分割しました。⇨ Tabを押してEdit modeに入り、W → Subdivide multiで8に設定してみましょ。

では、このオブジェクトをクロスにするために、Objectコンテキスト(F7) → Physicsサブコンテキストに移動しましょう。スクロールしてClothパネルを見つけ、Clothボタンを押してください。するといろいろな設定値が現れるでしょう。しかし今回はこれらのほとんどは扱いません。

これでクロスをアニメーションさせる準備はおしまいです。しかし、AltAを押してみても、あなたのかわいい布切れはまったくおもしろくない感じに落下するだけでしょ。これをなんとかするのが次の2つのセクションで説明するピン止めと衝突です。

成形のためにシミュレーションを使う

どのフレームでも、ClothモディファイアをApplyしてそのときのメッシュを固定化することができます。そして再度クロスを有効にし、最初と最後のフレームを設定してシミュレーションすることもできます。

旗を例にしてみましょう。単純なグリッドをポールにピン止めて、50フレームほどシミュレーションしてみると、旗は自然なポジションに落ち着いて止まります。その状態でクロスモディファイアをApplyします。そしてシーンの中ではためかせたくなったら再度クロスを有効にして、旗がカメラに写っている時間をシミュレーションのフレーム範囲に設定すればいいのです。

クロスを手軽にする

さて、前のセクションから続けて読まれているなら、あなたのクロスは少しカクカクしているかもしれません。それを手軽にするには、Editingコンテキスト (`{Shortcut{f9}}`) のModifiers/パネルで、SmoothやSubsurfモディファイアを使ってください。そして、同じコンテキストでLinks and Materialsを見つけ (頂点グループのために使ったのと同じパネルです)、Set Smoothを押してください。

それではAltAを押してみてください。すばらしい結果になるはずです。

アーマチュアとクロス

アーマチュアやよって変形されるクロス、および他のオブジェクトとの衝突のサンプルファイル: [Regression blend file](#)

クロスを手軽ボディとして使う



Using cloth for softbodies.

クロスはソフトボディを再現することにも使えます。これが主な目的ではないのは確かですが、ちゃんと働きます。この例では標準のRubberプリセットを使っています。設定は変えず、AltAを押すだけです。

写真の例のサンプルファイル: [Using Cloth for softbodies](#)

クロスと風



Flag with wind applied.

風と衝突のシミュレーションのサンプルファイル: [Cloth flag with wind and selfcollisions.](#)

流体 (Fluid) シミュレーション

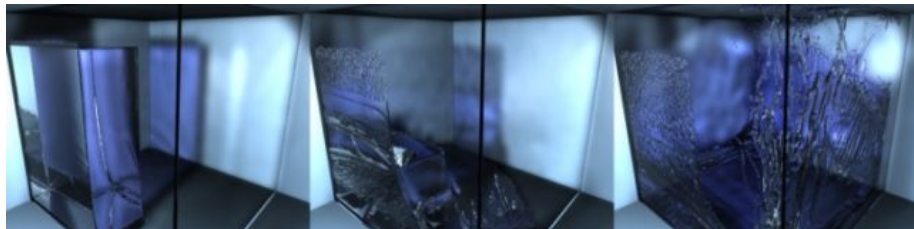
Mode: Object mode / Edit mode (Mesh)

Panel: Physics sub-context → Fluid

解説

Blenderでシーンを制作するとき、指定したオブジェクトを流体シミュレーションに参加させることができます。たとえば流体や障害物としてです。また別のオブジェクトのバウンディングボックスはシミュレーションの行われる領域として使われます (これはシミュレーションドメインと呼ばれます)。シミュレーション全体にかかわるパラメータ (たとえば速度や重力) もこのドメインオブジェクトに対して設定できます。

バイク (BAKE) ボタンを押すと、形状やセッティングの情報がシミュレーターに伝えられ、流体シミュレーションが実行されます。そして各フレームにおけるメッシュとそのプレビューが生成され、ディスクに保存されます。その後、表示しているフレームに応じて適切なデータが読み込まれ、表示またはレンダリングできます。



A breaking dam.

作業の流れ

基本的に、以下のような手順を踏みます。

- [シミュレーションドメイン](#) (シーンの中で流体の存在する領域) を設定する。
- [流体となるオブジェクト](#) を設定し、マテリアル、粘度、初期速度を指定する。
- 場合により、[流体をコントロールするオブジェクト](#) を設定する (発生源や吸収源)。
- 場合により、流体と関係するオブジェクトを設定する。
 - [障害物](#)
 - 流体に浮かぶ [パーティクル](#)
 - 流体を目的の形状に変形させる [コントロール](#)
- 場合により、[流体のプロパティをアニメートする](#)
- [シミュレーションをバイクする](#) (必要に応じて手直しし、再度バイクする)。



ドメインオブジェクトにバイクされます !

流体シミュレーションを計算すると、シミュレーションはドメインオブジェクトに対してバイクされます。

したがって:

- バイクに関するオプションはドメインオブジェクトが選択されているときのみ表示されます。
- バイクに関するオプションはドメインのページの [バイクのセクション](#) で説明されています。

さらに詳しく

Blenderの流体シミュレーションについてより詳しく知るには、以下のページを参照してください。

- シミュレーションについての [ヒント](#)。
- よりリアルなシミュレーションを行うのに役立つ [技術的詳細](#)。
- 制限や問題の解決法を知るには [付録](#)。また追加のリンクも紹介されています。

流体ドメイン

ドメイン (Domain) オブジェクト

このオブジェクトのバウンディングボックスがシミュレーションの境界として使われます。**流体オブジェクトはすべてこの中になければいけません**。ドメインの外の流体はバイクされません。ほんのしずくでもドメインの中から外には出られません。見えない壁に囲まれているような感じです。シーンの中にはドメインはひとつだけ存在できます。

ドメインはつねに箱型として扱われるので、**オブジェクトの形状は関係ありません** (バウンディングボックスの寸法は関係ありますが)。よって、通常は箱型以外の形状にする理由はありません。他の形状の障害物や境界を使いたい場合は、このドメインの内部にそれを作って配置してください。

このオブジェクトはシミュレーションの際に流体に置き換えられます。

ドメインオブジェクトにバイクされます！

流体シミュレーションを計算すると、シミュレーションはドメインオブジェクトに対してバイクされます。したがって、バイクに関するオプションはドメインオブジェクトが選択されているときのみ表示されます。

バイクに関するオプションについてはこのページの[バイクのセクション](#)を参照してください。

オプション



The fluid simulation options with Domain selected

Bake ボタン

バイクに関するオプションについてはこのページの[バイクのセクション](#)を参照してください。

Resolution

Render resolution

シミュレーションする際の細かさです。これはおそらく最も重要な設定項目でしょう。この値によって流体の細部の緻密さ、メモリとディスクの使用量、計算時間が決まります。

10cm mug at Resolution 70.

10cm mug at Resolution 200.

数値を上げると使用メモリは急激に増えるので注意してください。解像度が32では約4MB、64で約30MB、128では230MB以上になります。クラッシュやフリーズを避けるため、コンピュータのメモリの量に対して十分に低い解像度にとどめるように気をつけてください。また覚えておかなくてはいけないのは、多くのオペレーティングシステムでは1つのプロセス("Blender"など)が使えるメモリの量を制限していることです。実際にはもっと搭載していてもです。自分のコンピュータにおける制限を調べておきましょう。

解像度とドメインのサイズ

現実世界におけるドメインの大きさにふさわしい解像度に行きましょう (現実世界での大きさについては[Domain World/パネル](#)を参照)。ドメインが立方体でない場合、解像度は最も長い辺に対して計算されます。他の辺に対しては長さに応じて解像度が下がります (つまり、解像度が同じなら立方体のほうが使用メモリは少なくなるということです)。

Preview resolution

生成されるプレビューメッシュの解像度です。よってこれはシミュレーションそのものには影響しません。一見何も見えなくても、プレビューの解像度では見えないだけで実際にはわずかな流体が存在しているという場合もあります。

Display quality

バイクしたシミュレーションを3Dビューで表示 (*Viewport Display*)、およびレンダリング (*Render Display*) するときの描画の種類。

- *Geometry*: オブジェクト本来 (シミュレーションする前) の形状を使います。
- *Preview*: オブジェクト本来の形状を使います。
- *Final*: 最終的な高精度のメッシュを使います。

バイクデータが見つからない場合、デフォルトで本来の形状が使われます。

ドメインをバイクすると、3Dビューには (通常は) プレビューメッシュが表示されます。本来のドメイン形状を見たい場合はドロップダウンリストから *Geometry* を選んでください。

Time

Start

シミュレーションの始まる時間 (秒)。

この値を変えることで、シミュレーションの計算を途中の時間から始めることができます。その時間以前のドメイン変形や流体の動きは保存されません。

たとえば、シミュレーションが始まって4秒たった時点から実際のフレームが始まってほしい場合、4.0と入力します。

End

シミュレーションが終わる時間 (秒)。



StartとEndはバイクされるフレーム数とはまったく関係ありません

Startを3.0、Endを4.0に設定したとすると、1秒間ぶんのシミュレーションが行われます。この1秒間が、フレーム1からアニメーション終了フレームまで引き伸ばされます。つまり、たとえばBlenderを25fpsで250フレーム再生する設定にしていた場合 (Sceneコンテキスト → Renderサブコンテキスト → AnimおよびOutputパネル)、すでに3秒間経っているような状態で流体のシミュレーションが始まり、そこからスローモーション (通常の1/10のスピード) で流れます。1秒間のシミュレーションが10秒間のビデオとして生成されるからです。これを正しくするには、終了時間を $3.0 + 10.0 = 13.0$ として、250フレームを25fpsで再生したときの秒数と合わせます。これでシミュレーションは現実どおりの時間で流れます。10秒間のシミュレーションが10秒間のアニメーションとして生成されるわけです。このようにしてシミュレーションのスピードをコントロールできます。

Generate Speed Vector

このボタンを押すと、速度ベクトルが生成されません。デフォルトでは速度ベクトルが生成されディスクに保存されます。このベクトルはコンポジティングノードでモーショントラッカーをかける際に使用されます。

Reverse fluid frames

シミュレーションが逆向きに計算されます。

Bake directory

バイクに関するオプションについてはこのページの[バイクのセクション](#)を参照してください。

Domain World



The Domain World options.

Viscosity (粘度)

流体の「とろみ」のようなものです。具体的には、ある表面上である速度でオブジェクトを動かすのに必要な力の大きさです。数値を入力することもできるし、プリセット (はちみつ、油、水など) から選ぶこともできます。

自分で入力する場合に覚えておくべきこととして、通常、現実世界の粘度 (いわゆる動的粘性率 dynamic viscosity) はパスカル秒 (Pa·s) またはポアズ (P, 1P=0.1Pa·s、粘性流体の層流に関する法則を発見したフランスの物理学者ジャン＝ルイ・ポアズイユに因んだ単位)、また一般的にはセンチポアズ (cP, 1cP=0.001Pa·s) という単位で測られます。一方Blenderでは、動粘度 kinematic viscosity (動的粘性率 Pa·sを密度 $\text{kg}\cdot\text{m}^{-3}$ で割ったもの、単位 $\text{m}^2\cdot\text{s}^{-1}$) という指標を使っています。下記の表ではいくつかの流体の動的粘性率および動粘度を示しています。

入力は浮動小数点実数と指数で行います。このような方法 (科学的表記法) は非常に小さな数値から非常に大きな数値まで簡単に表せます。室温の水の粘度は1.002 cP、すなわち0.001002 Pa·sで、密度は約1000 $\text{kg}\cdot\text{m}^{-3}$ です。ここから動粘度を計算すると0.000001002 $\text{m}^2\cdot\text{s}^{-1}$ 、すなわち1.002に10の6乗をかけたもの (科学的表記では 1.002×10^{-6}) となります。溶けたガラスや鉄も流体ですが、粘度は非常に高いです。動粘度 $1.0\times 10^0 (=1.0)$ 。また動的粘性率でいえば 1.0×10^6 cPくらいを入力しないといけないでしょう。

注意してほしいのは、このシミュレーターは非流体、つまり「流れる」ということがない物体には向いていないということです。単純に非常に大きな粘度を入力しても剛体の振る舞いをするわけではなく、不安定になるだけでしょう。

粘度は変化します

Blenderのプリセットの粘度は、それぞれの流体に典型的で、アニメートしたときにそれらしく見えると思われる値を設定してあります。しかし、現実にはいくつかの流体、たとえばチョコレートシロップやはちみつのように糖分を含んだ流体などは、温度と濃度に応じて粘度が変わります。オイルの粘度はSAE値(米国自動車技術者協会による粘度分類)によって様々です。室温のガラスは基本的に固体ですが、1500Kではほぼ水のように流れます。

Blender Viscosity Unit Conversion

Realworld-size	Fluid	dynamic viscosity (in cP)	kinematic viscosity (Blender, in $m^2 \cdot s^{-1}$)
現実世界	Water (20°C)	1.002×10^0 (1.002)	1.002×10^{-6} (0.000001002)
におけるドメインの大きさ	Oil SAE 50	5.0×10^2 (500)	5.0×10^{-5} (0.00005)
(メートル)。	Honey (20°C)	1.0×10^4 (10,000)	2.0×10^{-3} (0.002)
コーヒー	Chocolate Syrup	3.0×10^4 (30,000)	3.0×10^{-3} (0.003)
カップなどを作るなら10cm (0.1m)、スイミングプールだったら10mなどでしょう。この長さはドメインの一番長い辺の長さとして計算されます。	Ketchup	1.0×10^5 (100,000)	1.0×10^{-1} (0.1)
	Melting Glass	1.0×10^{15}	1.0×10^0 (1.0)

この長さはドメインの一番長い辺の長さとして計算されます。

Optimization

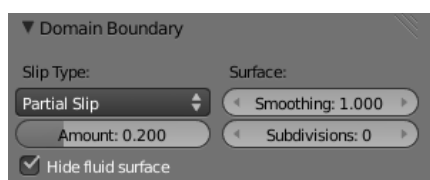
Gridlevel

シミュレーションでいくつの適応的グリッドレベルが使われるかを決めます。-1にしておくとも自動的に選択されます。

Compressibility

大きくて縦に長い流体の領域を高解像度でシミュレーションする際に問題が起こる場合は、この値を下げる方がいいかもしれません(そうすると計算時間は増すので注意してください)。

Domain Boundary



The Domain Boundary panel

ここですべり具合と滑らかさの設定をします。

Boundary type

障害物の表面のねばつき、「表面の粘着性 (Surface Adhesion)」を決めます。現実の世界では、粘着性は流体と、物体表面の粒状度、粘着度、弾力によって決まります。

No Slip

流体がびたりとくっきます(速度0)。

Free Slip

物体の上で流体が動きます(法線方向の速度は0)。

Part Slip

二つの中間です。0ではほぼNoslip、1ではFreeとまったく同じです。

Surface

Surface Smoothing

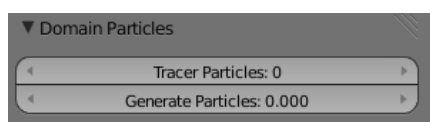
流体の表面のスムーズ化の度合いです。1.0が標準、0が無効です。大きくするほど滑らかになります。

Subdivisions

シミュレーション内で直接的に高解像度に再分割された表面を生成します(逆に、モディファイアなどを使うとシミュレーションの後で再分割することになります)。1は再分割をしない設定で、数値をあげるごとに流体のボクセルが一段階ずつ再分割されます。その結果メッシュデータは大きくなり、必要なディスクスペースも増します。あまり大きな値にすると表面を生成するのに時間がかかるので注意してください。

Hide fluid surface

Domain Particles



The Domain Particles panel

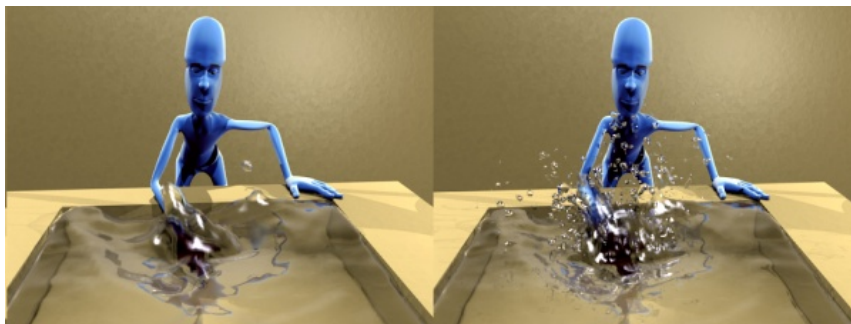
視覚的効果を高めるために、シミュレートした流体にパーティクルを加えることができます。

Tracer Particles

シミュレーションする際に流体の中に配置するトレーサーパーティクルの数。これを表示するには後述のパーティクルタイプのオブジェクトを作成します。このバイクにはドメインと同じディレクトリが使われます。

Generate Particles

生成する流体パーティクルの量(0=無効、1=標準、>1=それ以上)。これを使うにはSubdivisionsを2以上にしてください。



An example of the effect of particles can be seen here - the image to the left was simulated without, and the right one with particles and subdivision enabled.

バイク



The fluid simulation options with Domain selected

バイクボタン (BAKE)

シミュレーションを実行します。Blenderは通常通りに動き続けますが、ウィンドウの上部、レンダリングメニューの隣に、ステータスバーが表示されます。その横でEscまたはXを押すとシミュレーションを中止できます。シミュレーションが終了すると、指定したディレクトリ内に2つの“.bobj.gz”ファイル(プレビューオリティと最終クオリティ)と1つの“.bvel.gz”ファイル(最終クオリティ)が各フレームに対して作られます。

バイクディレクトリ

必須!

バイクしたメッシュを保存するためのディレクトリおよびファイルの接頭語です。

アニメーション出力の設定に似ていますが、ファイルの選択の仕方だけ少し違います。以前に生成したメッシュ(たとえば“untitled_OBcube_fluidsurface_final_0132.bobj.gz”)を選択した場合、接頭語(ここでは“untitled_OBcube_”)の設定に自動的に反映されます。このようにしてシミュレーションを違った設定で何度か行い、そのデータ群を簡単に切り替えられます。

デフォルトのディレクトリ“/tmp/”はおそらく望ましくないでしょう。適切なディレクトリ名およびファイル名接頭語を設定すると、それに応じた場所とファイル名でデータが保存されます。たとえば二つのシミュレーションデータが互いにぶつかったりせずすみずみ(ディレクトリ名だけを指定したい場合は、最後に“/”をつけるのを忘れないようにしてください)。

注意

ドメインは1つだけ

シーン同士の間にもたがったりリンクしたりする可能性のため、存在できるドメインはひとつのblendファイルにつきひとつだけです。

バイクドメインを選択する

バイクしたドメインは流体メッシュに変わります。変更を加えて再度バイクするためにドメインを選択したい場合は、どのフレームでもいいのでRMBで流体メッシュを選択してください。そしてバイクボタンを押せば再計算が始まります。

バイクはつねにフレーム1から始まります

流体シミュレーションはAnimパネルのSta設定に関係なく常に1フレーム目から始まります。シミュレーションをもっとあとのフレームから始めたいときは、ドメインの中で流体のアニメーションキーを記録して、目的のフレームまで静止させておく

必要があります。詳しくは[流体のプロパティをアニメートする](#)を参照。

ベイクはつねにAnimパネルのEndで設定されたフレームで終わります

仮にフレームレートが25fpsで、シミュレーションの終了時間が4.0秒(そして開始時間は0.0 秒)の場合、アニメーションは $4.0 \times 25 = 100$ フレーム目で終わるようにしてください。

以前のベイクデータを開放する

ベイクディレクトリの中身を削除するという破壊的な手段でこれを実行できます。二つ以上のシミュレーション結果が同じディレクトリ内にあるときは注意してください(違うファイル名を使っていることを確認してください。そうでないとどんどん上書きされます)。

ベイクデータを再使用する

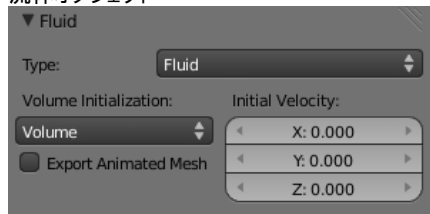
以前にベイクに使ったディレクトリおよびファイル名を入力すれば、シミュレーションの内容を切り替えられます。このようにして以前のベイクデータをまた使うこともできます。

ベイクにかかる時間

ベイクには「多大な」演算パワー(および時間)を使います。シーンによっては、ベイクは夜中にかけて行うのがいいでしょう。

メッシュにモディファイアが設定されている場合、流体の計算にはレンダリング用の設定値が使われます。その設定によっては計算時間とメモリは指数関数的に増大します。たとえばSubsurfされたメッシュを動く障害物に使う場合、Subsurfを無効または低いレベルにすると計算時間を減らす役に立ちます。正しく準備すれば、リアルな結果を生み出す方法はいつだって他にもあるものです。

流体オブジェクト



Fluid object settings

ドメイン内におけるこのオブジェクトの領域が、流体として使われることになります。現在のところ、ドメインの中に二つ以上の流体を作る場合は流体同士が重なってはいけません。また、法線が外側を向いていることを確かめてください。ドメインとは違って、このオブジェクトのメッシュ形状はそのまま流体の形状になります。

Volume initialization type

- Volume: オブジェクトの内部がすべて流体として初期化されます。これは閉じたメッシュの場合だけ機能します。
- Shell: メッシュの表面が薄い流体のレイヤーとして初期化されます。これは開いたメッシュでも使うことができます。
- Both: VolumeとShellを足し合わせたような状態です。これも閉じたメッシュでなければいけません。下図を参照。



Example of the different volume init types: Volume, Shell and Both (ShellはVolumeよりもわずかに大きい)

Animated Mesh/Export

メッシュがアニメートする場合(アーマチュア、シェイプキー、ラティスでの変形等)、このボタンを押してください。これを有効にすると計算がとても遅くなるので注意してください。また位置や回転のIPOでアニメートする場合(つまりオブジェクトの座標変換だけをする場合)は必要ありません。

Initial velocity

シミュレーションの最初の時点での流体の速度。単位はメートル/秒です。



法線の向きによって結果が大きく変わります！

Blenderは面の法線の方角を使って流体の「内側」と「外側」を決めます。すべての法線が外側を向いているようにしてください(Edit modeでCtrlN。またはSpaceのあとEdit → Normals → Calculate Outside)。法線が間違った方向を向いていると、あなたは激しい洪水に襲われるでしょう。なぜならBlenderはオブジェクトの存在する領域が「外側」であると考えからです。これは流体のVolume initのタイプにかかわらず起こります。

障害物(Obstacle)

このオブジェクトはシミュレーション内で障害物として使われます。流体オブジェクトと同様、現在のところ障害物も互いに重なってはいけません。また元のメッシュ形状が使われます。Volumeタイプの場合、法線が正しい方向を向いていることを確かめてください(Edit modeで、Mesh Toolsパネル→Editingコンテキスト [F9]からFlip Normalを使ってください)。とくに回転押し出しで作ったメッシュの場合は注意してください。メッシュ形状がアニメーションしない場合はバイクする前にSubSurfモディファイアをApplyしておくのもいいアイデアです。

Volume initialization type

- Volume: オブジェクトの内部がすべて流体として初期化されます。これは閉じたメッシュの場合だけ機能します。
- Shell: メッシュの表面が薄い流体のレイヤーとして初期化されます。これは開いたメッシュでも使うことができます。
- Both: VolumeとShellを足し合わせたような状態です。これも閉じたメッシュでなければいけません。下図を参照。



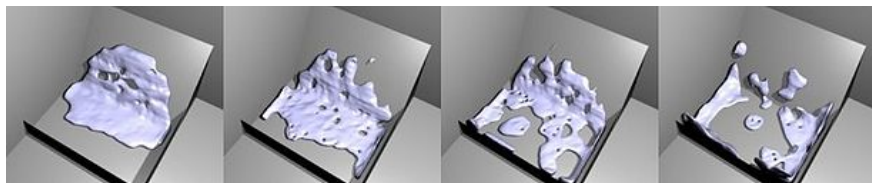
Example of the different volume init types: Volume, Shell and Both (ShellはVolumeよりもわずかに大きい)

Boundary type

障害物の表面のねばつき、「表面の粘着性(Surface Adhesion)」を決めます。現実の世界では、粘着性は流体と、物体表面の粒状度、粘着度、弾力によって決まります。

- Noslip: 流体がびたりとくっつきます(速度0)。
- Free(-slip): 物体の上で流体が動きます(法線方向の速度は0)。
- Part(-slip): 二つの中間です。0ではほぼNoslip、1ではFreeとまったく同じです。

メッシュが動いている場合は自動的にNoslipとして扱われることに注意してください。



Example of the different boundary types for a drop falling onto the slanted wall. From left to right: no-slip, part-slip 0.3, part-slip 0.7 and free-slip.

Animated Mesh/Export

メッシュがアニメートする場合(アーマチュア、シェイプキー、ラティスでの変形等)、このボタンを押してください。これを有効にすると計算がとても遅くなるので注意してください。また位置や回転のIPOでアニメートする場合(つまりオブジェクトの座標変換だけをする場合)は必要ありません。

PartSlip Amount

前述の、NoslipとFreeのミックスされる量。

動く障害物がサポートされました

Blenderは動く障害物をサポートするようになりました。

以前は、障害物が動いている場合は自動的にNoslip(くっつく)として扱われました。よって流体が動くオブジェクトにぶつかってはね返ってほしいときは、障害物の表面の流体がぶつかる位置に、ぴったりと、しかしほんの少しだけ手前に浮いているように透明な面をおく、というテクニックを使わなければいけませんでした。現在ではそれをする必要はありません。

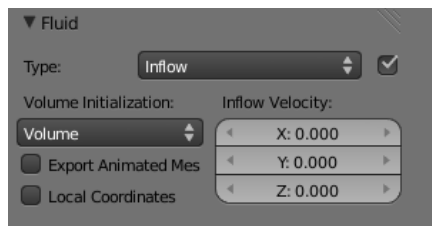
Impact Factor

動くオブジェクトとの衝突によって流体が増える／減る量を決めます。オブジェクトが動いていない場合はこの設定は何の効果もありません。しかし動いている場合は、流体がぶつかったとき、数値がマイナスならドメイン内から流体の体積が減り、プラスの数値なら体積が増えます。数値の範囲は-2.0から10.0です。

体積のコントロール

流体の量をコントロールするため、**ドメイン**の中に流体を追加または吸収するオブジェクトを設置することができます。

発生源 (Inflow)



Fluid inflow settings

このオブジェクトはシミュレーション内に流体を注ぎます。給水栓のようなものです。

Volume initialization type

- Volume: オブジェクトの内部がすべて流体として初期化されます。これは閉じたメッシュの場合だけ機能します。
- Shell: メッシュの表面が薄い流体のレイヤーとして初期化されます。これは開いたメッシュでも使うことができます。
- Both: VolumeとShellを足し合わせたような状態です。これも閉じたメッシュでなければいけません。下図を参照。



Example of the different volume init types: Volume, Shell and Both (ShellはVolumeよりもわずかに大きい)

Inflow velocity

オブジェクトの内部で流体が生成される速度です。

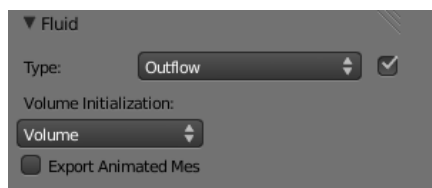
Local Coords/Enable

流体の発生にローカル座標系を使います。これはオブジェクトが移動・回転している場合に便利です。供給の流れはそれにしたがって動きをコピーします。これを無効にした場合は発生の位置や方向は変わりません。

Animated Mesh/Export

メッシュがアニメートする場合（アーマチュア、シェイプキー、ラティスでの変形等）、このボタンを押してください。これを有効にすると計算がとても遅くなるので注意してください。また位置や回転のIPOでアニメートする場合（つまりオブジェクトの座標変換だけをする場合）は必要ありません。

吸収源 (Outflow)



Fluid outflow settings

このオブジェクトの領域に流れ込んだ流体はすべて消去されます（排水口やブラックホールのようなものです）。発生源オブジェクトと一緒に使えば、ドメインが流体でいっぱいにならずにすみます。有効にされると、水上の竜巻や液体用掃除機のように流体を吸い込み、オブジェクトの移動にしたがって流体の消える位置も移動します。

Volume initialization type

- Volume: オブジェクトの内部がすべて流体として初期化されます。これは閉じたメッシュの場合だけ機能します。
- Shell: メッシュの表面が薄い流体のレイヤーとして初期化されます。これは開いたメッシュでも使うことができます。
- Both: VolumeとShellを足し合わせたような状態です。これも閉じたメッシュでなければいけません。下図を参照。

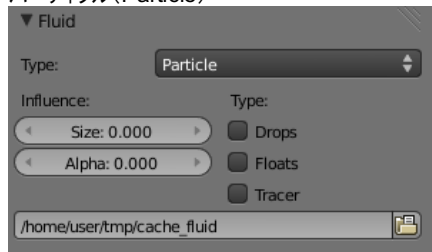


Example of the different volume init types: Volume, Shell and Both (ShellはVolumeよりもわずかに大きい)

Animated Mesh/Export

メッシュがアニメートする場合(アーマチュア、シェイプキー、ラティスでの変形等)、このボタンを押してください。これを有効にすると計算がとても遅くなるので注意してください。また位置や回転のIPOでアニメートする場合(つまりオブジェクトの座標変換だけをする場合)は必要ありません。

パーティクル(Particle)



Fluid particle settings

このタイプのオブジェクトはシミュレーションで生まれたパーティクルを表示するのに使われます。現在のところは流体と一緒に流れるトレーサーのみサポートされています。オブジェクトの形状や位置や種類は何でもかまいません。パーティクルボタンを押すと、シミュレーションのパーティクルシステムが適切な位置に作成されます。元のオブジェクトを動かした場合は、一度パーティクルシステムを消去する必要があるかもしれません。そしてfluidsimパーティクルを無効にし、再度有効にしてください。また、現在のところfluidsimパーティクルは通常のパーティクル用の力場や設定には反応しません。

Influence

Size Influence

パーティクルはばらばらな大きさにすることもできます。値が0ではすべて同じ大きさになります。

Alpha Influence

値が0より大きい場合、パーティクルのアルファ値はそのサイズによって変わります。

Particle type

Drops

表面ではねた流体は飛び散るしぶきようになります。真水のように表面張力の弱い流体の場合です。

Floats

冷たい海水やスープのように、表面張力が強く、流体がより重い場合です。Dropsと比べると表面から離れたしぶきは大きく、表面に戻るのも早いです。

Tracer

しぶきは流体の表面があった位置についていきます。さっきまでの水面の位置に浮かんでいる霧のような感じです。

Path (バイクディレクトリ)

使うべきパーティクルデータの場所です。これは通常はドメインオブジェクトの場合と同じ場所にします (CtrlC、CtrlVでコピーしてください)。

コントロールオブジェクト

解説

格子ボルツマン法を使い、流体はパーティクルでコントロールすることができます。パーティクルはローカルな力場を定義し、物理シミュレーションまたはターゲットシェイプによって自動的に生成されます。同時に、流体の自然な動きはできるだけ維持されます。

[\[video link\]](#)

例

この例では、Fluid Controlオプションを使って流体の一部をコントロールし、残りの流体に合流するまで特定の形(球形またはティーポット型のしずく)をもたせています。

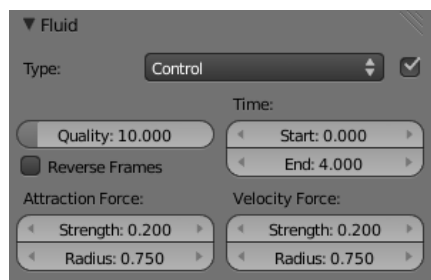


Falling drop (rendered in Yafaray)



"Magic Fluid Control"

オプション



Fluid control options.

Quality

品質を高くすると流体コントロールオブジェクトのパーティクルがより多く作られます。

Reverse Frames

コントロールパーティクルの動きを反転します。

Time

コントロールオブジェクトを有効にする時間の範囲を指定します。

Attraction force

コントロールオブジェクトから放射される力を設定します。プラスの数値では流体を引き寄せ、マイナスでは退けます。


Velocity force

コントロールオブジェクトが動いた場合、その速度を流体に作用する力とすることもできます。

関連項目

リリースノート: [Template:Release_Notes/2.48/FluidControl](#)

流体のプロパティをアニメートする

ドメインオブジェクトに対し、新しいタイプのIpoカーブFluidSimが使えます。Blenderでアニメート可能なほかの値とは違い、FluidSimはたんにを押しただけではキーフレームを作れません。Ipoウィンドウをクリックして手動で値をセットする必要があります。アニメートしたいプロパティをIpoウィンドウで選択し、Ctrl LMB をクリックで望みの場所にキーフレームを追加してください。



プロパティの入力

正確な位置をクリックする必要はありません。まずコントロールポイントを作り、それからTransform Propertiesパネルを開いてN、フレーム位置としてX値を入力し、つぎに好きなY値を入力すれば良いです。

流体ドメインはいくつかのコントロールチャンネルを持っています。

Fac-Visc

流体の粘性係数に掛け合わされる数値です。これはベイクの前に設定する必要があります。流体の粘度を時間とともに変化させ、たとえば水をワイ.....いや、油に変えたりできます。

Fac-Tim

シミュレーションの速度を変えられます。Video Sequence Editorがビデオをスピードアップ/ダウンできるのに似て、この値のカーブで流体の動きをスピードアップ/ダウンできます。値が0より小さいときは、シミュレーション内の時間(および流体)は止まります。0.0から1.0の場合は流体はゆっくり動き、通常よりも粘りがあるように見えます。1.0は通常の流体の動きです。それよりも大きな数値になると、流体の動きは速くなり、また粘度が下がったように見えるかもしれません。

GravX/GravY/GravZ

XYZそれぞれの方向の重力、言い換えれば流体にかかる慣性です(レーシングカーを運転しながらコーヒーを飲んでいる状況、あるいはアウトバーンでエスプレッソをすすする光景、もしくはスペースシャトルの中で花に水をやることを想像してください)。この要素を変化させると流体は外部からの力のためにぼしゃぼしゃとはねます。

流体、障害物、発生源、吸収源、およびパーティクルオブジェクトでは以下のチャンネルが使えます。

VelX/VelY/VelZ

ホースから噴き出す水などは、このチャンネルを使って圧力と方向を変化させれば再現できます。また水の流れに対して風が吹いている様子などもシミュレートできるでしょう。

Active

この値が0.0から大きくなると(たとえば0.1から1.0くらいに)、オブジェクトの機能(たとえば発生、吸収など)が復旧します。値が0まで下がり、またあるところまで上がると、また効果が表れます。水滴や、あらゆるムラのある流れに利用してください。このチャンネルは吸収源と障害物にも使えるので、流れが詰まっているような効果もシミュレートできます。

コントロールオブジェクトの力も操作できます。

AttrForceStr, AttrForceRa

Attraction forceの設定を変化させます。

VelForceStr, VelForceRa

Velocity forceの設定を変化させます。

ヒント

Blenderで流体シミュレーションをするときに役立つヒントです:

- シミュレーションが終わると大量のメッシュファイル(.bobj.gz)が出来上がりますが、驚かないように。1つのセットはプレビュー、もう1セットは最終クオリティのものです。それぞれのセットにはアニメーションの各フレームに対する.gzファイルが含まれ、そのそれぞれにシミュレーションの内容が入っています。つまり、全部必要なファイルです。
- 現在のところこれらのファイルは自動的に消去されません。よってシミュレーション結果を保存する専用のディレクトリを作ったほうが良いでしょう。流体シミュレーションをするのはANIMボタンを押すのに似ています。出来上がったファイルは自分でどこかのディレクトリに管理しておかなくてはなりません。流体シミュレーションを使うのをやめるときは、単純に流体の.bobj.gzファイルを削除すれば良いです。
- 何時間もかかるような高解像度のシミュレーションをする前に、低い解像度でシミュレートしてみて、すべてのタイミングがうまくあっているかをチェックしましょう。
- 流体オブジェクトはドメインオブジェクトのバウンディングボックス内に完全に収まっていないといけません。さもないと、ベイクは不正確になるか、または完全に失敗します。流体や障害物は複雑な構造のメッシュになることがあります。非常に小さなオブジェクトは、それを表現するには荒すぎる解像度だとシミュレーションに表れないかもしれません(解像度を上げるとこの問題は解決するかもしれません)。
- 発生速度やアクティブフラグなどのシミュレーションパラメータはFluidsimpoでアニメートできることを覚えておいてください(先述部分参照)。
- 複雑なシーンを一度のシミュレーションで再現しようとししないでください。Blenderは強力な編集機能があるので複数のアニメーションを合成することができます。

たとえば、二つの分離した流体が登場するアニメーションを小さなドメインで作るには、まず一つの流体だけで.aviをレンダリングします。それからドメインを動かしてもう一つの流体でアルファチャンネルを使って.aviファイルをレンダリングします。それから、Compositorの合成機能を使って二つの.aviを合成します。3つめのaviはすべての上に合成される煙や霧などでしょう。その上に雨を追加すれば、嵐を表現できるでしょう。そして稲妻や吹き飛ばされるごみ、などをすべて別々のアニメーションとして作り、それから合成すれば迫力のある結果になるのです。

- もしも問題が起きたり何かが思ったとおりにいかない場合は、私に知らせてください。blendファイルと問題の説明をnils at thurey dot deに送ってください。メールを送る前にwikiのページやblenderartists-forumをチェックしてね。

技術的詳細

物理的正確さ



“My cup runneth over”, created with Blender and Yafray.

流体アニメーションは制作に非常に時間がかかりますが、内部の仕組みをよく理解するほど、仕上がりを予想しやすくなるでしょう。Blenderの流体シミュレーションに使われるアルゴリズムは格子ボルツマン法(Lattice Boltzmann Method、LBM)といわれるものです。流体アルゴリズムには他にもナビエ-ストークス(Navier-Stokes、NS)エンジンや、平滑化粒子流体力学(Smoothed Particle Hydrodynamics、SPH)などがあります。LBMはこの二つの中間のような方法です。

一般的にいて、現在のコンピュータでは1メートルのタンクの水ですら正確にシミュレートするのは非常に困難です。街の中を暴れまわる波をシミュレートなどしようと思ったら、手に入る最も高価なスーパーコンピュータが必要で、それでも不足かもしれません。上に挙げたどのアルゴリズムを使ってもです。したがって、本当に思ったとおりの効果を生み出すためには、考え方を変えてみる必要があります。映画製作者たちがアナログ時代から(非常に効果的に)やってきたのと似た考え方、つまり、「それっぽくでっちあげよう！」ということです。

満足できるイメージを作り上げるためには、品質の良い流体シミュレーションが非常に重要です(それがすべてではありませんが)。まず、基礎となるべき流体シミュレーションを計算するというコンピュータ的汚れ仕事をBlenderにやってもらい、それから見る人が「現実の災害」だと納得するようなディテールを慎重に考えて付け足しましょう。

たとえば、以下のようにして巨大な都市を襲う津波を表現できます。小さなモデルを作り、小さな波を高解像度で作成し、そして100mの波と1mの波の違いに誰も気づかないように祈ります(実際、気づかないでしょう)。波には大量のノイズと雲でカラーに変化を加えたテクスチャを貼りましょう。波がぶつかるいろいろな部分には煙(または霧)のエミッターをたくさん追加し、衝突にあわせた時間と、衝突から考えられる方向に放射されるように調整します。車やごみ(そしておぼれる人々...)を波先に浮いて動いているようにアニメート(そしてバイク)します。上に向けた霧エミッターを波頭に配置し、波頭から空中に上がるしぶきを再現します。カメラをどこに置くのか、ズームレンズを使うかワイドレンズを使うか、などもよく考えます(見る人がかわいいような俳優たちを見下ろしている感じにするのか? それともいっしょにおぼれている感じにするのか?)。このように、流体シミュレーションに加えて細部への気配りすることで、作品の出来がぐっと上がるのです。

BlenderのLBMエンジンにとって以下のようなことは計算を難しくします。

- 巨大なドメイン
- 長い時間
- 低い粘度
- 大きな速度

水の粘度はもともと非常に低いため、特に低い解像度の場合には乱流が正しくとらえられないかもしれません。現在のところはまだ、コンピュータグラフィックスにおける流体シミュレーションのほとんどは、じっくり見れば現実の水とは違うところがあります。基本として、物理的な設定を当てにすぎないほうが良いです(たとえばドメインの大きさやアニメーションの秒数)。低い解像度でなるべくすべての動きが正しくなるように調整したあと、できるだけ(または必要に応じて)高い解像度でシミュレーションしましょう。

謝辞

流体シミュレーションの統合はGoogle Summer-of-Codeのプロジェクトとしてなされました。エンジンについての詳細は www.ntoken.comで見られます。以下のアニメーションは流体エンジンがBlenderに統合される前に流体エンジンで作られました: [Adaptive Grids](#), [Interactive Animations](#)。Blender-SoCプロジェクトをとりまとめてくれたChris Want、プロジェクトを導いてくれたJonathan Merritに謝意を表します。そしてもちろん、これらすべての発端であるGoogleにも感謝します。SoCの進行はここにポストされています: [SoC-Blenderfluid-Blog at PlanetSoC](#)。

エンジンそのものはドイツ・エルランゲンのDepartment of Computer Science 10 (System Simulation, LSS)において、Nils Thuereyによって、以下のかたがたの協力、監修のもとで開発されました: U. Ruede, T. Pohl, C. Koerner, M. Thies, M. Oechsner and T. Hofmann。

<http://www10.informatik.uni-erlangen.de/~sinihue/img/lsslogo.png>

<http://www10.informatik.uni-erlangen.de/~sinihue/img/unierlangenlogo.png>

付録

制限と解決策

- ひとつの.blendファイルにつきドメインは一つだけ(少なくともバージョン2.42に関しては)ですが、流体は複数でもかまいません。
準備段階ではチェックしたい流体をひとつだけ囲むようにドメインを動かしながら作業し、最終的なシミュレーションをするときにドメインを大きくして一度にすべての流体を囲みます(計算時間は増えるでしょう)。このようにすると、計算時間をコントロールできる利点があります。
- 準備が上手くいかないとしたら、法線の向きが正しいかを確認しましょう(Edit modeですべてを選択し、recalculate normalsを一度実行してみましょう)。
- 現在のところ無重力でのシミュレーションは問題が起こります。これが解決されるまでは、非常に小さな重力を設定してください。
- Volumeで初期化されるオブジェクトは閉じたメッシュで、「内側」がないといけません(よってplaneには使えません)。planeには Shellを選択するか、押し出しをしてください。
- バイクするとBlenderがフリーズすることがあります。そしてEscを押すとしばらくして動き出すようになります。これは解像度が高すぎてメモリスワップが起こっているためです。そうするとすべてが恐ろしく遅くなります。その場合は解像度を下げるといいでしょう。
- バイクするとBlenderがクラッシュすることがあります。これは解像度が物凄く高く2G以上のメモリが使われているときに起こります。解像度を下げてください。多くのオペレーティングシステムでは1つのプロセスが使えるメモリの総量は制限されています。コンピュータにそれ以上のメモリが搭載されていても、ひどい...
- メッシュは閉じていないといけませんので、もし(たとえば流体オブジェクトの)一部が上手く表示されない場合は、その部分の頂点がちゃんとつながっているか確認してください。不幸なことにSuzanne(猿)は閉じたメッシュではありません(目の部分が分離しています)。
- 流体シミュレーションでエラーメッセージ(たとえば“init has failed”で始まるような)が出るときは、ドメインの設定が正しいかどうか確認してください。設定をデフォルトの状態にリセットしてみるといいかもしれません。
- 単一の流体サーフィスを読み込むのにはこのスクリプトが使えます: [.bobj.-Import-Script](#)
- 1Gを超えるサイズの流体はバイクできません。LargeAddressAwareビルドでもです。これは現在の流体エンジンの制限です。
- 最初のフレームでは数百MBのデータでも、後のほうのフレームではGB単位になっているせいでバイクが進むと失敗するということがあります。その場合、後のほうの1フレームだけを中くらいまたは目的通りの最高の解像度でバイクしてみて、動くかどうかを確かめてください。
- subdivisionを1から2にすると、使用メモリは2倍になります。
- “generate particles”をつかうとメモリ使用量が増えます。表面積と複雑さが増すからです。ふつうのfluid-sim/パーティクルはおそらくもう少し使用メモリが少ないです。

関連項目

to do check these links, make sure they are compatible with Blender 2.6

- [Tutorial 1: Very Basic Introduction](#)
- [Tutorial 2: The Next Step](#)
- [Tutorial 1&2 Gui Changes for newer builds](#)
- [Another BSoD fluid tutorial](#)
- [Developer documentation \(implementation, dependencies, ...\)](#)

外部リンク

to do check these links, make sure they are compatible with Blender 2.6

- [An Introduction to Fluid Simulations in Blender \(video\) \(Blendernation link\)](#)
Learn the basics of how to set up a fluid simulation in Blender with an obstacle.
- [Fluid Simulator Tutorial \(video\) \(Blendernation link\)](#)
Very easy to understand video-tutorial to fluid simulation newcomers. Also covers some of the most common pitfalls.
- [Guide on Blender Fluid Simulator's Parameters \(Blendernation link\)](#)

煙 (smoke) シミュレーション

開発ノート

Blenderの新しい煙シミュレーションは論文'[turb.php|Wavelet Turbulence for Fluid Simulation](#)'および付随するサンプルコードに基づいています。BlenderにはDaniel Genrichによって実装され、現在はJanne Karhulによってメンテナンスされています。

内部処理

このシミュレーターはボリュメトリック流体ベースモデルを使い、結果はボクセルグリッドとして出力されます。このボクセルデータはBlenderの3Dビュー内でカスタムOpenGLシェーディングによってインタラクティブに表示され、ボクセルデータテクスチャを使ってレンダリングできます。Blenderの煙シミュレーションは**パーティクル**のまわりのボクセルをラップします。そのためには、パーティクルを放射するオブジェクトと、煙がレンダリングされる領域を囲む「ドメイン」オブジェクトが必要となります。

Note

ドキュメントのこのパートはバージョン2.58を使っています。

作業の流れ

煙シミュレーションは流体シミュレーションに似ています。シミュレーションにはドメイン(Domain 領域)とフロー(Flow 流れ)オブジェクトが必要です。

- [ドメイン](#)にするオブジェクトを作成します。シミュレーションの領域を規定するメッシュです。
- [フロー](#)にするオブジェクトを作成します。煙を生成する領域を規定するものです。
- [衝突](#)オブジェクトをセットします。シーンの中で煙とかわるオブジェクトです。
- 煙に[マテリアル](#)を適用する。
- シミュレーションを[バイク](#)する。

何か困ったことがあれば[付録](#)を参照してください。

煙ドメイン(Domain)

流体シミュレーションと同じく、ほとんどの設定項目はドメインオブジェクトが選択されているときに現れます。

ドメインの作成

煙をシーンに追加する前に、煙のシミュレートが行われる領域を規定する必要があります。Blenderの物理シミュレーションではこの領域をドメインと呼んでいます。寸法はカメラにあわせて後で変えられるので、まずは立方体でいいでしょう。デフォルトの立方体をSおよびドラッグで大きくしましょう。

煙をシーンに追加する前に、煙のシミュレートが行われる領域を規定する必要があります。Blenderの物理シミュレーションではこの領域をドメインと呼んでいます。寸法はカメラにあわせて後で変えられるので、まずは立方体でいいでしょう。デフォルトの立方体をSおよびドラッグで大きくしましょう。

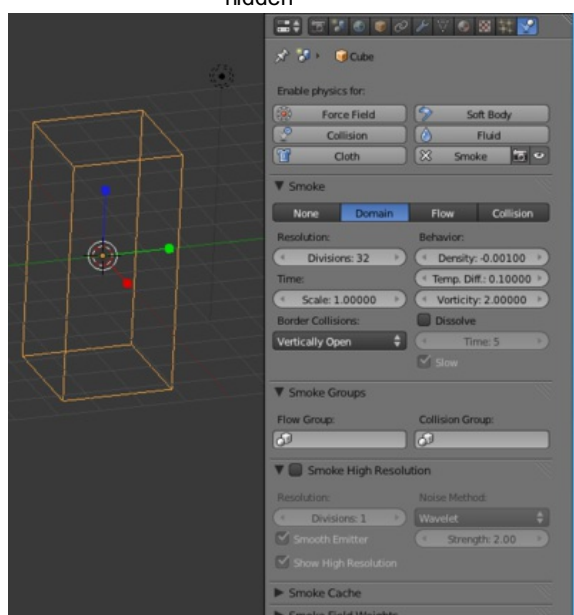
object Modeにいたることを確認して、physicsタブに移動しましょう。smokeを追加し、Domainのラジオボタンを選択します。まずはこれでおしまいで。あとでまたセッティングをします。



The physics tab might be hidden

Add smoke

Chose domain for the cube



The Smoke Domain Object

共通オプション

Resolution

煙の細部の緻密さです。32では数秒でバイクが完了しますが、100にするとほとんどのコンピュータでは30分ほどかかるかもしれません。

Time Scale

シミュレーションのスピード。

Border Collisions

Vertically Open

ドメインの上部および下部の壁に当たった煙は消えます。

Open

ドメインの壁に当たった煙はすべて消えます。

Collide All

ドメインの壁は衝突オブジェクトとして扱われ、煙はぶつかってドメイン内にとどまります。

Temperature and Density

密度と温度が煙に与える影響。数値を上げると煙は速くのぼります。

Vorticity

乱流、回転、あるいは渦の度合い。

Dissolve

時間の経過とともに煙が消えます。

Time

煙が消える速さ。

Slow

Timeではなく1/Timeを使います。煙が消えるのがより遅くなります。

煙グループオプション

to do

高解像度オプション

高解像度オプションを使うと、まず低い解像度でシミュレーションしてからノイズを施して解像度を上げることができ、高い解像度でシミュレーションせずに済みます。アニメーションを作る際、低解像度で簡単にシミュレーションをし、そのあとで煙の全体的な流れを変えずにディテールを付け加えることができます。

いくつかの方式が使えます。デフォルトのWaveletは'[turb.php|Wavelet Turbulence for Fluid Simulation](#)'の実装です。

Resolution/Divisions

ノイズを使って煙の解像度を上げます。

Smooth Emitter

スムーズ化してムラを防ぎます。

Show High Resolution

高解像度の煙を表示します。

Noise Method

Wavelet

FFT

Strength

ノイズの強さ。

フィールドウェイトオプション

様々な力および力場が煙にどの程度影響するかを決めます。

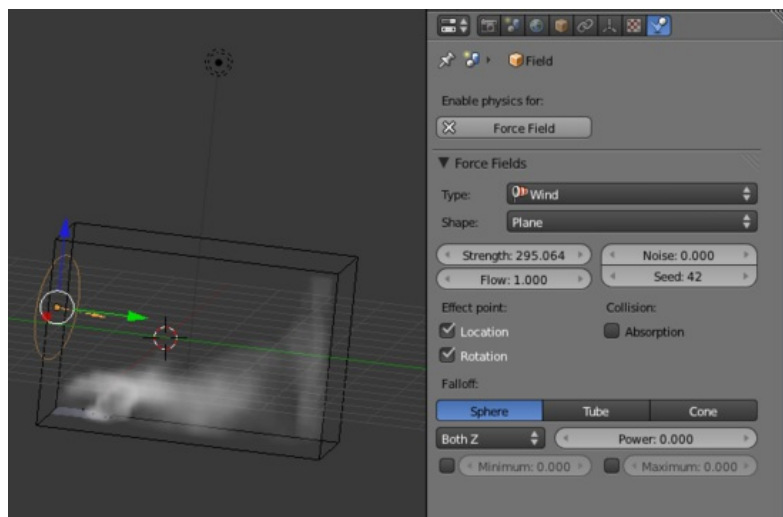
Gravity

煙が重力の影響を受ける度合い。

All

力場全般の影響力。

その他の項目はそれぞれの力場の影響力です。



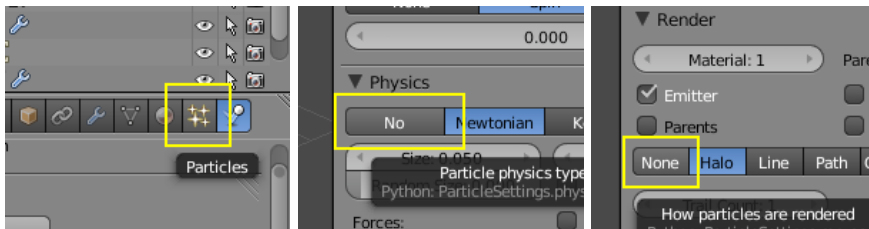
フロー (Flow) オブジェクト

フローオブジェクトの作成

煙を取り囲む領域を規定したら、今度はその内部で煙を放射するオブジェクトを追加しましょう。小さい立方体をもう一つ追加し、ドメインの内部に納まっていることを確認します (3Dビュー内で **ShiftA** » Mesh » Cube)。

Edit Modelに入り、physicsタブで小さい立方体に煙を追加します。今回はFlowを選んでください。

煙はオブジェクトそのものから放射されるのではなく、オブジェクトが放射したパーティクルから放射されます。よってわれわれはパーティクルシステムの設定もしなくてはなりません。小さい立方体を選択した状態でparticleタブに移動してください。新規のパーティクルシステムを追加し、われわれは発生源には動いてほしくないのでphysicsはオフにします。またパーティクルそのものが表示されないように、レンダリングもオフにします。

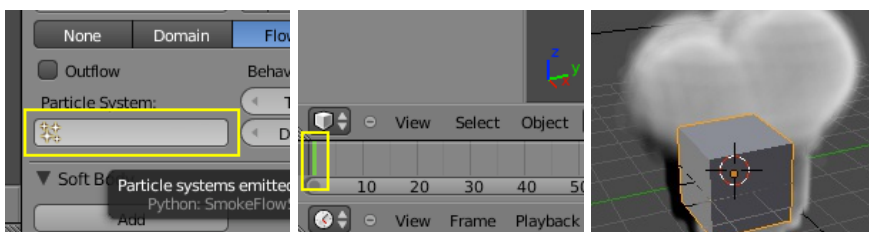


The particles tab is right next to the physics tab

We don't want the particles to be affected by physics

We also don't want to see the particles

physicsタブに戻り、smokeセクションでパーティクルシステムを選択します。リストの中に(われわれは名前を変更していないので)'ParticleSystem'という項目があるはずですが、これで、timelineウィンドウでフレームを進めると立方体から煙が出てくるでしょう。またAltAでアニメーションをスタート/ストップすることもプレビューできます。



Select the newly created particle system here

Either scrub on the timeline or use ALT+A

Now there should be smoke in the viewport

設定

Outflow

シミュレーションから煙を消去します。

Particle System

このオブジェクトから放射されるパーティクルシステム。

Initial Velocity

エミッターパーティクルの速度を煙がひきつぎます。

Multiplier

速度が煙に伝わる度合い。

Initial Values

Absolute Density

面からパーティクルが放射される密度を固定します。

Density

密度の初期値。

Temp. Diff.

環境の温度に対する差。

衝突 (Collision)

Collisionオプションを使うと、煙が他のオブジェクトにぶつかってさえぎられるようになります。現在のところ静止している衝突オブジェクトのみサポートされています。

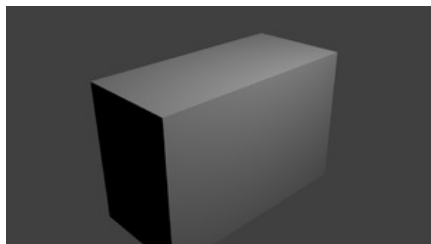
力 (Forces)

Blenderの力場 (windやvortexなど)も使うことができます。他の物理シミュレーションシステム同様、煙に対しても影響を与えられます。

煙のマテリアル

マテリアルの作成

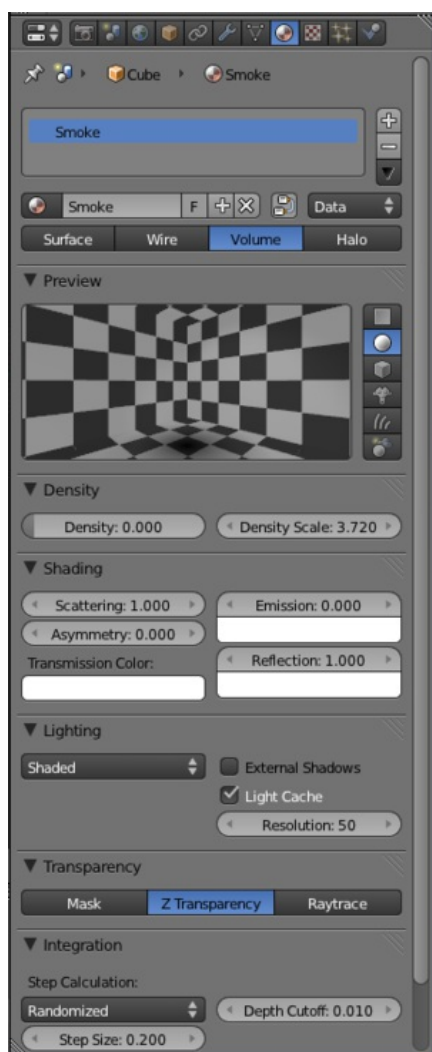
煙はシミュレートするのは簡単ですが、レンダリングはそうではありません。



The Render without the correct smoke material.

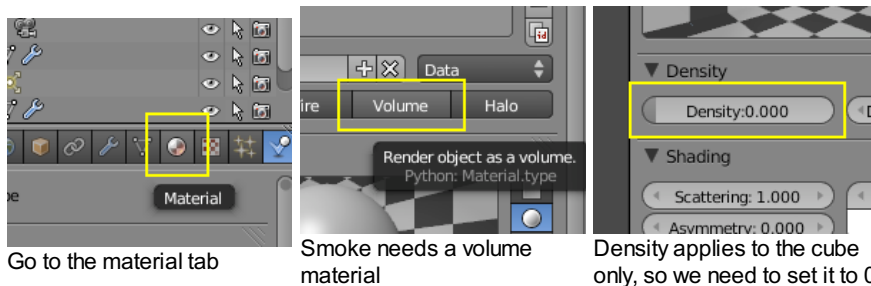
今の時点でレンダリング(F12)してみると、たんなる箱が現れます。またアニメーションをレンダリング(CtrlF12)すると、まったく何も動きません。

The material must be a volumetric material with a Density of 0, and a high Density Scale.



The material settings.

最初の点に関してはドメインオブジェクトのマテリアルを調整することで解決できます。煙を正しくレンダリングするには複雑なマテリアルが必要です。ドメインオブジェクトを選択して、Materialタブに移動してください。マテリアルの種類をVolumeにして、density(密度)は0にします。densityを0より大きくすると、ドメインオブジェクトそのものがVolumeマテリアルで満たされてしまいます。ドメインではなく煙にマテリアルを適用するには別の設定をします。これはあとでやりましょう。



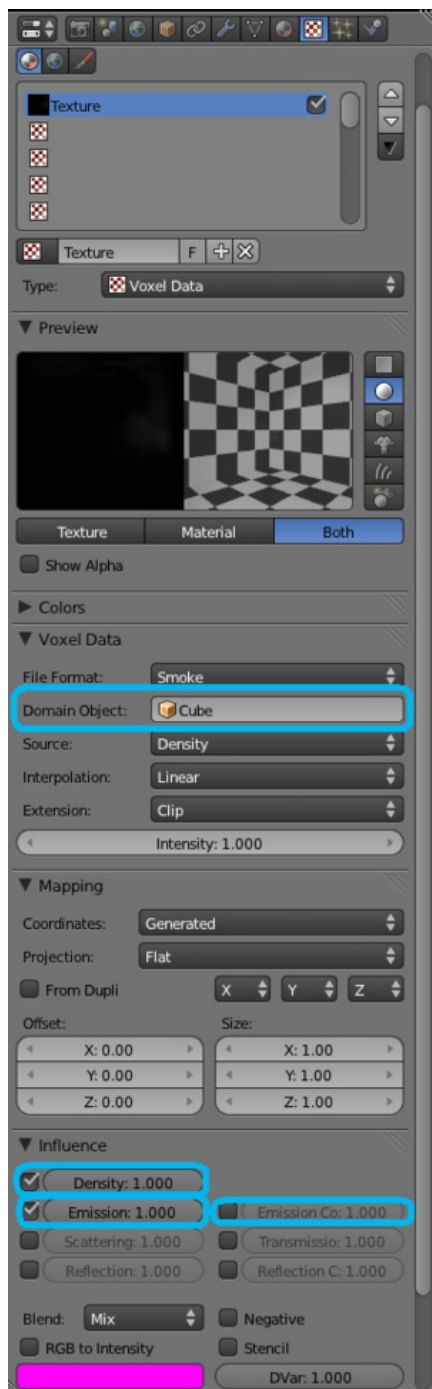
Go to the material tab

Smoke needs a volume material

Density applies to the cube only, so we need to set it to 0

テクスチャの追加

上記に加えて、煙にはテクスチャが必要です。Blender 2.5には煙をレンダリングするための**ボクセルデータ** ([eng](#))という新しい種類のテクスチャがあります。そこにドメインオブジェクトを指定し、influenceを設定しましょう。

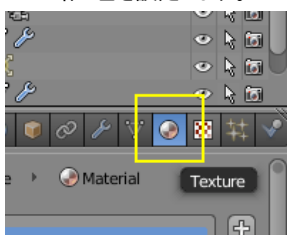


The texture settings.

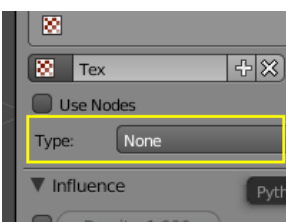
textureタブに移動してテクスチャの種類をVoxel Dataにし、設定項目が現れたらdomain objectとしてわれわれのドメインオブジェクトを設定しましょう (名前は特に変えていないので'Cube'になっているはずです)。influenceのセクションで、densityにチェックをして数値は1.000にします (Emissionにも自動的にチェックが付くはずです)。これで静止画のレンダリングに関してはうまくいくはずです。Emission Colorを調整することで色も変えられます。

💡 煙がよく見えるようにするには

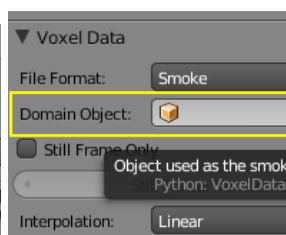
worldタブで、horizonに暗い色を設定します。



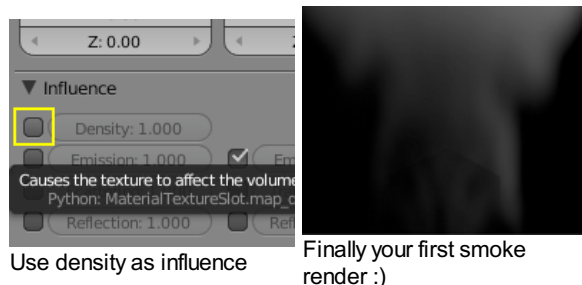
We need to add a texture of the smoke



Type should be Voxel Data



The domain is once again our big cube



Use density as influence

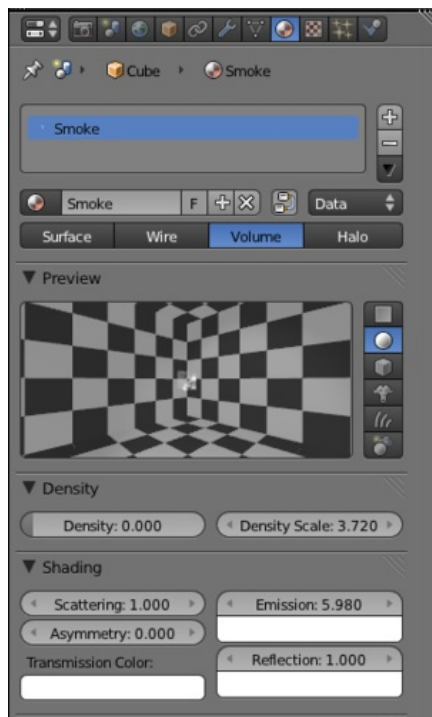
Finally your first smoke render :)



The rendered smoke. It's hard to see, but it's there.

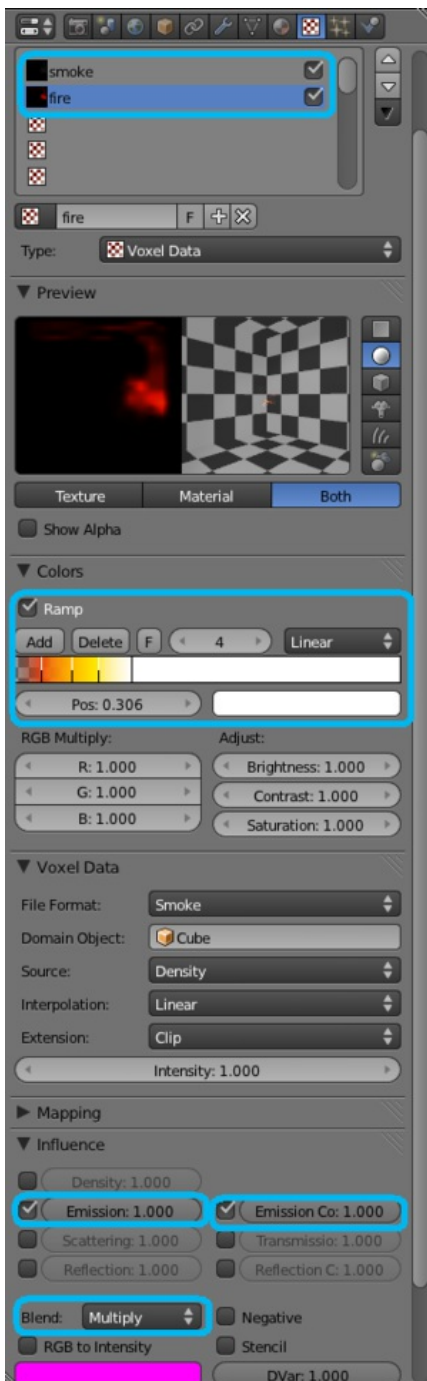
煙シミュレーターの拡張: 炎

別のテクスチャを設定すれば炎を表現することもできます！ 炎を作るには、MaterialsパネルでEmissionの数値を上げてください。

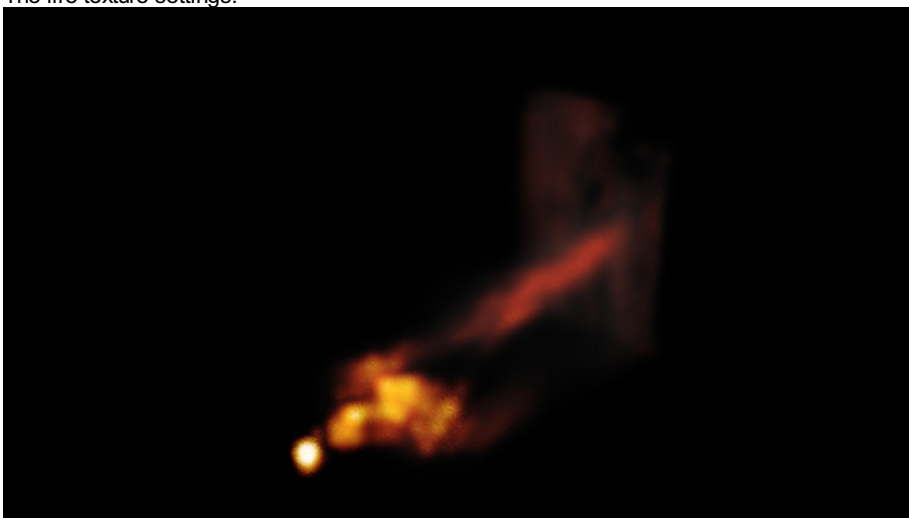


The Fire material.

それから、新しいテクスチャを追加します(先ほどのテクスチャはそのままにしてください。でないと煙が消えてしまいます) 炎らしいグラデーション(Ramp)を設定(アルファチャンネルも使用)して、emissionとemmission colorで使われる色を変更します。blend(合成方法)はMultiply(乗算)にしてください。



The fire texture settings.



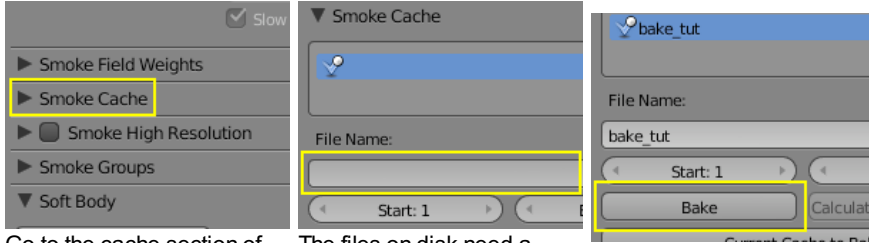
The fire render.

煙シミュレーションをバイクする

アニメーションをレンダリングしたい場合は、まずシミュレーションをバイクする必要があります。バイクとはたんに、シミュレーションを計算することです。煙をバイクするには、そのデータファイルを保存しなければいけません。計算結果に名前をつけて蓄えます。場合によってはバイク中にBlenderがクラッシュする可能性があります。トラブルシューティングを参照。

timeline ウィンドウでフレームを移動したりAltAで3Dビューをアニメーションさせたりしたときに、すでにある種のバイクはメモリ内にされています。しかし、レンダリングするためにはバイクデータはディスク上になければいけません。また、バイクする前には.blendファイルを保存してください。

つぎにドメインオブジェクトを選択してphysicsタブに移動して、Smoke Cache セクションで設定をします。テキストボックスにキャッシュのファイル名を入力してEnterを押してください。そしてEnterボタンを押すと、シミュレーションのデータがディスクに保存されます。タイムラインをドラッグすると滑らかに煙が動くのに気づきましたか？ これでアニメーションのレンダリングができるようになったはずですよ。



Go to the cache section of your smoke domain

The files on disk need a name

Finally we're ready to bake

付録

トラブルシューティング

- **Q.** Domainボタンを押すとクラッシュする。
- **A.** ドライバが古いと起こります。すべて最新のものに更新してください。

- **Q.** バイクするとクラッシュする。
- **A.** メモリが足りないためです。解像度を下げてください。

- **Q.** 煙がレンダリングされない。
- **A.** もう一度解説をよく読んでみてください。

- **Q.** 炎がおかしな結果になる。または見えない。
- **A.** マテリアルのemmissionを高い数値にしているか確認してください。また煙のテクスチャを設定しているか、炎のテクスチャをMultiplyにしているかを確認してください。

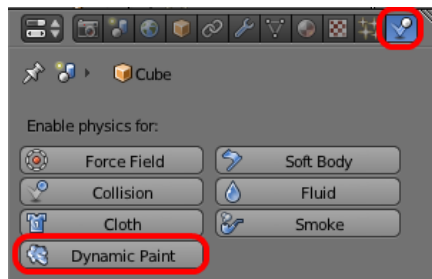
外部リンク

- [In-Depth introduction to smoke and fire in Blender 2.5 covering most of the pitfalls](#)
- [Guide to realistic fire in Blender by MiikaH](#)

ダイナミックペイント

ダイナミックペイント(Dynamic paint)は、オブジェクトをペイントキャンバスやブラシやブラシに変え、頂点カラーや連番画像、ディスプレイメントを作成できる新しいモディファイア & 物理システムです。これにより従来は難しかったたくさんのエフェクトが可能になります。例えば、雪の中の足跡、地面を濡らす雨だれ、壁に貼りつくペイント、徐々に凍っていくオブジェクトなど。

下記のガイドでは、ダイナミックペイントのユーザインターフェイスと機能全般について非常に基本的なことを説明しています。



ダイナミックペイントの起動

モディファイアの追加

ダイナミックペイントは、プロパティエディタの "Physics" タブから利用できます。

タイプ

モディファイアには二つの違うタイプがあります。

[Canvas](#)

オブジェクトがダイナミックペイントの Brush からペイントされるようになります。

[Brush](#)

オブジェクトが Canvas にペイントするようになります。

注意

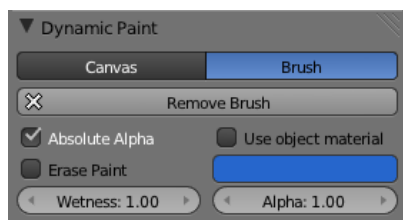
BrushとCanvasを同時にONにすることもできます。この場合、同じオブジェクトの "Brush" がその "Canvas" に影響を与える事はありませんが、シーン内の他のオブジェクトに相互作用できます。

その他役に立つページ

- [A step-by step introduction\(英文\)](#)
- [A detailed guide that covers every setting with images and examples\(英文\)](#) (現在更新停止中)

Dynamic Paint Brush

メインパネル



Brush main panel

最初の Brush パネルでは、Brush が Canvas カラーサーフェスに影響を与える方法を設定できます。

Absolute Alpha

この設定は Brush のアルファへの影響を制限します。これがないと Brush はフレーム毎にサーフェス上に何度も繰り返し「追加」されていき、アルファ値、さらには Brush の Canvas 上での影響度も増加していきます。しかし大抵の場合、すでに Brush レベルになっていれば、Brush のアルファは増加しない方がいいのです。

Erase Paint

Brush がペイントを追加する代わりに既存のペイントを消すようにします。

Wetness

新しいペイントがどれくらい「濡れる (Wet)」かを決めます。Wetness は "Paint" サーフェスの "Wetmap" で視覚化されます。"Drip" と "Spread" エフェクトのスピードも、ペイントがどれだけ濡れているかに依存します。

Use object material

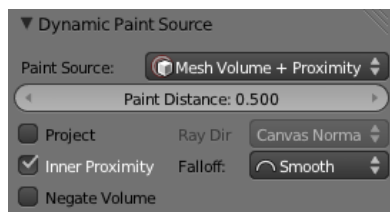
ONにすると、Brush の色として使用されるマテリアルを設定できます。これにはマテリアルのベースカラーと、リンクするすべてのテクスチャが含まれ、要はレンダリングされるディフューズカラーと一致します。現時点ではこの設定は "Blender Internal" レンダラを使用している時のみ利用可能です。

OFFの場合は、Brush の色を下のカラーボックスから指定できます。

Alpha

Brush のアルファもしくは可視性を決めます。最終的な Wetness 値はアルファの影響も受けます。

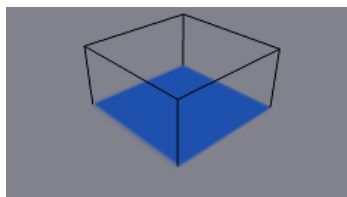
Source パネル



Brush source パネル

Brush "Source" 設定は、Brush がどう影響・接触するかを決めます。

現在、5つの Brush の挙動タイプを選択でき、それぞれ個別の細かい設定を持っています。



Brush Source - Volume

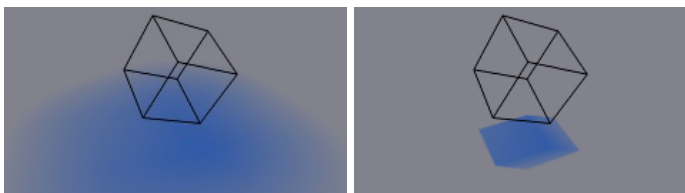
Mesh Volume

これはデフォルトオプションです。Brush が Mesh ボリューム内のすべてのサーフェスポイントに影響します。

Proximity

Brush Mesh 表面上の、最も近いポイントとの距離が設定した物である時のみ使用します。ボリュームはサーフェスに近くないため、必ずしもボリューム内部に影響が及ぶとは限らないことに注意してください。

Proximity の減衰タイプは、"Smooth"、"Sharp" から選択、もしくはカラー Ramp で調整することができます。



Brush Source - Proximity. Brush が "Project" 設定が ON の場合。Brush 周囲のすべての Canvas ピクセルに影が Normal 方向の Canvas にのみ影

響します

響しているのに注目

Project

Brush を Canvas へ決められた方向から投影します。基本的にこれは「方向を揃えた」Proximity とみなすことができます。

Mesh Volume + Proximity

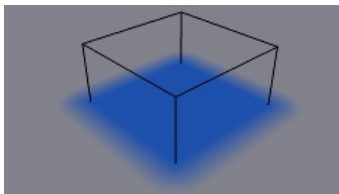
Volume タイプと同じですが、設定した距離より外にも影響します。「Proximity」タイプと同じ減衰タイプが使用できます。

Inner Proximity

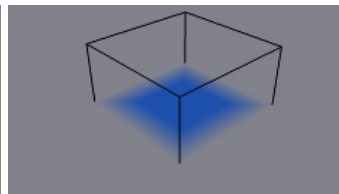
Proximity を Mesh ボリューム内側にも適用します。

Negate Volume

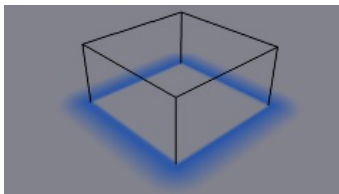
Mesh ボリューム内の Brush のアルファを反転します。



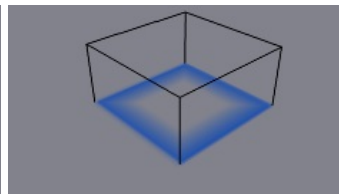
"Volume + Proximity" Brush、追加の設定はなし



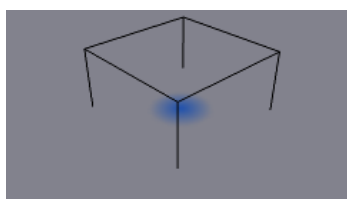
Inner Proximity。Proximity の減衰度がボリューム内で視覚化されている



Negate Volume。ボリュームの内側が完全に透過している



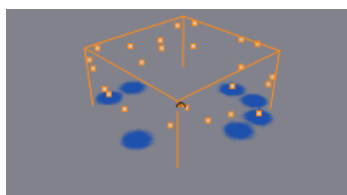
Inner Proximity と Negate Volume を同時に ON



Brush Source - Object Center

Object Center

非常に遅くなることもある Brush オブジェクト Mesh の Proximity の代わりに、Center(オリジン)との距離のみを計算します。これは非常に速く、多くの場合これで十分です。

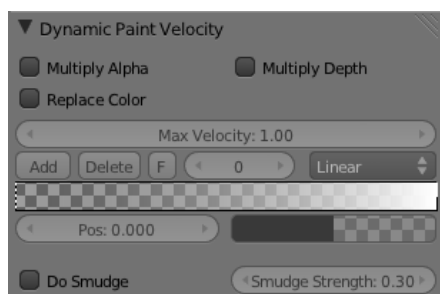


Brush Source - Particle System

Particle System

Brush の影響度を、選択されたパーティクルシステムからのパーティクルによって決定します。

Velocity パネル



Velocity パネル

このパネルには、オブジェクトの Velocity(速度)を元にした Brush オプションがあります。

上側にはカラーランプ(Color Ramp)と、関連する設定があります。基本的にカラーランプは Brush の Velocity 値を表現しています。左側が 0 Velocity になり、右側が「最大 Velocity」になります。スピードは「Blender 単位/フレーム」を尺度としています。

その上のチェックボックスはカラーランプの影響を決めるのに使用できます。

Multiply Alpha

現在の速度に応じてカラーランプのアルファ値を使用し、Brush のアルファと乗算します。

Replace Color

Brush カラーをランプのカラーで置き換えます。

Multiply Depth

Brush の「接触深度」エフェクトと乗算します。基本的には、Brush のスピードに応じてディスプレイと波の強さを調整できます。

Smudge 設定

Smudge を ON にすると、Brush が移動時にサーフェス上の色を「スマッジ(こする)」(もしくは「スミア」)するようになります。このエフェクトの強さは "Smudge Strength" プロパティから設定できます。

Smudge が ON になっていても、Brush は通常のペイントエフェクトを行います。もしスマッジブラシのみにしたい場合は、アルファを 0 にしてください。また、Smudge と一緒に "Erase" オプションを使用することもできます。

Waves パネル



Brush Waves パネル

このパネルは Brush の "Wave" サーフェスへの影響を調整するのに使用します。

"Wave Type" メニューで、この Brush が波のシミュレーションでどんなエフェクトを与えるのかを選択することができます。その下には二つの細かい調整用の設定があります。

Factor

Brush の "depth" がシミュレーションにどれだけ強く効果を与えるかを設定します。負の値で Brush が水に下げる代わりに引き上げることもできます。

Clamp Waves

時には Brush が非常に深くサーフェス内に入り、シミュレーションがおかしくなることがあります。この設定を使用することで、一定の深さに影響を「制限」することができます。

さらに4つの "Wave Type" オプションが利用可能です。

Depth Change

このオプションは、Brush のサーフェスとの接触の深さが「変わった」時点で波をつくるようにします。Brush がそのままであれば、影響はありません。

このタイプと負の "Factor" を使用することで、船のようなオブジェクトが移動する時の「波紋」がよい感じに作成できます。

Obstacle

接触時、常に一定の影響をサーフェスに与えます。波もこの Brush タイプで反射します。ただし Brush が留まっている場合、波シミュレーションアルゴリズムの性質の所為により、不自然な「くぼみ」をサーフェスに生成します。

Force

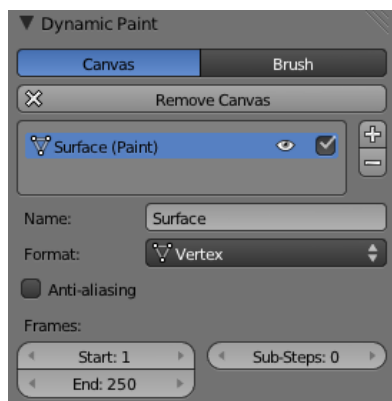
直接波の動きの速度に影響を与えます。Brush の接触深度によるエフェクトと一対一で対応していませんが、力の強さは深度に依存していません。

Reflect Only

このタイプは単独ではサーフェス上に目に見える効果はありませんが、すでにサーフェス上にある波を反射します。

Dynamic Paint Canvas

メインパネル



Canvas のメインパネル

Canvas の最初のパネルには、ダイナミックペイントサーフェスのリストがあります。これらのサーフェスは、基本的にはペイントのレイヤで、それぞれ独立して働きます。これらは別々に設定したり、バイクすることができます。

サーフェスタイプ/フォーマットが3Dビューで結果をプレビューできる場合、プレビュー表示をON/OFFできる目のアイコンが表示されます。

チェックボックスはサーフェスをアクティブにするか否かをON/OFFします。もし選択されていない場合、計算やプレビューは行われません。

各サーフェスには簡単に区別がつけられるよう、名前をつけることができます。

その下では、サーフェスタイプの設定と、品質とタイミングの設定を調整することができます。

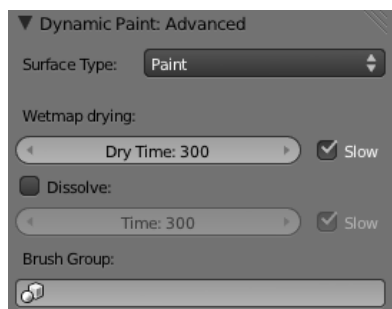
サーフェスにはフォーマットとタイプがいくつかあります。フォーマットはデータを格納・出力する方法を決め、現在二つのフォーマットが利用可能です。

- Image Sequences (連番画像)。ダイナミックペイントが UV 展開された画像ファイルを、出力に設定された解像度で作成します。
- Vertex (頂点)。ダイナミックペイントが直接 Mesh の頂点データを操作します。結果はポイントキャッシュに格納され、ビューに表示できます。ただし、Vertex Level を使用するには、メッシュを多く分割する必要があります。

Quality 設定では、画像の解像度 (Image Sequence用) と Anti-Aliasing (アンチエイリアシング) が調整できます。

そして、サーフェスの処理開始 (Start) と終了 (End) フレーム、サブステップ (Sub-steps) 数を設定できます。Sub-steps はフレーム間に追加でサンプリングする数で、非常に速いブラシがある場合に必要になります。

Advanced パネル



Canvas advanced パネル

"Advanced" パネルでは、サーフェスタイプと関連する設定を調整できます。

各サーフェスにはサーフェスが何に使用されるかを定める「タイプ」があります。以下のタイプが使用可能です。

- Paint
- Displace
- Waves
- Weight

共通するオプション

各サーフェスタイプには特別な設定があります。ほとんどのタイプには、*Dissolve* と *Brush* 設定があります。

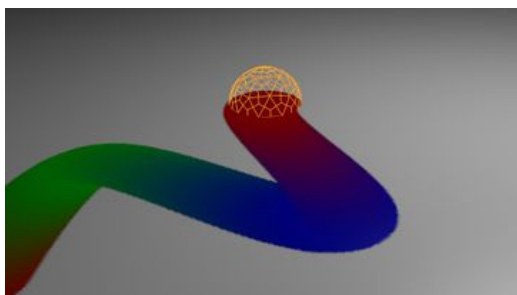
Dissolve

設定された期間内にサーフェスを元の状況にスムーズに戻すのに使用されます。

Brush Group

ブラシオブジェクトを選択・使用する、オブジェクトグループを設定するのに使用します。

Paint

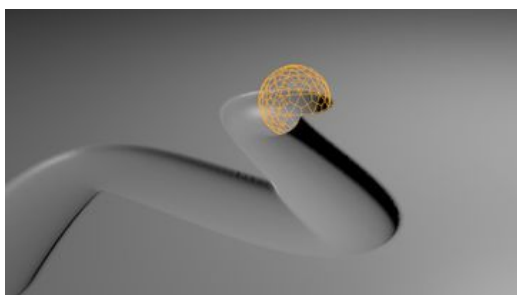


サーフェスのペイント

"Paint(ペイント)" は基本的なサーフェスタイプで、Color(色)と Wetness(湿り度) 値を出力します。Vertex サーフェスの場合、結果は頂点カラーとして出力されます。

ウェットマップ(Wetmap)は、Wetness のペイントを視覚化する白黒の出力です。白は最大に濡れており、黒は完全に乾いています。通常、レンダリング用のマスクとして使用されます。いくつかの「ペイントエフェクト」は、ウェットペイントのみ作用します。

Displace



Displace Surface

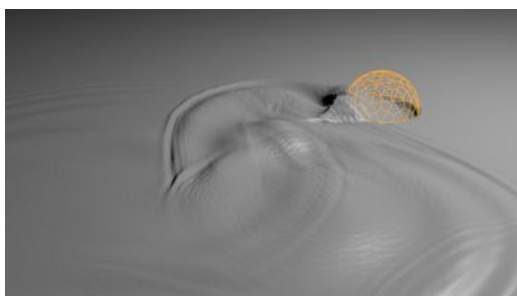
このタイプのサーフェスは、Brush オブジェクトとの接触の深さを出力します。



Tip

Displace出力が荒すぎると思われる場合、大抵はモディファイアスタック内の Dynamic Paint モディファイアの後に「Smooth」モディファイアを追加すると役に立ちます。

Waves



Waves サーフェス

このサーフェスタイプは、シミュレートした波のモーションを生成します。Displace のように、Wave サーフェスも Brush の接触深度を Brush の強さを決めるのに使用します。

以下の設定をモーションの調整に使用できます。

Open Borders

メッシュの「端」で波が跳ねかえらず通り抜けることができるようになります。

Timescale

シミュレーション結果に影響することなくシミュレーション速度を直接調整します。低い値でシミュレーションが遅くなり、高いと早くなります。

Speed

サーフェス上を波が伝わる速さに影響します。この設定もシミュレーションのサイズに対応します。半分のスピードでサーフェス二倍と同等です。

Damping

時間とともに波の力を弱めます。基本的に波の消える速度を調整します。

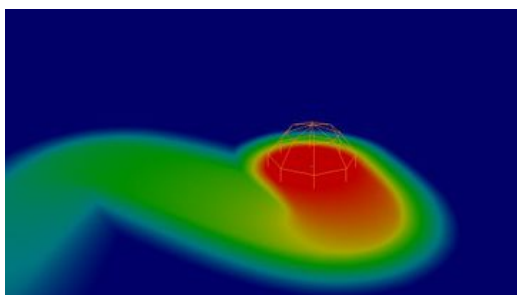
Spring

水を「ゼロレベル」にまで引き戻す力を調整します。



Tip

時々ブラシ周辺の波の動きが非常に不安定になることがあります。通常、Wave の Speed や、Brush の "wave factor"、さらにメッシュ / サーフェスの解像度を小さくするといいでしょ。

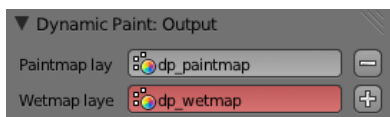
Weight

Weight サーフェス

これは Vertex フォーマットでのみ使用可能な、特別なサーフェスタ입です。他の Blender のモディファイアやツールで使用できる頂点ウェイトグループを出力します。

Tip

通常、Weight サーフェスでは各ウェイト値の間の減衰度をスムーズにできる、"proximity" ベースの Brush を使用することを推奨します。

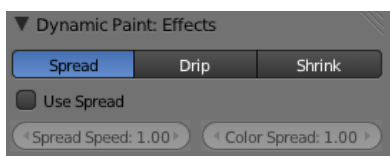
Output パネル

Canvas output パネル

"Output" パネルでは、サーフェスへのその結果の出力方法を調整できます。panel you can adjust how surface outputs it's results.

"Vertex" フォーマットのサーフェスでは、結果を生成する対象の Mesh データレイヤ(サーフェスタ입により、Color / Weight)を選択できます。[+][−] アイコンを使用し、名前付きのデータレイヤを追加・削除できます。与えられた名前の付いたレイヤが見つからない場合、赤で表示されます。

"Image Sequence" サーフェスの場合、使用する「UV レイヤ」と、出力ファイルを保存するディレクトリ、ファイル名、画像フォーマットを指定できます。

Effects パネル

Canvas effects パネル

これは "Paint" タイプ用の特別な機能です。Canvas サーフェス上にアニメーションを生成します。

現在、3つのエフェクトが利用可能です。

Spread

ポイントの周囲にゆっくりとペイントが広がり、最終的にはつながった領域すべてを塗りつぶします。

Drip

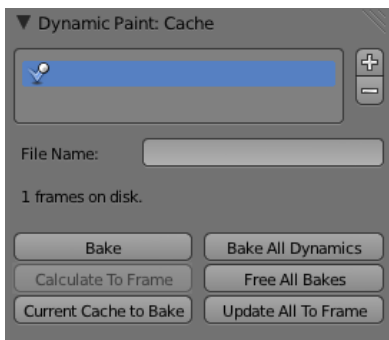
Blender のフォースフィールドや Gravity、Velocity と、ユーザ設定した影響度 (Influence) で指定された方向に、ペイントが動きます。

Shrink

ペイントされたエリアがゆっくりと完全に消えるまで縮小していきます。

Spread と Drip エフェクトは "wet paint" のみ影響を受け、ペイントが濁くと止まるまで移動が徐々に遅くなります。

Cache パネル



Canvas cache パネル

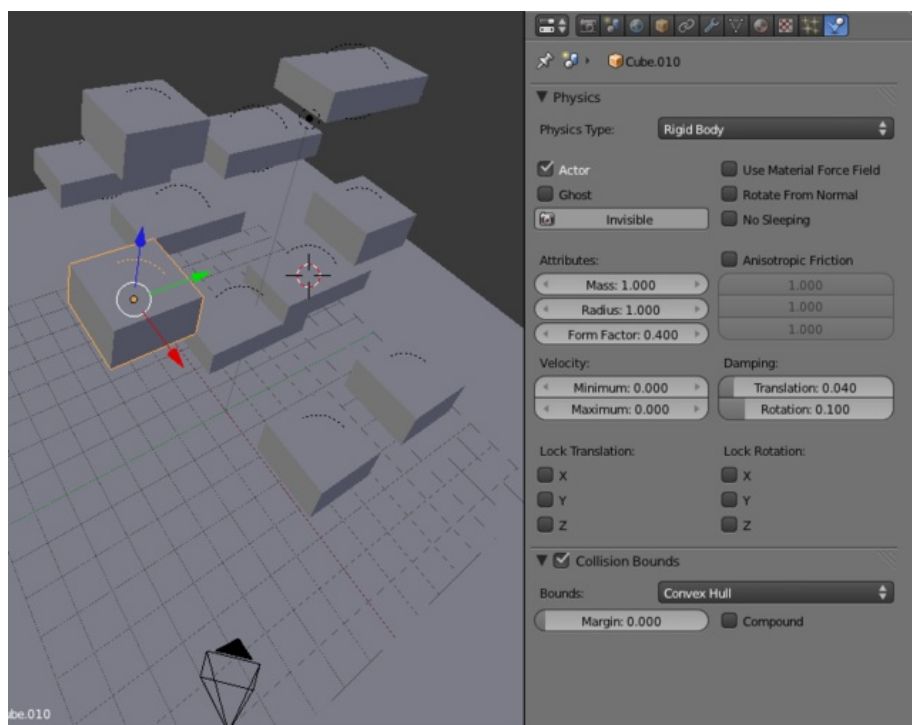
このパネルは現在 "vertex" フォーマットのサーフェスでのみ表示されます。ポイントキャッシュの調整とバイクに使用できます。

Converting Game Engine Physics

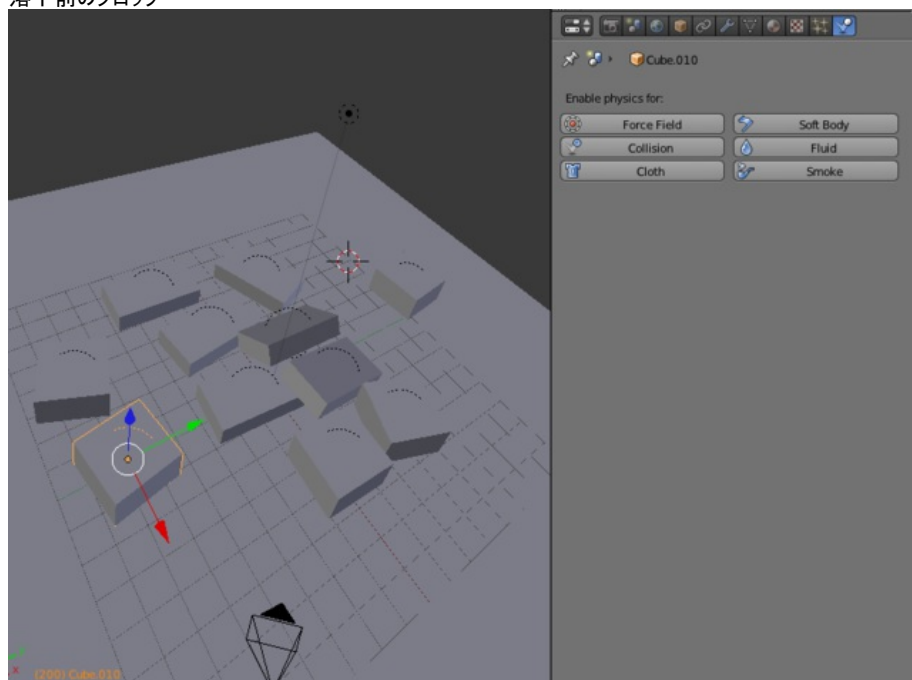
(ゲームエンジンの物理演算の変換)

オブジェクトによって壊される壁や、たくさんのオブジェクトの正確な物理法則による崩壊、落下、跳ね返りのアニメーションを作りたくなるかもしれませんが、ひとつずつキーフレームを挿入し、物理や加速度をシミュレートする F-Curve の調整に試行錯誤することもできますが、Blender ゲームエンジンの物理演算で簡単に自動で同じことができます。いまの Blender には、Blender ゲームで記録したアニメーションを Blender のアニメーションキーフレームに変える機能があります。

アニメーションの記録は Game (ゲーム) » Record Animation (アニメーションを記録) で行います。記録したアニメーションは Alt+A で再生できます。



落下前のブロック



山積みブロック

ものが積み上がって静止した状態だけがほしいなら、最終フレームに移動し、すべてのキーフレームを NLA に変えて削除すると、素早く削除できます。

モーショントラッキング

はじめに

モーショントラッキング(Motion tracking)は Blender で新しく利用できるようになった技術です。まだ開発中で、現在カメラトラッキングの基本的な機能のみ対応しています。しかしすでに制作での使用に耐えるものになっています。

利用方法

モーショントラッキングは現在の SVN Trunk で利用可能で、2.61リリースに含まれることでしょう。すべてのプラットフォームでデフォルトで使用可能で、そのまますぐに利用できます。

以下は現在 Blender で利用できるモーショントラッキング用ツールの説明の概要です。

手動2Dトラッキング

実写映像、特徴点、モーシヨンのすべてに利用できる汎用のアルゴリズムはありません。そのようなアルゴリズムの作成は可能ですが、非常に遅く、まだ失敗する可能性もあるため、2Dトラッキングを機能させる唯一の方法は手動でトラッキングのアルゴリズムを選択し、設定することです。現在デフォルト設定で、あまりブレがなく、特徴点がパースなどであまり大きく変形しない一般的な実写映像ならうまく動作すると思われます。

2Dトラッキングの改良はすでに私たちの ToDO リストにあります。現時点ではあまり優先度は高くありません。もしアルゴリズムと設定について自信がなく、このドキュメントを読みたくない場合は、自分で設定をいじり、動作する物を見つけてみてください。

グリースペンシルやグリッドを使用した手動レンズキャリブレーション

すべてのカメラは歪んだ動画を録画しますが、これはレンズの挙動の所為で歪みのないものではありません。正確なカメラモーシヨンを得るには、正確な焦点距離(Focal Length)の値と、歪みの「強度(Strength)」を知ることが必要です。

現在、焦点距離はカメラの設定もしくは EXIF からしか取得することができません — Blender 内には推測できるツールはないのです。しかし、歪みを補うおおよその値を探すのに役立つツールがいくつかあります。これは完全に手動のツールでもあり、歪みモデルによる影響を受けたグリッドや、映像内の直線を決める変形した昇目を使用することができます。

グリースペンシル(Grease Pencil)を使用することもできます — 単に Poly Line ブラシを使用して映像の上に直線とおぼしき線を描き、distortion 値を調整してグリースペンシルを実写上の線に合わせるだけです。

さらにカメラを正確にキャリブレーションする唯一の方法は、OpenCV の Grid Calibration ツールを使用することです。OpenCV は同じ歪みモデルを使用しており、問題はないと思われます。

カメラモーシヨンの解析

数学的見地からすると、カメラモーシヨンの解析とオブジェクトモーシヨンの解析に違いがないにも関わらず、現在はカメラの解析のみ対応しています。まだ、トライポッドモーシヨンやドミナントプレーンモーシヨン(すべてのトラック可能な特徴点の一つの平面上にある場合)の解析に未対応といった、いくつかの制限があります。将来的にはこれらの制限は解決される予定です。

シーン配置と安定化用基本ツール

解析後、もっと合成に便利るように、現実のシーンを3Dシーン内に配置させる必要があります。シーンの配置を行うため、床とシーンの方向、X/Y軸を決めるツールがあります。

また、時には最終結果の見た目をもっと良くするため、カメラからの動画を安定化させる必要があることもあります。2D Stabilization を利用し、カメラのジャンプや傾きを補正することで、カメラからの動画を安定することができます。

実写映像にシーンの合成を行う基本的なノード

映像に簡単にシーンを合成するための新しいノードがいくつかコンポジターに追加されました。2Dスタビライゼーション(Stabilize 2D)、歪み、歪み補正用ノード(Lens Distortion)があり、簡単に使用できます。

未実装のツール

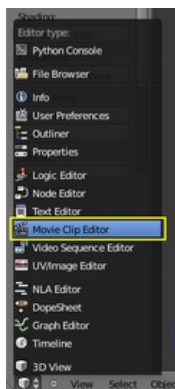
私たちの ToDO リストにあります。いくつかのツールはまだ Blender で利用できません。現在対応していないのは、回転シャッターフィルタリング、オブジェクトモーシヨン解析、モーシヨキャプチャなどです。しかし、現在実装されているものを使用し、これらをなんとかやってみることもできます。

マニュアル

Movie Clip エディタ

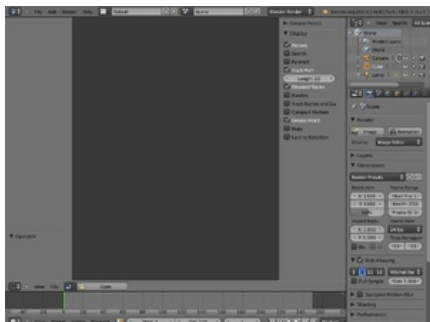
ほとんどのモーショントラッキングツールは、Movie Clip エディタというエディタに集約されています。現在、モーショントラッキングに関する物以外のツールは全くないのですが、将来的にはマスキングなどに使用できるよう拡張されるかもしれません。これがモーショントラッキングの習慣に比べ、このようにもって抽象的な名前に呼ばれている理由です。

このエディタはエディタタイプリストの最後にあります。



エディタタイプメニュー

Movie Clip エディタに切り替えた時、インターフェイスは以下のように切り替わります。



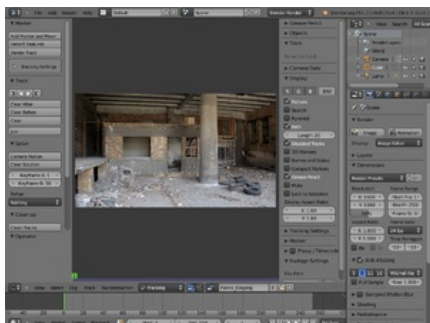
Movie Clip Editor インターフェイス

次は当然、作業を行うための新しいビデオクリップを開きます。これにはいくつかの方法があります。

- Movie Clip エディタのヘッダの **Open** ボタンを使用する
- Clip » Open メニューを使用する
- AltO> ショートカットを使用する

Movie Clip エディタでは動画ファイルと連番画像 (Image Sequence) の両方が使用できます。もし連番画像を使用する場合、ファイルの名前に一つだけ制限があります。最後の数字群は連続して増加していなければなりません。

ビデオクリップが Movie Clip エディタに読み込まれたら、インターフェイスに新たなパネルが表示されます。



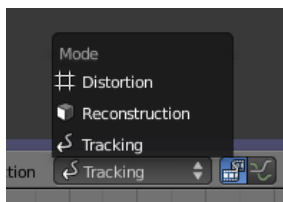
クリップを開いた時の Movie Clip エディタ

スクリーンにたくさんの新しいツールが現れ、すべてに短い説明がされています。

まず最初に書いておきたいのですが、カメラソルバ (Camera Solver) は、完全に分割された3つのステップで構成されています。

- 映像の2D トラッキング
- カメラ固有パラメータ (焦点距離、歪み係数) の指定・推測・キャリブレーション
- カメラの解析、シーン配置、シーン再構築

Movie Clip エディタ内のツールは使用されるステップごとに分けられており、2Dトラッキングだけが出来る時も、インターフェイスがシーン配置ツールとごちゃになることはありません。現在表示されるツールのカテゴリは、エディタのヘッダにある Mode メニューで変更できます。






Movie Clip エディタ Mode メニュー


ただし、ほぼすべてのオペレータはメニューから呼び出すことができるため、違うエディタモードのパネル内に表示されているツールを見るのに、毎回モードを変更する必要はありません。

Tracking モードでは、トラッキングとカメラ解析に関連するツールのみが表示されます。カメラ解析ツールがここにあるのは、解析後、大抵の場合は現在のトラックを再トラックしたり、もしくは解析をもっと正確にするため、新しいトラックを置きたくあるからです。

Tracking モードで利用可能なツール

Markers パネル

- **Add Marker and Move** オペレータは、新しいマーカーをマウス位置に置き(この場合はボタンの下、理想的ではありませんが、これは動作方法の解説ですので)、その後必要な位置に移動できます。必要な場所に移動した時、LMB  で新しいマーカーの位置を決定できます。また、 Enter と  Space もマーカーの位置決定に使用できます。

ただ、Ctrl LMB  を使用し、マーカーを直接映像の上に置く方が早いです。このショートカットではクリックした位置にマーカーを置きます。あと、マウスボタンを離すまで、マーカー位置をマウス移動で調整したり、トラックプレビューウィジェットを使用し、マーカーの位置の精度をコントロールすることもできます。

- **Detect Features** オペレータは、カレントフレームの特徴点を可能な限りすべて検知し、各特徴点にマーカーを置きます。このオペレータは他のフレームを考慮しないため、移動するオブジェクトに属する特徴点にもマーカーを置くこともあり、さらにカメラがこのショットから向きを変える途中である場合、カメラ移動後はマーカーが置かれません。

このオペレータにはいくつかのプロパティがあります。

Placement は、マーカーを置く場所をコントロールするのに使用します。デフォルトではフレーム全体に追加されますが、グリースペンシルで検知範囲として対象にする物を囲むと、マーカーはその囲んだエリア内だけに置くこともできます。これが "Inside Grease Pencil" の動作です。

また、完全に対象外のエリア(木や人など)を囲み、このエリアの外側にマーカーを置くこともできます。これが "Outside Grease Pencil" の動作です。

Margin は、画像の端と生成されたマーカーの距離をコントロールします。もしマーカーが画像の端に近すぎると、すぐにトラッキングに失敗してしまうため、手動で削除しなければなりません。手動による削除の量を減らすのにこのパラメータが利用できます。

Trackability は、マーカー配置時の最小トラッキング可能範囲を制限します。この値は特徴点発見アルゴリズムによるもので、基本的には低い値ではこの機能がすぐに追跡に失敗するようになり、高い値で追跡の失敗が少なくなります。マーカーの追加量をこの値でコントロールできます。

Distance で、配置されるマーカー間の最小距離を設定します。マーカーがお互いに近すぎる場所に置かれるのを防ぐのに必要です(このような配置はカメラソルバを混乱させます)。

- **Delete Track** は、読んで字のごとくのオペレータで、選択されたすべてのトラックを削除します。

Track パネル

- 最初の列のボタンは選択されたトラックの追跡を行うのに使用します。追跡は以下の条件で行います(ボタン順):
 - 1フレーム前
 - 逆順方向
 - 順方向
 - 1フレーム後

このオペレータは後述する Tracking Settings パネルの設定に依存します。

もし順方向の追跡中、アルゴリズムがいくつかのマーカーの追跡に失敗したと返した場合、これらを OFF にし、残りのマーカーについて追跡を継続します。アルゴリズムがフレームごとの追跡中に失敗したと返した場合、マーカーは OFF にはされず、新しいフレームでの特徴点の最も可能性の高い位置が使用されます。

- **Clear After** カレントフレームより後の選択中の全トラックで、追跡とキーフレーム挿入が行われたマーカーを削除します。
- **Clear Before** カレントフレームより前の選択中の全トラックで、追跡とキーフレーム挿入が行われたマーカーを削除します。
- **Clear** カレントフレーム以外の選択中の全トラックから、すべてのマーカーをクリアします。
- **Join** オペレータは、選択中の全トラックを一つに統合します。選択中のトラックには同じフレームで共通の追跡された、もしくはキーフレームが挿入されたマーカーを持たせないほうがいいでしょう。

Solve パネル

Camera Motion オペレータは、映像上にあるすべてのトラックと、このパネルで指定された二つのキーフレームを使用し、カメラモーションを解析(Solve)します。いくつか必要条件があります。

- 両キーフレームには最低8つの共通のトラックがあること
- この二つのキーフレーム間に著しい視差効果があることが望ましい

解析中にすべて上手くいけば、Reprojection(再投影)エラーがインフォメーションスペースと Movie Clip エディタのヘッダに表示されます。Reprojection エラーとは、映像上に投影した、各トラックから再構築された3D位置と、トラックの元の位置との距離の平均です。基本的に、Reprojection エラーは0.3未満では正確な再投影であり、0.3~3.0でまだ使用に耐えうる非常にいい解析が行われていることを意味します。3より上の値では、いくつかのトラックをもっと正確に追跡する必要がある、もしくは Focal Length(焦点距離)値や Distortion Coefficient(歪み係数)の設定がおかしいことを意味します。

Refine オプションで、解析中にリファインすべきパラメータを指定できます。この手のリファインは、カメラの特性をよく知らない場合や、ソルバがその特性用の最良のパラメータを見つけそうな時に便利です。ただし、それでもおおよその初期値を知っている必要があります — 最初に完全に異常な値を設定した場合、正しい値を見つけるのを失敗するでしょう。(訳注:intrinsic→特性と訳しています --[Yamyam](#))

Cleanup パネル

このパネルには一つのパネラとその設定があります。このパネラはまずいトラック(追跡が短すぎる、もしくは正確な再構築に失敗したトラック)をクリンアップします。各閾値はボタン下のスライダから設定できます。また、他にもいくつかの対応がまずいトラックに対し行えます。

- 単純に選択する
- トラックシーケンスのまずいセグメントを削除する
- トラック全体を削除する

Clip パネル

このパネルは現在、編集中のクリップを表示中の全ての3Dビューの背景として設定するパネラ **Set as background** が一つあります。表示中の3Dビューがない、もしくは Movie Clip エディタがフルスクリーンで開かれている場合は何も起こりません。

Tracking モードで使用可能なプロパティ

Grease Pencil パネル

標準のグリースペンシルパネルで、新しいグリースペンシルレイヤとフレームをコントロールできます。ひとつだけ他のエディタでのグリースペンシルの挙動と違うところがあります。新しいレイヤが「必要に応じて」作成された時(それ以前にレイヤを追加せずに描いた場合)、レイヤのデフォルトカラーがピンクになることです。これは多くの動画でストロークがわかりやすくなるからです。

Objects パネル



Movie Clip エディタの Objects パネル

このパネルには、トラッキングやカメラ、オブジェクトの解析に使用できるすべてのオブジェクトのリストがあります。デフォルトでは、このリストにはカメラ解析に使用するためのオブジェクトが1つだけあります。これは削除不可能で、他のオブジェクトはカメラ解析に使用することはできず、追加オブジェクトはすべてオブジェクトのトラッキングと解析にしか使用できません。これらのオブジェクトは Follow Track と Object Solver コンストレイントから参照することができます。Follow Track はデフォルトでは Camera オブジェクトを使用します。

新しいオブジェクトは、**+** ボタンを使用して追加でき、また、アクティブオブジェクトを **-** ボタンで削除することもできます。このパネルの下側のテキストエントリは、アクティブオブジェクトのリネームに使用されます。

間違ったオブジェクトにトラックが追加・トラッキングされてしまった場合は、Track » Copy Tracks と Track » Paste Tracks を使用し、他のオブジェクトにコピーできます。

(カメラやオブジェクトトラッキングに使用される)全種類のオブジェクトのユースケースは同じです。特徴を追跡し、カメラデータを設定し、モーションを解析します。カメラデータはすべてのオブジェクト間で共有され、カメラモーションの解析中のみ、カメラ特性のリファインが行われます。

Track パネル



Movie Clip エディタの Track パネル

最初に、トラック名はこのパネルで変更できます。トラック名は Follow Track コンストレイントのように、トラッキングデータと他のエリアをリンクするのに使用されます。

次にここでコントロールできるのは、マーカーのON/OFFフラグです(目のアイコンボタンを使用)。マーカーをOFFにした場合、その位置はソルバにもコンストレイントにも使用されなくなります。

目の隣のカギボタンは、トラックをロックします。トラックのロックにより、すべての編集が不可能となります。これは「完了した」(全映像において正確にトラックされた)トラックを不意に編集してしまうのを防ぐのに役立ちます。

このパネルに置かれたウィジェットは「トラックプレビュー」と呼ばれ、パターンエリアの内容を表示します。これはどれだけいい追跡が行われているかを確認(元の位置からずれをなくすのをコントロール)するのに役立ち、必要な位置にトラックを戻すのに便利です。トラックの移動はこのウィジェットをマウスドラッグすることで直接行うことができます。

もしアンカー (Anchor) が使用されていれば(画像上の追跡中の位置が、ペアレントに使用されている位置と違う)、プレビューウィジェットはアンカー位

置の周辺を表示します。このような設定は、マスクの端を置くべき場所にいい特徴がない時にマスキングするのに便利です。このテクニックの詳細は後述します。

プレビューウィジェットの下小さな領域で、プレビューウィジェットの縦の長さを大きくできます(領域は二つの水平線で強調されています)。

次の設定はチャンネルのコントロールです。トラッキングはグレースケール空間で行われるため、特徴と背景間のコントラストが正確なトラッキングを行うのに足りないことがあります。このような場合、いくつかのカラーチャンネルを OFF にすると助かることがあります。

最後はカスタムカラー (Custom Color) とそのプリセットです。これは Movie Clip エディタと3Dビューで使用される、デフォルトのマーカーカラーを置き換える設定で、違うタイプの特徴点 (例えば遠くの平面の特徴点と近くの平面の特徴点など) を区別するのに便利です。また、カラーはトラックの「グループ」にも使用することができ、Select Grouped オペレータを使用し、カラーでトラックのグループを選択することもできます。

Camera Data パネル

このパネルには現在 Movie Clip エディタで編集集中の動画を映写するのに使用されるカメラの全ての設定があります。

まず最初に、あらかじめ定義された設定を使用することができます。新しくプリセットを追加したり、使用しないプリセットを削除することも可能です。ただし、歪み係数 (Distortion Coefficient) や重要点 (Principal point) のような設定はプリセットには含まれておらず、カメラプリセットを使用しても設定はしないといけません。

- **Focal Length** は、文字通り動画が撮影された焦点距離です。mm とピクセルで指定可能です。大抵の場合、焦点距離は mm ですが、時々 (ネット上のチュートリアルなどでは)、ピクセルのこともあります。そんな場合も直接慣れた単位で設定できます。
- **Sensor Width** カメラ内の CCD センサの幅です。この値はカメラの仕様書にあります。
- **Pixel Aspect Ratio** は、CCD センサのピクセルのアスペクト比です。この値もカメラの仕様書で知ることができますが、推測することもできます。例えば、映像は 1920×1080 だけど、画像自体は 1280×1080 の場合、ピクセルのアスペクト比は以下になります。

$$1920 / 1280 = 1.5$$

- **Optical Center** は、カメラ内で使用されているレンズの光学的な中心です。大抵の場合、画像の中心と同じなのですが、いくつかの特別なケースでは違いができます。このような場合はカメラ・レンズの仕様書をチェックしてみてください。光学的な中心を画像の中心に設定するには、スライダの下の **Center** ボタンをクリックしてください。
- **Undistortion K1, K2 and K3** は、動画の撮影で生じるレンズの歪みを補正するのに使用される係数です。現在この値は Distortion モードで利用できるツールを使用し、手動でのみ調整できます (まだキャリブレーションツールはありません)。基本的に既知の焦点距離では解析まで K1 を調整するだけで、ほぼ正確になるでしょう (ただし、グリッドとグリースペンシルも「不可能な」歪みを防ぐのに考慮されます)。

Display パネル

このパネルには Movie Clip エディタ内に表示される物をコントロールするすべての設定が含まれています。

- **R, G, B** と **B/W** ボタン (一番上) は、フレームプレビューに使用されるカラーチャンネルのコントロールと、全フレームをグレースケールにするのに使われます。

トラッキングアルゴリズムがグレースケールの画像で動作するため、そしてどのチャンネルを OFF にすれば特徴点のコントラストが上がり、ノイズが減るかがいつもすぐわかるわけではないため、必要な設定です。

- **Pattern** は、トラックのパターンエリアに合致する矩形の表示を OFF にするのに使用できます。Movie Clip のビューをより鮮明にし、どれだけいいトラッキングが行われているかを確認するのに役立ちます。
- **Search** は、トラックのサーチエリアに合致する矩形の表示を OFF にするのに使用できます。Movie Clip のビューをより鮮明にし、どれだけいいトラッキングが行われているかを確認するのに役立ちます。選択中のトラックのサーチエリアのみ表示されます。
- **Pyramid** は表示するもっとも高いピラミッドレベルを表示できるようにします。ピラミッドについては Tracking Settings パネルのセクションで後述しますが、基本的にどれだけ多くのトラックを一つのフレームから他に移動できるかを判断するのに役立ちます。
- **Track Path** と **Length** は、トラックの経路の表示をコントロールします。一つのフレームのみに注目した、トラックの移動を視覚化できる方法です。トラックがその位置からジャンプしているかどうかを判断するのに便利です。
- **Disabled Tracks** は、カレントフレームで OFF になっているすべてのトラックを隠せるようにします。これはトラッキングが十分な精度で行われている時、ビューを綺麗にし、コントロールしやすくするのに便利です。
- **Bundles** は動画クリップ解析後に使用する物で、次のように動作します: 各トラックの解析された位置を動画クリップ上に投影し直し、小さな点として表示します。点の色は投影された座標と元の座標の距離に依存し、もし、充分近ければ緑に、そうでなければ赤になります。これはうまく解析されていない、調整が必要なトラックを見つけるのに便利です。
- **Names and Status** は、トラックの名前と、その状況 (キーフレームがついた (Keyframed)、OFF (disabled)、トラッキングされた (Tracked)、評価後 (Estimated)) を表示します。選択中のトラックの名前と状況が表示されます。
- **Compact Markers**。マーカーの表示方法 (黒い枠と黄色の表面色) はどんな映像上でも (明暗両方で) トラックが見えるようになっています。しかし、それが邪魔になる場合、このオプションでマーカーをもっとコンパクトに表示します。枠線が黒の破線になり、表面より上に表示することで、マーカーエリアはたった 1 ピクセルの大きさになります。
- **Grease pencil** は、グリースペンシルのストロークの表示と作成を可能かどうかを意味します。
- **Mute** は、動画フレーム自体の表示を黒の四角形に変更し、不正確な追跡が行われたトラックや、全く追跡されていないトラックを見つけるのに役立ちます。
- **Grid** (Distortion モードのみ) は、本来の正射投影に歪みモデルの影響を与えたグリッドを表示します。このグリッドは手動キャリブレーション (グリッドの歪んだラインが映像中の直線に相当) に使用できます。
- **Manual Calibration** (Distortion モードのみ) は、

グリースペンシルのストローク用の歪みモデル (Distortion model) を適用します。このオプションは手動キャリブレーションを行うのにも役立ちます。このプロセスの詳細は後日作成します。

- **Stable** (Reconstruction モードのみ)。このオプションは表示中のフレームに 2D stabilization 設定の影響を与えます。これは実際の映像自体を変更しない唯一のプレビューオプションです。
- **Lock to Selection** 再生もしくは追跡時、全映像中にわたり、選択中のトラックをエディタの同じスクリーン位置に表示します。このオプションは追跡処理のコントロールと、トラックがずれ始めた時、もしくはジャンプした時に追跡処理を止めるのに役立ちます。

- **Display Aspect Ratio** は表示のアスペクト比のみ変更します。追跡や解析プロセスには影響しません。

Tracking Settings パネル

共通オプション

このパネルには2D追跡アルゴリズム用のすべての設定があります。どのアルゴリズムを使用しているかにより別々の設定が表示されますが、共通する、すべてのトラッカー用の設定がいくつかあります。

Adjust Frames は、どのパターンを追跡させるかをコントロールし、もし重要な物であるなら、そのフレームからのパターンが追跡されます。以下にもっとわかりやすい例を挙げます。

トラッカーのアルゴリズムはサーチエリア内の二つの画像と、最初の画像内の追跡されたポイントの位置を受け取ります。そしてトラッカーは最初の画像のポイントの、二つめの画像での位置を検索しようとします。

では、シーケンスの追跡が起こる仕組みです。二つめの画像は常に、マーカの位置が分からないフレーム(次の追跡フレーム)から作成された画像です。しかし、最初の画像との違いがトラッカーに送られることがあります。以下はほぼ大半に使われる組み合わせです。

- キーフレームが打たれたトラックのあるフレームから作成された画像。この設定は(トラッカーにより返されているオリジナルパターンにほぼ一致する位置なため)オリジナル位置からのずれを阻止しますが、小さなジャンプを引き起こしたり、カメラモーションの所為(遠近による変形など)で特徴点に変形された時に失敗する可能性があります。このような設定は、**Adjust Frames** が 0 に設定されていた場合に使用されます。
- 最初の画像としてトラッカーに送られているカレントフレームから作成された画像。この状態では、パターンが隣接した二つのフレーム間でトラックを行います。特徴点が大きく変形するケースでの処理が可能ですが、オリジナル位置から大きくずれる可能性もあるため、調整が必要です。このような設定は、**Adjust Frames** が 1 に設定された場合に使用されます。

Adjust Frames が1より大きい場合のトラッカーの挙動は、**Adjust Frames** フレーム毎にトラック用のキーフレームが作成され、キーフレームが打たれた画像と次の画像が使用され、その間のトラックを行うような感じになります。

Speed は、連続トラックの速度をコントロールするのに使用できます。このオプションはトラック品質には何も影響せず、単にトラックが正確に行われているかどうかを確認するだけに役立ちます。ほとんどの場合、トラックはリアルタイム再生より大幅に速く行われ、トラックが位置からずれ始めていくのは難しいです。このような場合、**Speed** を Double もしくは Half に設定し、二つのフレームのトラック中に遅れを加えれば、ずれに早期に気づいてトラック処理をキャンセルし、トラックの位置を調整することができます。

Frames Limit は、Track Sequence オペレータが呼ばれている際、どれぐらいのフレームがトラック可能かをコントロールします。Track Sequence オペレーション毎に、最大 **Frames Limit** フレーム、トラックします。これもトラックのずれの確認と修正に役立ちます。

Margin は、トラックが画像の境界の近くになりすぎた時に、トラックを OFF にするのに使用できます。このスライダは、ピクセル数で「近すぎる距離」を調整します。

KLT tracker オプション

KLT tracker は、デフォルトで使用されるアルゴリズムです。大半の特徴点の類とその動きをトラック可能です。ピラミッドトラックを使用しており、これは次のように動作します。まず、アルゴリズムは全体的なモーションの方向を見つけるため、パターンより大きな画像をトラックします。その後、最初のステップから位置を絞り込み、最終的な位置の精度を高めるため、少し小さめの画像をトラックします。これは何度か続けて行われます。このトラックのステップ数は、**Pyramid Level** オプションと同じで、私たちが最初のステップで書いているとおり、トラックは最も高いピラミッドレベルで起こります。そのため、ピラミッドレベル=1ではパターン自身と同等になり、各レベルごとにトラック画像は2倍になります。

サーチエリアは最高ピラミッドレベルより大きくなり、サーチエリアと最高ピラミッドレベルの間の「フリースペース」は、どれぐらい多くの特徴点の一つのフレームから他に移動できるかを決めます。

ほとんどの一般的なケースでは、デフォルト設定で動作すると思いますが、時にはピラミッドレベルを変更しないといけません。例えばぼやけた映像の場合、新たにピラミッドレベルを追加することで、トラックする助けとなります。

このアルゴリズムは、特徴点の一つの方向に移動し、特徴点の周囲のテクスチャが他の方向に移動するような時は失敗することがあります。

SAD tracker オプション

各ステップにおいて、SAD tracker はサーチエリア全体を検査し、二つめの画像上に、トラック中のパターンに最も似たパターンを見つけます。これは非常にすばやく動作しますが、いくつかのケースでは失敗します。例えば、トラック中の特徴点によく似た別の特徴点があり、それがサーチエリアに現れた時で、この場合、SADは一つの特徴点から他にトラックを飛ばしてしまいます。

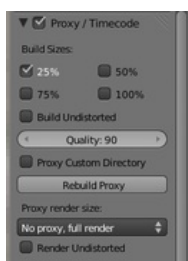
Correlation は、依然トラックが成功しているとみなす、二つのパターン間の相関関係の閾値を決めます。0で全く相関関係がないことを意味し、1で完全に相関関係にあることを意味します。

一つ制限があります。現在、16×16の特徴のみ動作します。

Marker パネル

このパネルには、マーカ位置、パターンとサーチエリアの寸法、パターン中心からのアンカーポイントへのオフセットの数値設定があります。すべてのスライダは名前どおりの設定です。

Proxy / Timecode パネル



Movie Clip エディタ 内の Proxy / Timecode パネル

このパネルにはプロクシイメージ(Proxy Image)と、動画のタイムコード(Timecode)に使用されるオプションがあります。

プロクシイメージで、Movie Clip エディタ内に低解像度で画像を表示することができます。これは少ない RAM 量のマシンで4K映像のトラッキングを行うような場合に便利です。

最初の4つのオプションは、生成するプロクシイメージの解像度を設定するのに使用します。現在、オリジナルサイズの25%、50%、75%、100%が作成できます。100%のサイズの Proxy は、デコードできなかった、壊れたフレームを含む動画に使用できます。

Build Undistorted は、プロクシイメージビルダが歪んでいない元の画像から上記で設定したサイズの画像も作成します。これは歪んでいない映像をすばやく再生するのに役立ちます。

生成されたプロクシイメージは、JPEG を使用してエンコードされ、JPEG コーデックの品質は、**Quality** スライダーによってコントロールされています。

デフォルトでは、生成されたすべてのプロクシイメージは、<元映像のパス>/BL_proxy/<クリップ名> フォルダに格納されますが、この場所は **Proxy Custom Directory** オプションを使用することで手動で設定できます。

Rebuild Proxy は、上で設定されたすべてのサイズのプロクシイメージと、後で使用できる全タイムコードを再生成します。

Use Timecode Index は、動画ファイルに使用でき(そしてよく使用され)ます。基本的に、タイムコードはフレームの検索を高速化し、精度を高めまます。カメラとコーデックによっては、別のタイムコードでいい結果になることもあります。

Proxy Render Size は、表示に使用されるプロクシサイズの解像度を決めます。もし **Render Undistorted** が設定された場合、画像は使用されている歪みのないフレームからイメージを作成します。もし何もプロクシが生成されていない場合、レンダリングサイズは "No proxy, full render" に設定され、歪み補正レンダリングが有効になり、フレーム描画時に自動的に歪み補正されます。

Reconstruction モードで利用可能なツール



Proxy / Movie clip エディタ内の2D Stabilization パネル

Reconstruction モードで利用可能な新たなパネルに、2D stabilization パネルがあります。

このパネルはショットの2Dスタビライゼーションに使用されるデータを定義するのに使用します。オプションがいくつかこのパネルで使用できます。

まず最初に、カメラのジャンプや位置を補うトラックのリストがあります。これは次のように動作します: 位置のスタビライゼーションに使用されるトラックリストからトラックを取得し、最初のフレーム上のこれらのトラックすべてから中間点を見つけます。各フレームごとに、アルゴリズムがフレーム全体の動きから、スクリーン座標上で同じ位置を持つこのポイントを作成します。

時にはカメラのジャンプを完全に補正する必要がないことがあります。そんな場合、**Location Influence** が使用できます。

また、カメラが少し回転し、その映像が若干傾いていることもあります。**Stabilize Rotation** オプションはこの傾きを補正するものです。これを設定するには新たなトラックが1つ必要になり、次のように動作します: 動画の最初のフレームで、このトラックが前述のリストからの複数のトラックの中心点につながられ、水平線とこの線分との角度が全映像中に渡って一定に保たれます。その映像に適用される回転量は、**Rotation Influence** で調節できます。

もしカメラが大幅にジャンプしていれば、画像の境界近くに黒い領域が目立つことになります。このブラックホールを取り除くために、最小スケールを探す **Autoscale** オプションがあり、その映像にこれを適用すると画像の境界の近くのすべてのブラックホールが削除されるでしょう。最大スケールファクターをコントロールするオプション (**Maximal Scale**) と、その映像に適用するスケール量のオプション (**Scale Influence**) もあります。

レンダリングエンジン

- Blender Internal (Blender 内蔵)
- [Cycles](#)

レンダリング

レンダリングは CG の最終工程で(ポストプロセスを除く)、この段階になってようやく、作成した3次元のシーンに対応した2次元の画像が作られます。レンダリングは CPU 負荷の高い処理です。自分のコンピューターで 1枚ずつ画像をレンダリングすることもできますし、PC が分担作業するためのネットワーク、レンダーファーム (render farm) を利用して画像の部分ごと、またはフレームごとにレンダリングすることもできます。画像やアニメーションの作成過程に関わる Blender の全機能を、この章で説明します。

マテリアル、テクスチャ、照明、カメラの用意ができれば、レンダリングを始められます。最初のレンダリングで結果に満足する可能性は低いので、試しレンダリングを何度も行うつもりでください。この章では、目的の画質を達成するためのレンダリング工程用オプションと設定について説明します。

Blender には内蔵のレンダリングエンジンがあります。高速なレンダーで、うまく調整すれば良い結果を生みます。もっと高度な描画機能を持つ、他の複数の外部レンダーを読み込みます。

世界各地の Blender ユーザーは、処理能力の大きく異なる PC を使っています。CG においてレンダリングは、CPU や ディスク容量を湯水のように消費することのある工程です。特に共同で作業している場合には、10時間の DVテープをアップロードしたり、編集作業を行ったりしていると、テラバイトの容量があるサーバーを簡単に使い尽くします。このため、小さな PC に大仕事をさせるやり方がいくつもあります。画質を維持しつつ可能な限り火力を上げるために複数のオプション設定が用意されています。このページでは Render パネルにある主要なオプション機能を説明し、他はその後説明します。

概観

現在のシーンをレンダリングするには、Render (レンダー) パネルにあるアイコンつきの **Render (レンダー)** ボタンを押すか、F12 を使います。レンダリングされた画像の表示方法は [Output \(出力\) オプション](#) で指定できます。[レンダーウィンドウ](#) もご覧ください。

動画はアイコンつきの **Animation (アニメーション)** ボタンを押すと作れます。Output (出力) ディレクトリに、指定したファイル形式で保存されます。通常は一連のフレーム画像として出力します。[ビデオ出力](#) や [アニメーション](#) をご覧ください。

画像は Dimensions (寸法) パネルで指定した寸法でレンダリングされます。

作業の流れ

一般に、レンダリングには以下の工程があります:

1. シーン内にすべてのオブジェクトを作ります
2. [シーンに照明を当てます](#)
3. [カメラを配置します](#)
4. テスト画像を作ります。高速化して待ち時間を減らすため、アンチエイリアスやレイトレースなどを省き、25% ぐらいのサイズでレンダリングします
5. マテリアル/テクスチャや照明の設置と調整を行います
6. ある程度満足する質になるまで、上記手順を繰り返します
7. 修正箇所を徐々に少なくして、**だんだん**と高品質なフルサイズの画像をレンダリングします。演算時間も徐々に長くなります
8. 画像を保存します

分散型レンダーファーム

多くの頭脳を結集させれば、レンダリング時間全体を減らすことができます。CPU の割り当てにはいくつかレベルがあります。

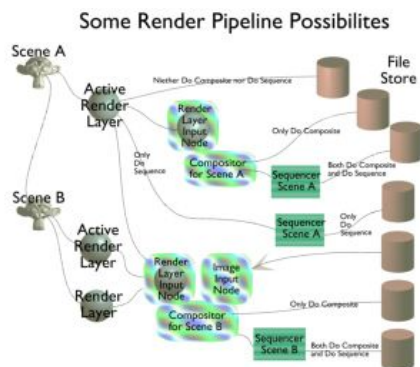
レベル1。マルチコアの CPU があれば、スレッド数を増やすと Blender は その数の CPU をレンダリングの計算に使います。

レベル2。LAN に繋がった 別の PC が利用できれば、作業をフレームごとに分割することができます。例えば 200フレームのアニメーションをレンダリングするなら、同程度の性能の PC が5台あれば、1台目にフレーム 1-40、2台目に 41-80、といった具合に分担します。他より遅い PC には、単に少なめにフレームを割り当ててください。LAN レンダリングを行うには、.blend ファイルのあるフォルダーを共有ドライブに設定します (テクスチャなどの外部ファイルは Pack (パック/梱包) しておく必要があります)。各 PC で Blender を起動し、この .blend ファイルを開きます。PC ごとに開始と終了フレームを設定してレンダリングを開始しますが、.blend ファイルの保存は行わないでください。出力パスに相対パスを指定すれば、レンダリングしたフレームはホスト PC に保存されるでしょう。

レベル3。WAN レンダリングを行うことができます。.blend ファイル (Pack した状態で!) をインターネットを介してメールかファイル共有か Verse-Share して、レンダリングの一部を誰かの PC に行ってもらいます。作業が完了すれば、できたフレームをメールで送ってもらえるでしょう。信頼できる友人がいれば、この方法で協力することができます。

レベル4。レンダーファームのサービスを利用できます。BURP (Big and Ugly Rendering Project) のようなレンダーファームサービスは組織運営されています。.blend ファイルをメールで送ると、レンダリングは複数台の PC に割り振られます。BURP を例に出したのはこのサービスが無料で、Blender ユーザーである会員の PC を BOINC 型のバックグラウンド処理で使用するためです。他のサービスは定期契約型課金や、従量型の課金を行っています。

レンダリング作業台の統合



レンダリングのパイプライン処理の例

Blender には3つの独立したレンダリングの作業台があり、画像処理はパイプライン内をひとつずつ順番に流れます:

- レンダリングエンジン
- [コンポジター](#)
- [シーケンサー \(VSE\)](#)

どれか一つを個別に使うことも、連続した工程として使うこともできます。例えばシーケンサー単体を動画ストリームのポストプロセスに使えます。コンポジター単体を画像の色調補正に使えます。シーンの画像をアクティブなレンダーレイヤーに従って計算し、コンポジターやシーケンサーを使わずに、そのまま保存できます。図の上部右側にこの可能性が示されています。

複数シーンを直接、または中間ファイルを介して Blender 内でつなげ、レンダリングすることもできます。各シーンはレンダーレイヤーを複数持つことができ、各レンダーレイヤーはコンポジター内でミックスされます。アクティブレンダーレイヤーとは、表示されていて、アクティブにチェックがされているレンダーレイヤーを指します。もし表示されているレンダーレイヤーにアクティブ/有効のチェックがされていなければ、一覧で次にチェックのあるレンダーレイヤーが画像の算出に使われます。コンポジティングとシーケンサーが無効なら、画像は最終的なレンダリング結果として表示されます。

Compositing (コンポジティング) が有効なら、レンダーレイヤーはコンポジターに送られます。ノードは画像を操作して、保存が可能な Composite 出力に送ります。もしくは、*Sequence* が有効ならシーケンサーに送られます。

Sequencer (シーケンサー) が有効なら、コンポジターが有効ならコンポジターの結果が、有効でなければアクティブなレンダーレイヤーがシーケンサーの Scene ストリップに送られます。そこで VSE の設定にしたがって操作が行われ、最終的にそのシーン用の画像ができます。

.blend ファイルに複数のシーンがあると、話は少し複雑になります。例えばシーン A とシーン B があるとします。Compositing が有効なら、シーン B のコンポジターにある Render Layer ノードはシーン A のレンダーレイヤーを入力にできます。これはポストプロセスがされていない画像であることに注意してください。もしシーン A からコンポジットやシーケンス済みの画像を取り込みたい場合は、シーン A のレンダリング結果をコンポジターやシーケンサーを使ってファイルに出力しておいて、シーン B のコンポジターでそれを入力にした Image Input ノードを使います。

図の下半分は、Blender の極地を示しています。シーン A の、ポストプロセスの施された画像と動的成分のレンダーレイヤーを、シーン B からの二つのレンダーレイヤーとコンポジター上で混ぜ合わせ、最終的にシーケンサーを通して観賞用に保存しました。

こうした例は Blender を使ってできることのほんの一部に過ぎません。このまま読み進めてすべてのオプションについて学び、独自のワークフロー開発の創作性を養ってください。

Render 設定パネル

内蔵のレンダリングエンジンのオプション機能、または外部エンジンを使っていればそのオプション機能は、すべて Render タブにあります。

Render

(レンダー)

ここで [静止画](#) や [動画](#) をレンダリングできます。

画像がレンダリングされる場所を選ぶこともできます。これは [レンダリングの表示](#) で説明されています。

Layers

(レイヤー)

レイヤーメニューには、[レイヤー \(eng\)](#) や [パス \(eng\)](#) を使ったレンダリング用のオプション機能があります。

Dimensions

(寸法)

レンダリング画像の大きさの設定 ([出力オプション](#) をご覧ください) と、連続画像のレンダリングのオプションがあります ([アニメーション](#) をご覧ください)。

Anti-Aliasing

(アンチエイリアス)

[アンチエイリアス](#) は、ジャギー (ピクセルのぎざつき) のない高品質なレンダリングを行うために重要です。

Motion Blur

(モーションブラー)

[モーションブラー](#) は、動きのある画像をレンダリングするときに重要なエフェクトです。各フレームが完全な静止画になっているストップモーションで見られるような、現実味に欠けた不自然な見た目を回避します。

Shading

(陰影処理)

レンダリング時に演算される陰影効果を制御するオプションです。選択を解除すると無効化できます。

- [Textures](#) (テクスチャ)
- [Shadows \(eng\)](#) (影)
- [Subsurface Scattering](#) (表面下散乱)
- [Environment Maps \(eng\)](#) (環境マップ)
- [Ray Tracing \(eng\)](#) (レイトレース)
- [Color Management](#) (色調補正)

有効にするとニアなワークフローを使います

- [Alpha](#) (アルファ)

透明なピクセルのレンダリングのしかたを設定します。

Output

(出力)

画像をレンダリングする場所やファイル形式を決めます。[出力オプション](#) をご覧ください。

Performance

(パフォーマンス)

コンピュータのメモリやプロセッサに関連した、レンダリング処理性能を調整します。[パフォーマンス](#) をご覧ください。

Post Processing

(ポストプロセス)

画像がレンダリングされたあとで用いられる効果を調整します。[コンポジター](#) か [シーケンサー](#) を使っていれば、シーンを直接レンダリングする代わりに、この効果を与えることができます。

Fields は [動画のレンダリング](#) で使われます。

[Dithering \(ディザ処理\)](#) はピクセルをぼかします。

[Edge Rendering](#) を有効にして、手描きもしくはコミック調の効果を作ることができます。

Stamp

(刻印/スタンプ)

[Stamping](#) を使って、レンダリングした画像にテキストを重ねたり、対応形式 (PNG、JPEG および EXR) なら stamp メタデータを画像に入れたりできます。

Bake

(焼き込み/バイク)

[Render Baking \(バイク\)](#) は、照明、影、色の情報などの理想のレンダリング効果を持たせたテクスチャファイルを作る処理です。陰影処理の計算を省くことで恩恵を受けられるような、リアルタイムの画像生成で役立ちます。

Freestyle

バージョン 2.67 から Blender に同梱されている、非写実的な表現のためのレンダラーです。詳しくは [Freestyle](#) をご覧ください。

The Camera

(カメラとは)

カメラは Blender から画像をレンダリングする手段を提供するオブジェクトです。カメラはレンダリングの際に、シーンのどの部分が見えるのかを定義します。シーンには複数台のカメラを設置できますが、何かしら絵を生成するためには、最低一台のカメラが必要です。

Add a new camera

(新たなカメラの追加)

モード: Object モード

ホットキー: ⇧ ShiftA

メニュー: Add (追加) » Camera (カメラ)

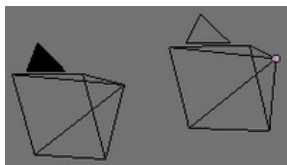
Object モードで ⇧ ShiftA を押し、ポップアップメニューで Add » Camera を選びます。

Change active camera

(アクティブなカメラの変更)

モード: Object モード

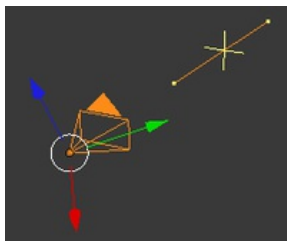
ホットキー: Ctrl0 NumPad



アクティブカメラ(左)

アクティブカメラとは現時点でレンダリングとカメラビュー (透視投影は現実のカメラで撮られた絵に似ていますが、平行投影はもっと専門的なビューで見取り図に最適ですが、オブジェクト間の距離を伝える用途には向いていません)。

これらの投影を設定するには [このページ](#) の消失点や等角投影をご覧ください。
Panoramic (パノラマ状) は円柱投影を使ってシーンをレンダリングします。



クリッピング範囲と焦点を表示したカメラ

- Focal Length (焦点距離)

透視投影とパノラマ状カメラで利用でき、レンズの焦点距離を度かミリメートルで表現します。平行投影モードのときは Focal Length (焦点距離) 設定が Orthographic Scale (平行投影のカメラスケール) に変わります。この設定はカメラから見える領域の大きさを決めます。

- Shift X/Y (X/Y をずらす)

カメラの描画域をずらします。なお Shift 値は実際のカメラ位置に相対的なもので、実際のカメラ位置は変えませんが、したがって普通はこの設定はカメラ位置の調整に使うべきではありません。

- Clipping Start/End (クリップ範囲)

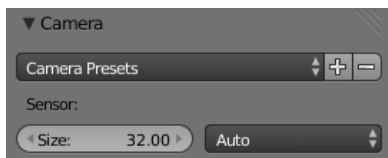
クリップ範囲を指定します。範囲内にあるオブジェクトだけがレンダリングされます。Limits を有効にするとカメラの視線上に二つの黄色の点でクリップ範囲が描かれます (図「クリッピング範囲と焦点を表示したカメラ」)。

註釈

3D ビュー ウィンドウのプロパティシェルフ(N)にもカメラと同じ Orthographic/Perspective や Clip Start/Clip End のような設定があります。こうした設定はカメラのレンダリングには影響せず、カメラビュー以外でのビューの設定を変えるだけです。詳細は [3D ビューのオプション](#) をご覧ください。

Camera Presets

(カメラプリセット)



カメラプリセットパネル

ToDo

- Camera Presets
- Sensor

Depth of Field

(被写界深度)



カメラ被写界深度パネル

- Depth of Field オブジェクト

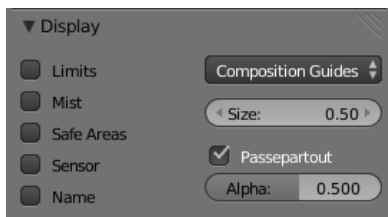
[Depth of Field](#) を使っているとき、リンクしたオブジェクトが焦点を決めます。オブジェクトをリンクすると Distance (距離) 欄は利用できなくなります。

- Distance (距離)

焦点距離です。カメラの視線上に黄色の十字で表示されます。Limits が有効でないと十字は見えません。[Defocus Compositing Node \(eng\)](#) と組み合わせて使います。

Display

(表示)



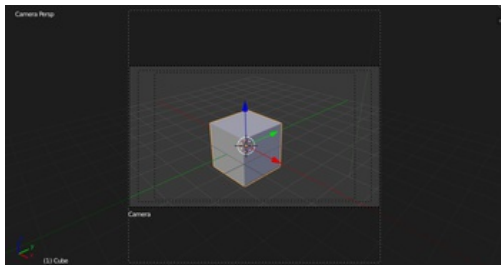
カメラの表示パネル

- Limits (リミット)

クリッピング範囲と焦点の表示非表示を切り替えます。

- Mist (ミスト)

Mist (霧) の見える範囲を有効化/無効化します。範囲はカメラの視線上に白い二つの点で表示されます。Mist の範囲は World の Mist セクションで設定します。



セーフエリア、センサー、名前を表示しているカメラビュー

- Safe Areas (セーフエリア)

有効化するとカメラビューのときに点線の枠が追加され、重要なものを置くのに「安全」とみなせる領域を区切ります。

- Sensor (センサー)

カメラビューに点線の枠を表示します

- Name (名前)

名前の表示/非表示を切り替えます

- Size(サイズ)

3Dビューのカメラアイコンの大きさです。カメラのレンダリング結果には影響せず、見た目の設定です。カメラアイコンの拡大縮小はいつものように S キーでも行えます。

- Passepartout(外枠)、Alpha(アルファ)

カメラの視野の外を Alpha の設定に基づいて暗くします。

Composition Guides

(構図ガイド)

ドロップダウンメニューから利用できる Composition Guides は撮影の組み立てに役立ちます。八種類のガイドが利用できます:

- Center

枠の縦横それぞれを二分割する線を引きます。

- Center Diagonal

対角線を引きます。

- Thirds

枠の縦横それぞれを三分割する線を引きます。

- Golden

縦横を黄金比で分割します(枠の四辺から、幅か高さ×約 0.618 の位置)。

- Golden Triangle A

左下隅から右上隅に対角線を引き、その線に左上隅を通る垂線と右下隅を通る垂線を引きます。

- Golden Triangle B

Aと同じですが、反対側の隅を使います。

- Harmonious Triangle A

左下隅から右上隅に対角線を引き、左上隅と右下隅からそれぞれ、対辺の長さ× 0.618 の位置まで線を引きます。

- Harmonious Triangle B

Aと同じですが、反対側の隅を使います。

Camera Navigation

(カメラ操作)

シーン内でカメラ位置を簡単に操作する方法を紹介します。

註釈
どの種類のオブジェクトもアクティブな「カメラ」になれることを忘れないでください。例えば光源の位置や方向を決めるのにこの操作を使うことができます

Move active camera to view

(ビューにアクティブカメラを合わせる)

モード: Object モード

ホットキー: CtrlAlt0 NumPad

アクティブカメラの位置や方向を、いまの視点に合わせる機能です。

カメラを選択し、(カメラではなく)3Dビューを目的の位置や方向に動かします。CtrlAlt0 NumPad を押すと、カメラ位置と方向がいまの視点に合わせられ、カメラビューに切り替わります。


Camera View Positioning

(カメラビューの位置操作)

カメラビューでプロパティパネル(N)の View パネルにある Lock Camera to View を有効にすると、カメラビューのまま 3D ビューポートをいつものように操作できます。操作方法は 3D で移動を行うときとまったく同じです。

Roll, Pan, Dolly, and Track


(ロール、パン、ドリー、トラック)

カメラでこの動きをするには、まずカメラがアクティブオブジェクトになるように**選択されている**必要があります(カメラを通じて見ているときは、枠の辺を RMB -クリックして選びます)。以下の操作はカメラビューにいることを想定しています(0 NumPad)！ そうすれば、オブジェクト操作に使われるのと同じコマンドを使ってカメラを操作できます：

Roll: R を押してオブジェクトの回転モードに入ります。デフォルトではカメラをローカルの Z軸(カメラビューに直交する軸)を中心に回転します。カメラロールとはこの動きを指します。

Pitch(縦方向の Pan): ローカルの X軸を中心とした回転を指します。R を押してオブジェクトの回転モードに入り、X を二回押します(一回目で *グローバル* 軸を選び、二回目で *ローカル* 軸を選びます。この操作はどの軸でも行えます。[軸ロック](#) をご覧ください)。

Yaw(横方向の Pan): カメラのローカルの Y軸を中心とした回転に対応しています。そうです、それです、R を押して Y 二回！

Dolly: カメラをドリーするには G を押して MMB  を(または Z を二回)押します。

Sideways Tracking: G を押してマウスを動かします(X か Y を二回押して純粋な水平/垂直方向への移動撮影を行えます)。

Aiming the camera in Flymode

(飛行モードでカメラを向ける)

カメラビューにいるときに [飛行モード](#) は、実際にアクティブカメラを動かします…。

[\[video link\]](#)

Perspective in Render

(レンダリングにおける視点パース)

F12 を押ししてレンダリングすると、カメラの「視点 (perspective)」を通して見た絵ができます。3D ビューで上、前、横、あるいは自分で決めた視点からモデルを *見る* ことができるように、さまざまな視点でものを *描く* ことができます。その絵を作る上で、視点はレンズのサイズ、レンズの種類、オフセットを考慮します。視点によって消失点の数は変わります。

3D の立方体を見ると、3種類の辺が見えます: 縦方向、横方向、奥行き方向です。縦の辺がすべて平行なら、これらの辺に対して消失点はありません。しかし平行でなければ、定規を使って延長すると、ある同じ点で交差します。この点を消失点 (vanishing point) と呼びます。

特殊な用途向けに、別の種類のレンダリングカメラを据えて視点を変えることができます。後述する理由により、特に建築設計においては消失点の数を制限したくなるかもしれません。建築家と製図担当者にはものや建物を正しい寸法と相対比率で描く責任があります。

レンダリング例では建物は様々に歪んで見え、まるで泥でできていて崩れそうに見えます。もし建築業者にこの建設を頼めば、ほんとうに上に行くほど狭くなる、傾いた壁や部屋の建物を見るはめになるでしょう。

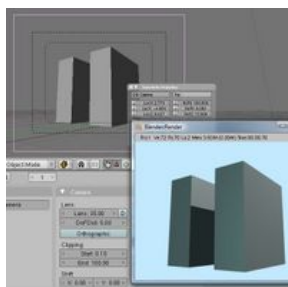
古代ギリシャ時代にさかのぼれば、高い柱は当初、柱を見上げたときに両側の辺が上下にまっすぐに見えるよう、最上部が最下部よりも太くなるように作られました。その後、建物が実際より高く見える錯覚を狙って、柱の最上部は細くなりました。ルネサンス時期には、芸術作品に消失点を利用する考え方が発展しました。Blender には、同じことをするための独自の方法がいくつか用意されています。

註記

以下の章の理解には [カメラの設定](#) の知識が必要です

Three Point Rendering

(3点透視図法)



通常の3点透視図法

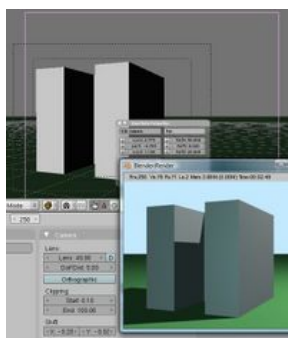
背の高い建物を、通常の 35mm カメラを使って地表レベルの片側から見上げる (あるいは見上げた視点でレンダリングすると)、3点透視図法になります。垂直方向の線に沿ってそれぞれ定規を置くと、建物の上のある点で収束します。

水平方向の線は片側に収束し (この例では左側)、奥行き (後退) 方向の線は別の第三の点に収束します (この例では右下側に寄ったどこかで)。消失点が3つあるため、3点図法と呼びます。

これは事実で何も間違っていない。次に外にでるとき、背の高い建物を見てみてください。そこで実際に見えるものと同じです。ただ、あなたは頭の中では建物は四角いことを知っていて見え方を補正できるので、建物が倒れる心配をしないですみます。

Two Point Rendering

(2点透視図法)

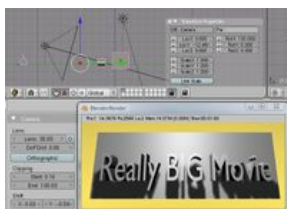


2点垂直図法

通常の完成予想図 (architectural rendering) は2点透視図法と呼ばれています。垂直方向の線が平行で、水平方向の線をたどると1点に収束していて、奥行き (後退) 方向の線が第2点に集中しています。

建築家がこの2点図法を好むことが多いのは、建物の側面が完全に垂直で、内側に倒れそうには見えないからです。印刷紙や画面の線も同様にまっすぐであるとすれば、組版や構造図にもとても良い図法です。

2点透視図法では、前もってカメラが地平線と同じ高さになるようにする必要がありますが、この結果建物の上半分が切り取られ、地平線がど真ん中にあるためとても退屈な見た目になります。建築写真家は「ソフトレンズ」を使ってこの問題を解決します。ソフトレンズは絵をフィルム上の別の場所に移します。



2点水平図法

このテクニックはふつうの大きさのものだけでなく、高い建物にもうまく働きます。

ほとんどの場合、二つの消失点は水平方向および奥行き方向の線上にあり、垂直方向の線は平行になります。ただ、平行な水平線を持ち、垂直および奥行き線が消失点を持つ作品もあります。これは作品の量感や高さを脚色し、誇張します。

この効果を得るにはカメラを地表レベルの中央に置き、角度を上向きに合わせ、額縁 (passepartout、実際にレンダリングされる範囲を示す、カメラビューの明るい部分) を Shift の Y 値を使って下げます。図「2点水平図法」では、カメラは作品の地表レベルにあり、上向きに30度回転しています。短い減衰を持つ明るいキーライトは中央では明るく、両端に向かうにつれて減衰し、奥行きをさらに強調して印象的な照明を作ります。

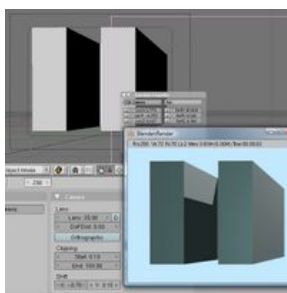
To achieve 2-point rendering:

(2点透視図法を極める)

- 短い広角レンズ、たとえば レンズの Size が 10 mm のものを建物の近くに置か、建物から遠離れた位置に長いレンズを置きます。二つの違いは描かれる建物の奥行きに影響し、レンズが長いほど建物は薄くなって動きに欠けるか、いびつに見えます。図は 40mm のレンズを使っています。
- カメラは対象の片側にずらし、建物の垂直方向の辺の歪みを最小限にするため、建物の中腹に置きます。この垂直位置 (Z値) は地表レベルより少し高くしても、あるいは (ものや建物の上部を見せたいなら) 頂上より高くしても構いません。建物の正面下部の隅を外へ広がって見せるには、カメラを上を上げます。
- カメラの角度を変えて建物からそらしします。高さを地平線とちょうど同じ高さにして、上下にずれないようにします (ちなみに図の Z角度 は 20度です)。これで垂直方向の線が平行になります。カメラを対象に向けるほど水平方向の線の消失点が近くなり、消失点が近づくことで、知覚される奥行きも深まります。
- 垂直方向の線が遠近両方で垂直に上下して見えるように、カメラの角度をわずかに (1度程度) 下げる必要があるかもしれません。線が曲がっていればもっと長いレンズを使ってください。垂直の測定には、3Dビューをカメラビューにして、額縁 (passepartout) やモニターのパixelを使ってください。
- カメラを対象から近づけ/遠ざけて、対象がコーナー近くに、正しい大きさで表示されるようにします。
- Shift: X と Y 設定を調節して、対象を正しく配置します。

One Point Rendering

(1点透視図法)



One Point Render

1点図法では垂直および水平の線が平行になり、奥行き線が1点に集中します。正面にある面が水平な長方形になることと、建物が遠くに後退していくらか奥行きがあるように見えることから、建築家はこの図法がほんとうに好きです。

カメラが建物の底と同じ高さになるように地表レベルに置かれると、実に印象的に見える一方で、変わった感じの規則正しさがあります。タイトル画像がこの方法で描かれることがあります。

To get 1-point (1pt) renders

(1点透視図法を極める)

- 奥行き線をもっと印象的にするために、カメラに短焦点の広角レンズを使います。たとえば Size 10 mm は建物に非常に近いレンズです。ふつうの見た目に近づけたい場合は 35mm レンズを使い続けてください。
- カメラは対象の片側にずらして、最上部よりわずかに高くなるように置か (対象の最上部を見たい場合)、または地表レベルに置きます。図のカメラはほぼ地表レベルにあります。地表レベルより 下にカメラを置くと、底部の奥行き方向と水平方向の線がよく溶け込み (一致して)、とても印象的な効果を生みます。
- カメラの背中がまっすぐに見えるように直立させ、対象の垂直方向の線が平行になるようにします。水平方向の辺も平行になるように、カメラの Z軸を対象に向けて 少しだけ回します。技術的には、視差を修正していることとなります (ガールフレンドを感心させる程度の雑学ですが)。図では対象に向けてカメラを 0.5度回転させています。

- カメラを対象から近づけ/遠ざけて、額縁 (passepartout) に対してふさわしい大きさに見えるようにします。
- Shift: Y 設定を調節し、額縁の底 (建物の前にある地面に幾分寄せたい場合はタイトルの高さ) が建物の底と同じ高さになるようにします。X 設定を調節して建物が中央になるようにします (または中央からわずかにずらすことで趣きのある見た目にしたり、隣にある駐車場を見せたりします)。

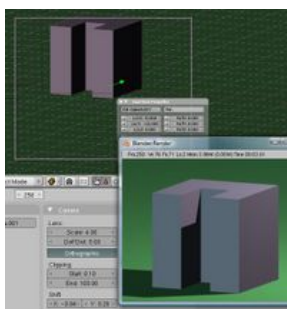
例のスクリーンショットではレンズは 35、X は負の値、Y は正の値です。カメラは建物の底と同じ高さにあり、対象の右にずらしています。X と Y をゼロにすれば、建物はカメラから外れて、額縁の左上隅にずれて見えるでしょう。

水平方向の平行な辺

水平方向の辺が平行になるようにカメラを回す際に、額縁 (passepartout) の線をガイドとして使うことができます。

Orthographic Rendering

(平行投影図法)



Orthographic Render

ゼロ点図法では、垂直、水平、奥行きの線がどれもそれぞれ平行になるもので、一般的に45度、30度、または60度の角度で描かれます。同じ組の辺はどれもお互いに平行になるため、消失点はありません。

(訳注: 図はひとつ前のバージョンでは等角投影図になっており、原文にその画像の説明が残されていましたが、その箇所は省きました)

平行投影図法ではものの形を非常に正確に描画します。3D ウィンドウのユーザービューからの眺めは、平行投影の視点です (View > Orthographic (平行投影) がオンの場合)。

平行投影を作るには次の手順を踏みます:

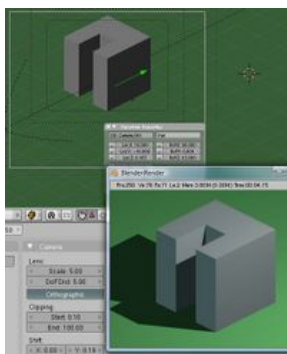
- カメラパネルで Orthographic (平行投影) を有効にします。これで面が少なくとも一つ、カメラと向かい合わせになります。
- カメラを対象に向けます。
- カメラ位置からスケールを変えて、対象が望むサイズになるようにします。

ただ、平行投影のカメラに関しては光線が視野からカメラに収束しないため、レンズのサイズは無意味です。光線は平行に入るので、カメラサイズの拡大縮小を行なって、この巨大な平面を取り入れる大きさだけを変えることができます。

平行投影は通常、対象に対して 30、45、または 60 度の角度で作られます。具体的な寸法書きは三角形の数学をよく使う読者に任せます。

Isometric Rendering

(等角投影図法)



Isographic Render

いまのうちに、等角投影図法も見ておきましょう。等角投影図法は非常に特殊な平行投影で、製図や3人称視点のコンピューターゲームで非常によく使われます。

等角投影図法では奥行き方向および水平方向の線が、地平線に対して30度をなすようにします。そして垂直方向の線は、そうです、垂直になるようにします。[Wikipediaにある複雑なベクトルの計算\(英文\)](#)には、便利なものがあります。等角投影図法を作るには次の手順を踏みます:

- カメラを Orthographic (平行投影) にします

- カメラに対して、プロパティエディタの Constraints コンテキストで、"Track To" constraint(トラック制約)を作ります。対象オブジェクト欄に名前を入力してオブジェクトを追従させ、To:(先)を -Z、Up:(上)を Y にします。
- XY平面でオブジェクトから45度の位置になるようにカメラを置き、オブジェクトから30度見上げた角度にします。オブジェクトが XYZ (0,0,0) にあるとすれば、カメラは (10, -10, 10)、または左側面からの視点になる (-10, -10, 10) になければなりません。
- Scale (平行投影のスケール)パラメータでカメラを拡大縮小し、オブジェクトが額縁 (passepartout)におさまるようにします。Scale (平行投影のスケール)はプロパティエディタの オブジェクトデータ コンテキスト、Lens (レンズ) パネルにあります。
- Shift を調節します。オブジェクトが中央に来るように Y 値を合わせます。

Depth Of Field (DOF) Explained

(被写界深度とは)

現実世界のカメラのレンズや人間の眼球は、光を屈折するレンズ(角膜)や光量を制限する虹彩に光を通過させて、映像、CCD/Cmos センサーや網膜上の像に焦点を合わせます。レンズや虹彩の相互作用により、一定距離だけ離れているものには焦点が合い、それより前や後ろにあるものには焦点が合いません。この距離をカメラや目からの深度 (depth) または「Z」距離と呼んでいます。

(現実世界では)レンズに、光はある方向から一定の角度で入ります。視点によって、見えるものは変わります。近づくほど、ものをさまざまな角度から見るができます。建築用の図面や敷地図のような「平らな」絵を作るために、Blender は平行投影 (orthographic) の描画も行えます。したがってレンダリングには透視投影 (perspective) と平行投影 (orthographic) という二種類があります。透視投影は、視野からレンズに一定の角度で入る光をシミュレートします。平行投影は、無限大の背景平面または平らな網膜に、まっすぐに入る光をシミュレートします (デフォルトでは無効にされています)。



ものに焦点が合う(距離の)範囲は、虹彩の直径によって変わります。カメラでは虹彩の直径を「F値 (f-stop)」で操作します。言い換えると、あなたには上下左右に見えている **視野** があります。お望みならばあなたの描く「絵」にもあります。目からある一定の範囲、または **深度** だけ目から離れたものには、焦点が合います。たとえば、あなたが夜間、10~15フィート(3~5メートル)離れたものに焦点を合わせることができるものとして。10フィートより近いものや15フィートより遠いものはぼやけます。このときあなたの **被写界深度 (depth of field)** は5フィート(2メートル)です。

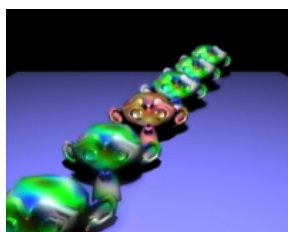
虹彩が大きいくほど、被写界深度は小さくなります。日中、遠くに広がるたくさんものに焦点を合わせられるのはこのためです(日中は虹彩が小さくなります)。映画の撮影では、焦点が完璧に合っていることを確かめるために、カメラと俳優の鼻の距離を測る役目の人がいます。

この完璧な深度 (**焦平面 (focal plane)** と呼ばれます) から離れるほど、ものはぼやけて見えます。実は、焦平面の前後にある、ものがぼやけていないとみなせる範囲が被写界深度です。Blender では、この距離を Dof Distance、「Depth of Field Distance」と呼び、カメラのオブジェクトデータパネルで調節できます。別の方法として、Dof Object 欄にオブジェクト名を入力すれば、オブジェクトにカメラの焦点を自動的に合わせ続けることができます。

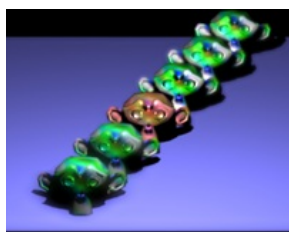
Field of View and Lens Size

(視野とレンズのサイズ)

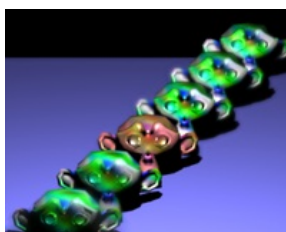
視野はレンズの大きさによって変わります。カメラでは 35mm レンズがだいたい標準サイズです。これはこのレンズの撮る写真が、目から見える絵に似た大きさであること、やや近くから撮影できることが理由です。Blender では、Camera の設定でレンズのサイズを変更します (35mm がデフォルト)。遠く離れた写真を撮る長いレンズも同じ視野を持ちますが、多くの演出家が好む独特のパスを持っています。カメラが演者から遠く離れているため、場面は「凝縮され」、一帯が平坦になります:



10 units 離れた位置からの
35mm レンズ



60 units 離れた位置からの同位
置/回転の 210mm レンズ



50 units 離れた位置の
210mm. 35mm の撮影に似た
絵が枠に入るよう位置調整

Zooming in Blender

(Blender のズーム操作)

ズームは絵の一部を広げる機能です。人はこういったことはできません。いえ、前言撤回して...我々もできます。我々はソファから起き上がって、見たいものに近づけます(しかしこれは「ズーム」と言うより「移動」のようなものです)。Blender ではどちらの操作も行えます。移動撮影 (track in/out、あるいは truck (トラック) in/out) のためにカメラを近づけたり遠ざけたりすることも、レンズのサイズを変えることも、その両方を使うこともできます。これらの動作はそれぞれ、カメラやオブジェクトに補間曲線 (interpolated/lpo curve) を割り当てて自動化できます。

Depth of Field in Computer Graphics

(コンピュータグラフィックスにおける被写界深度)

コンピュータグラフィックス(CG)においては、物理的なレンズや虹彩は存在せず、被写界深度 (depth-of-field) は無限大で、常にすべてのものに焦点が合います。しかしながら美術的な理由により、主人公に焦点を合わせて他は少しぼかし、背景に視聴者が目を合わせて気が散らないようにしたい

ことがあります。また、主人公だけに焦点が合えば、主人公を見分けやすくなります。このため、あるエフェクトを—すなわち **被写界深度エフェクト** を作って画像を合成し、ポストプロセスを施して現実味のある結果を作ります。

Rendering and Saving Images

(画像のレンダリングと保存)

レンダリングの設定を調整し終わったら、[画質](#) や [ファイル形式](#) に基づいて実際に画像をレンダリングする必要があるでしょう。静止画のレンダリングは非常に簡単です。[アニメーションのレンダリング](#) はもう少し複雑で、次の章で扱います。

アクティブカメラから画像をレンダリングするには、Render パネルで、大きな画像のあるボタンをクリックします。デフォルトでは 3Dビューが UV/Image エディタと置き換えられ、レンダリング結果が現れます。

Displaying Renders

(レンダリング結果の表示)

こうしてレンダリング結果が Image エディタに表示されます。Display ドロップダウンメニューでは、この表示方法を指定できます:

Keep UI

Image エディタに描画しますが、UI を切り替えません。レンダリング結果を見るには、自分で Image エディタを開く必要があります。

New Window

新たにフローティングウィンドウを開いてレンダリング結果を表示します

Image Editor

3Dビューは Image エディタに置き換えられ、レンダリング結果を表示します

Full Screen

Image エディタが UI を置き換え、レンダリング結果を表示します

どのオプションを選んでも、レンダリング表示は Esc を押すと閉じ、元のビューに戻ります。

Saving

(保存)

レンダリングした画像は Image メニューにある保存オプションを使って保存できます。


Animation Playback

(アニメーションの再生)

Render パネルにある 'Play' ボタンを押すと、レンダリングしたアニメーションを新たなウィンドウで再生します。

実行中のアニメーションプレイヤーに画像や動画をドラッグ&ドロップすることもできます。新たなデータで再生が再実行されます。

キーボードショートカット

- A フレームスキップの切り替え
 - P ピンポン再生 (再生と逆再生の繰り返し) の切り替え
 - ↵ Enter 再生開始 (一時停止時)
 - 0 NumPad ループ再生の切り替え
 - Padperiod 手動でフレームを進めます
 - ← ひとつ後ろのフレームに戻ります
 - → ひとつ前のフレームに進みます
 - ↓ 10フレーム戻ります
 - ↑ 10フレーム進みます
 - ⇧ Shift↓ 逆向きに再生
 - ⇧ Shift↑ 通常方向に再生
 - ⇧ Shift 押し続ける間フレーム番号を表示
-
- LMB  ドラッグして再生箇所を選びます
-
- CtrlPlus ズームイン
 - CtrlMinus ズームアウト
 - Esc 終了
-
- 1 NumPad 60 fps
 - 2 NumPad 50 fps
 - 3 NumPad 30 fps
 - 4 NumPad 25 fps
 - ⇧ Shift4 NumPad 24 fps
 - 5 NumPad 20 fps
 - 6 NumPad 15 fps
 - 7 NumPad 12 fps
 - 8 NumPad 10 fps
 - 9 NumPad 6 fps
 - / NumPad 5 fps
 - Minus 再生速度を遅く
 - Plus 再生速度を速く

Display Options

(表示オプション)

レンダリングした画像が Image エディタに表示されると、新たなメニュー項目が使えるようになります。

Slot Menu(スロット)

レンダリング前に新しいスロットを選ぶと、次のレンダリング結果を一時領域に保存できます。スロットにレンダリングされた画像は、そのスロットを選ぶと見ることができます。空のスロットを選ぶと Image エディタに空のグリッドが表示されます。J を使うと保存したレンダリングを巡回できます。AltJ で逆方向に巡回できます。

Render Layer

[レンダーレイヤー \(eng\)](#) を使っていれば、表示するレイヤーをこのメニューで選びます。

Render Pass(パス)

[レンダーパス \(eng\)](#) を使っていれば、表示するパスをこのメニューで選びます。

Image Painting

このアイコンは画像ペイントの有効/無効を切り替えます

表示モード

最後の4つのボタンは画像の表示方法を指定します

RGB

画像をアルファチャンネルを使わずに表示します

RGBA

透明なピクセルを、アルファチャンネルを表すチェッカー模様で置き換えます

Alpha Channel

グレースケール画像を表示します。白い領域は不透明になり、黒い領域はアルファ値 0 を持ちます

Z Depth

[カメラ設定](#) で指定したとおりに、Clip Start から Clip End の間でカメラからの奥行きを表示します

Curves Panel

Curves パネルは Properties パネルから利用できます。画像の色調整に使えます。

Render Quality

(レンダリング品質)

レンダリングした画像の品質には多くの要素が関わっています。レンダー設定を一切変えずにシーンをレンダリングした絵には、きっと満足できないでしょう。以前の章ではモデリング、陰影処理、テクスチャ、そしてシーンに照明を当てる方法を学びました。こうした分野に関する設定の最適化は高品質な画像を作る助けになりますが、レンダリングのボタンを押す前に重要になり始める設定がいくつかあります。これらはレンダリング画像の見た目に直接影響します。

次の章で扱うレンダーレイヤーとレンダーパスはどちらも、シーンの一部要素から画像を合成するものです。エフェクトを「あとから」レンダリングするのではなく、レンダーから直接作ることが必要な場合もあります。

[Color Management and Exposure](#)

(色調補正と露出)

3D レンダリングの重要な側面として見落とされがちなのが色調補正です。一般にリニアレンダリングとして知られる色調補正を使わない場合、レンダリングエンジンはシーンの照明効果を正しく解釈するものの、モニタには不正確に表示されます。Blenderはこの手順を単純化しますが、レンダリング画像の色空間が処理経路にどのように関係するか、知っておくことは重要です。

[Anti-Aliasing](#)

(アンチエイリアス)

アンチエイリアスは、対照的な色の並ぶ場所に現れる、境界線のぎざつきを取り除きます。これはレンダリング品質の面で非常に重要です。この設定をしないと、ふつうは絵がいかにも CG なアマチュア作品のように見えるでしょう。

[Exposure \(Lighting\)](#)

(露出(明暗))

露出は科学用語で、カメラのフィルムやセンサーが光にさらされる時間を表します。露出が長いほど画像は明るくなります。CG では記録されている光の値を増減させて、長い短い露出をシミュレートします。照明の設定か、またはより優れた [色調補正の設定](#) で調整します。

[Depth of Field](#)

(被写界深度/DOF)

現実世界のカメラは、決まった焦点距離を持っています。これは焦点にあるものとレンズ間の距離です。この区間(被写界深度)から外れた位置にあるほど、対象にピントがあわなくなります。デフォルトではすべてのオブジェクトに完全に焦点が合った状態でレンダリングされます。被写界深度の使い方によって、距離感を不自然にしたり、自然にしたりできます。

[Motion Blur](#)

(モーションブラー)

カメラには決まったシャッター速度やフィルムの露出時間があります。撮影時に動いていたものは、ある程度ふれた状態で写るでしょう。高速移動しているものほど、低速なものよりふれて写るはずですが。カメラの構造上生まれた絵の歪みですが、人間はこれが見えることを期待しているので、なくしてしまうと絵が嘘っぽくなるかもしれません。CG では重要なエフェクトです。

Anti-Aliasing

(アンチエイリアス)

コンピュータの生成する画像はピクセルできていて、当然ながら各ピクセルには一つの色しか割り当てられません。レンダリング工程においてレンダリングエンジンは、各ピクセルで見えているオブジェクトに基づいて、ピクセルに一つ色を割り当てる必要があります。これはよく特に鋭さのある境界線や細い線のある場所できれいな結果を生まず、とりわけ斜めの線でははっきり目立ちます。

エイリアシングとして知られるこの問題の打開に、アンチエイリアシング技術を使えます。基本的に各ピクセルは 5 ピクセル以上の大きさがあるものとしてレンダリングされ(オーバーサンプリング)、平均をとった色がレンダリングされるピクセルに割り当てられます。

アンチエイリアシング、またはオーバーサンプリング(OSA)を制御するボタンは Renderコンテキストの Anti-Aliasing パネルにあります。

Options

(オプション)

Anti-Aliasing チェックボックス
オーバーサンプリングを有効にします

5 / 8 / 11 / 16

使用するサンプルの数です。数値 5, 8, 11, 16 は特定の抽出パターンのプリセット値です。値が高いほど境界線はきれいになりますが、レンダリングは遅くなります。

デフォルトでは、Blender は固定された "Distributed Jitter" テーブルを使います。一つのピクセル内のサンプルは、次の二つの性質を保証する方法で分配され、小刻みに動かされます:

1. どのサンプルも隣接サンプルと等間隔
2. サンプルはすべてのサブピクセル位置を縦横両方向に等しく覆う

下の図はサンプル数が 5, 8, 11, 16 のときの Blender の抽出パターンです。複数ピクセルにわたって等しく分散していることを示すため、隣接ピクセルのパターンも描かれています。各ピクセルは同一のパターンを持っていることに注意してください。

5 samples

8 samples

11 samples

16 samples

Full Sample(フルサンプル)

アンチエイリアスの抽出のたびに、Render Layer の結果全体を保存します。Compositing でのアンチエイリアスの問題を解消します。

Filtering

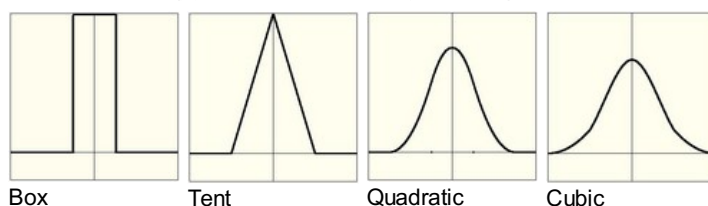
(フィルター)

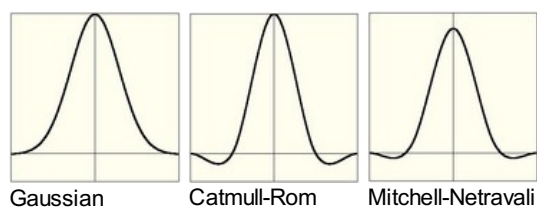
サンプルがレンダリングされると、サンプルごとに色とアルファの情報が入手できます。これは各サンプルのピクセルへの貢献度を定めるため重要です。

もっとも単純な方式では全サンプルの平均をとり、ピクセルの色にします。これを "Box Filter" と呼んでいます。この方式の欠点はピクセルの縁に非常に近い位置にあるサンプルを考慮しないことで、この結果隣接ピクセルの色にも影響することがあります。

フィルターメニュー: サンプルの平均をとるために使うフィルター種類を決めます: |Box|Blender でもともと使われていたフィルターで、比較的低品質です。Box Filter はピクセル自身の範囲内にあるサンプルだけをピクセル色に加算します。他のフィルターの数式は、サンプルの色の一定量を他のピクセルにも確実に分配します。

Box	低品質の box shaped 曲線(上記参照)
Tent	過度に単純化したフィルターで鮮明な結果になります
Quadratic	2次曲線
Cubic	3次曲線
Gauss	ガウス分布。もっともぼやけます
Catmull-Rom	Catmull-Rom フィルターはもっとも鮮明になります
Mitchell-Netravali	Mitchell-Netravali は優れた万能型フィルターで、ほどよい鮮明さを与えます





Gaussian

Catmull-Rom

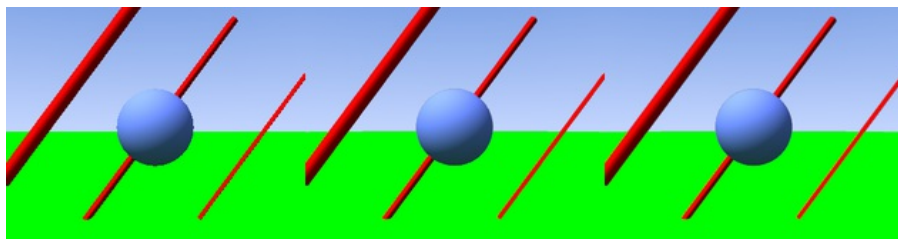
Mitchell-Netravali

Filter Size

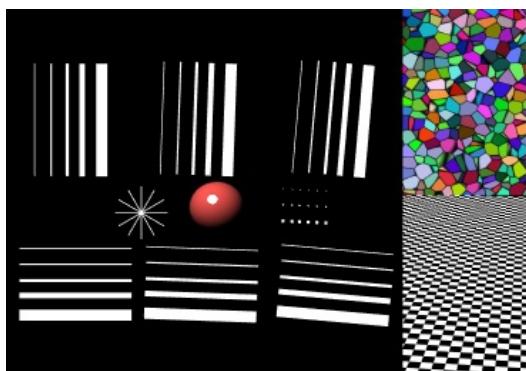
(フィルターサイズ)

フィルターサイズ値を小さくするほどサンプルは中央に寄り、画像はぼやけます。フィルターサイズを大きくすると鮮明な結果になります。最後の二つのフィルターには負の部分があり、鮮明さが余分に増えることに注意してください。

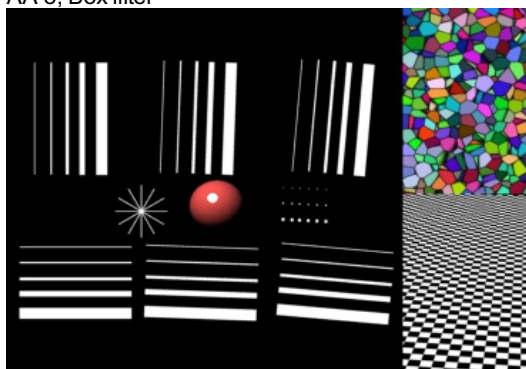
Examples



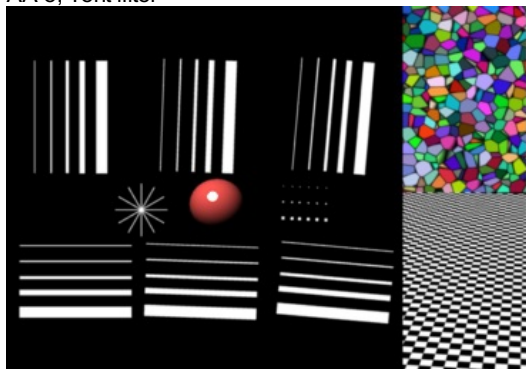
AAなしのレンダリング(左)AA=5(中央)AA=8(右)



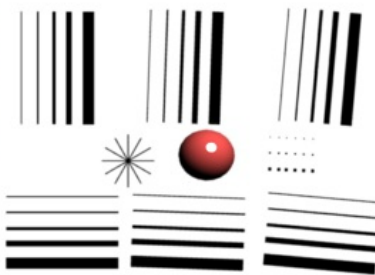
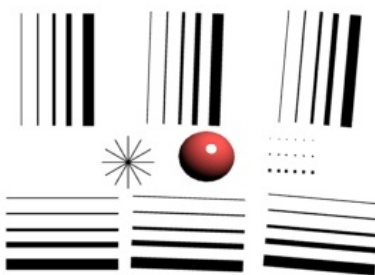
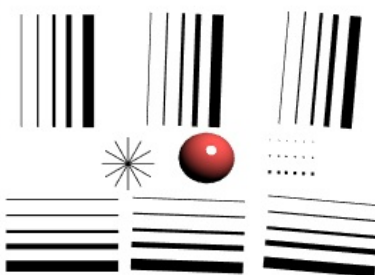
AA 8, Box filter

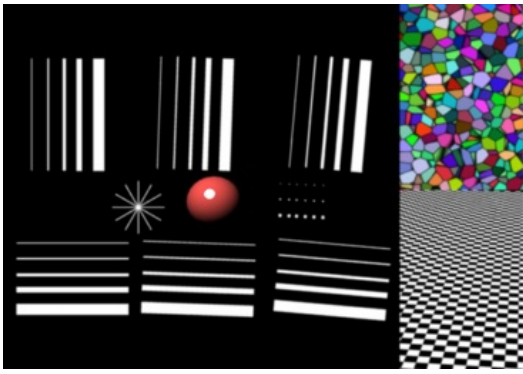


AA 8, Tent filter

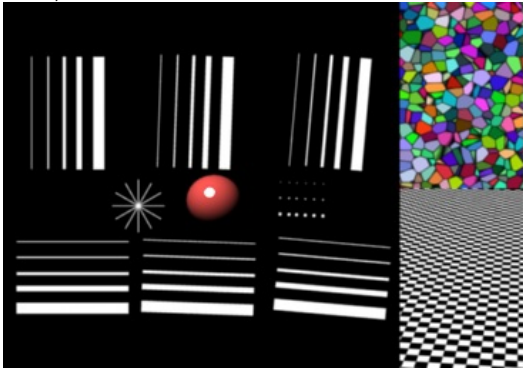
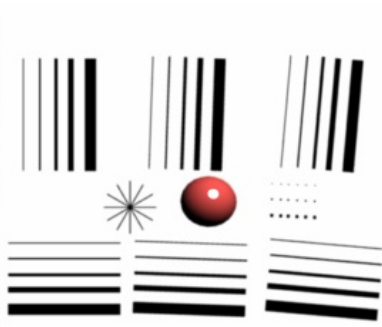


AA 8, Quadratic filter

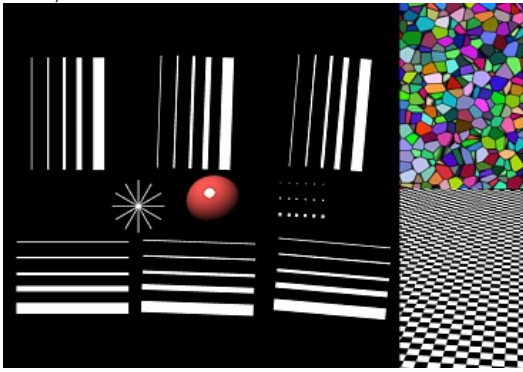
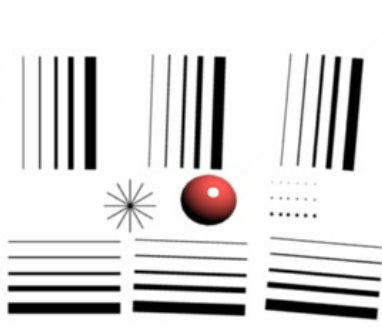




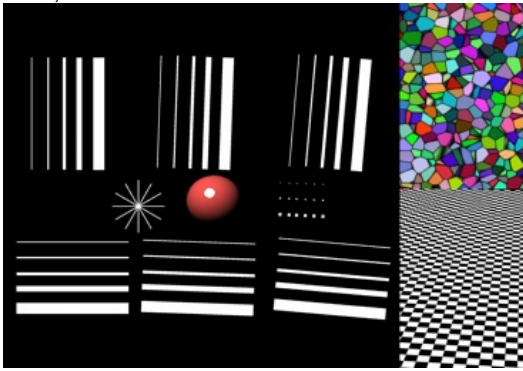
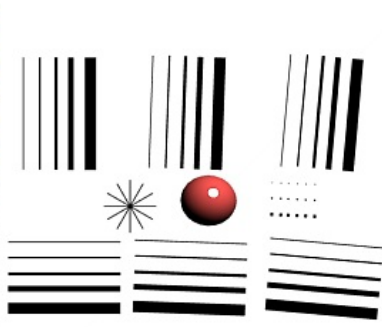
AA 8, Cubic filter



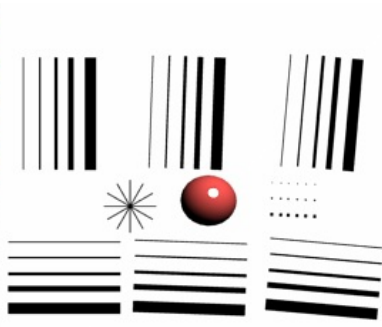
AA 8, Gaussian filter



AA 8, Catmull-Rom filter



AA 8, Mitchell-Netravali filter



アニメーションのレンダリング

静止画をレンダリングすると、できあがった後に、レンダリングバッファから画像を見たり保存したりできます。一方、アニメーションは一連の画像、すなわちフレームからなり、レンダリングされた後に自動的に、ディスクに直接保存されます。

フレームのレンダリング後、クリップの編集をする必要があるかもしれません。もしくは先にコンポジター (compositor) を使ってグリーンスクリーンマスク、マッピング、色補正、被写界深度 (DOF) などを画像に施します。この結果はシーケンサー (sequencer) に流し込まれ、そこでストリップ (strip) を切ったり混ぜたりして、最終的な重ね合わせができあがります。

最後に、シーケンサーからレンダリングし、フレームを圧縮して再生可能なムービークリップを作ることができます。

作業手順

たいいの場合、アニメーション内のさまざまなフレームの発生時期やライティング、配置、材質などをチェックするために、中間的なレンダリングを何度も行います。ある時点で、公表用の完成したアニメーションの、最終的なレンダリングをする準備ができます。

ムービーやアニメーションの制作には、二つの取り組み方があります。とるべき手法はムービーのレンダリングに必要な CPU 時間によって変わります ([レンダリングのパフォーマンス \(eng\)](#) をご覧ください)。「代表の」フレームを希望の解像度でレンダリングして時間を測り、最終的にムービーに使われるフレーム数をかけて、合計レンダリング時間を推測します。

合計レンダリング時間が1時間以上なら、各フレームごとに静止画を出力する **フレーム分割 (Frame Sequence)** 手法をとりたくなるでしょう。たとえば、映画用の1分のビデオクリップをレンダリングする場合、フレーム数は (60秒/分) × (24フレーム/秒 (または1440フレーム/分)) になります。各フレームのレンダリングに30秒かかるなら、1分あたり2フレームをレンダリングでき、720分 (12時間) のレンダリング時間が必要です。

レンダリングは可能な限りすべての CPU 時間を使います。夜は使われていないコンピュータを使って夜通しレンダリングすべきです。またはレンダリングの間 Blender の優先度を低く設定し、他の作業をします。RAMの空きに注意してください！

直接法 (Direct Approach) は (強く推奨されるものではなく、普通の方法ではありませんが) 出力形式を AVI か MOV にして、アニメーションを直接ムービーファイルにレンダリングする方法です。Blender はアニメーションの全フレームを持ったひとつのファイルを作ります。次に、Blender の VSE を使ってこのアニメーションに音声トラックを追加し、MPEG形式でレンダリングしてムービーを完成させます。

フレーム分割法 (Frame Sequence Approach) はもっと安定した手法です。出力形式を静止画像のものにして (例: JPG、PNG、MultiLayer)、アニメーションのレンダリングボタンをクリックします。出力は一連の画像になり、各画像が順にフレームになります。

Blender はアニメーションの各フレームに対してファイルを作ります。その後、Blender のコンポジターを使って任意のフレーム操作 (ポストプロセス) を行えます。さらに最終的な連続画像を Blender の VSE で読み込んで、音声トラックをアニメーションに追加し、MPEG形式でレンダリングして書き出してムービーを完成させることができます。フレーム分割法のほうが少し難しく、ディスク容量が多く必要ですが、柔軟性があります。

以下は、手法選択の助けになるガイドラインです。

直接法を選ぶ条件

- レンダリング時間の合計が1時間以下の、短いもの
- 安定した電源供給
- コンピュータで他のことをしない

フレーム分割法を選ぶ条件

- レンダリング時間の合計が1時間以上
- ポストプロダクションの作業が必要
 - 色調／ライティングの調整
 - グリーンスクリーン／マット置換
 - 階層化 (Layering) ／コンポジティング
 - 最終的に複数の形式やサイズの製品が必要
- 圧縮／コーデックに、中間フレームと調整が必要
- 一部で正確なタイミングが必要 (音声トラックへのリップシンク等)
- コンピュータを使用する際にレンダリングを中断し、中断箇所からの再開ができてほしい

フレーム分割法 (Frame Sequence) の作業手順

1. はじめにアニメーションを用意します。
2. 寸法 (Dimensions) パネルでレンダリングのサイズ、アスペクト比、使用するフレームの範囲を、フレームレートと同じように (設定済みのはずです) 選びます。
3. 出力 (Output) パネルで、画像としてアニメーションをレンダリングするように準備します。普通は劣化しない形式を使います (私は PNG や Multilayer がロスレスなので好きです)。
4. 同じく出力パネルで、出力先パスとファイル形式を選びます。例えば "/render/my-anim-" です。
5. アニメーションのフレーム範囲の、開始と終了を確認します。
6. .blend ファイルを保存します。
7. アニメーション (Animation) ボタンを押します。コンピュータが全フレームのレンダリングを終えるまで、時間のかかる作業 (睡眠、ゲーム、私道の掃除等) をして待ってください。
8. アニメーションがいったん完成すれば、OS のファイルエクスプローラーで出力先フォルダを開きます。この例では .render です。そこには 0000 から最大で 9999 までの連番の振られた画像 (.png や .exr など、レンダリングの形式として選んだもの) がたくさんあるでしょう。これら一枚一枚がフレームです。
9. ここで、Blender の [ビデオシーケンスエディタ \(video sequence editor\)](#) を開きます。
10. 追加 (Add) メニューの [画像を追加 \(Add Image\)](#) を選びます。出力先フォルダにある、アニメーションに加えたいフレームをすべて選択します (全選択は A を押すと簡単です)。シーケンスエディタにはストリップとして追加されます。
11. これで、ストリップを編集してエフェクトを加えることができます。または単にそのままにしておくこともできます。音声ストリップのような他のストリップを

加えることができます。

12. アニメーション全体をスクラップして、全フレームが含まれていることを確認します。
13. ポストプロセッシング(Post Processing)パネルにある Scene Render ボタンのうち、シーケンサー(Sequencer)を有効にします。
14. Format パネルで希望のコンテナとコーデック(MPEG H.264等)を選び、設定します。ビデオコーデックについては次のページをご覧ください: [出力のオプション \(eng\)](#)
15. アニメーションのレンダリングボタンをクリックすると、Blender がシーケンスエディタの出力をムービーとしてレンダリングします。

この手間をかけるのはなぜでしょうか？ まず第一に、一枚ずつフレームをレンダリングすれば、いつでもレンダリングウィンドウで Esc を押してレンダリングを止めることができます。すでにレンダリングしたフレームは、独立したファイルとして書き出されているため、なくなることはありません。中断した箇所に範囲をあわせて、いつでも続きを行えます。

あとからフレームを編集し、ポストプロセスを施すことができます。シーケンスエディタでいかしたエフェクトを加えることができます。ほとんど苦勞せずと同じシーケンスを別の解像度でレンダリングしたり(640x480、320x240 他)、別のコーデックを使ったりできますが(違うファイルサイズや品質を得られません)。

オプション

出力(Output)パネル

デフォルトでは、アニメーションは 出力(Output) パネルで指定されたディレクトリ(アニメーションの場所と拡張子)にレンダリングされます。AVI形式が選択されると、名前は #####_#####.avi になります。ここで、##### はアニメーションの開始および終了フレームを示す、必要ならゼロで0詰めされた、4桁の整数です。

一方、画像形式が選択されると、#####という名前の一連の画像(##### はフレーム番号)がディレクトリに作られます。

ファイル拡張子(File Extensions)

ファイルタイプごとに正しい拡張子を出カファイルに付加します

上書き(Overwrite)

レンダリング時に既存のファイルを上書きします

プレースホルダー(Placeholders)

レンダリング中、プレースホルダーである空のフレームを作ります

ポストプロセッシング(Post Processing)パネル

シーケンサー(Sequencer)

3D シーンのアクティブカメラのビューの代わりに、シーケンスエディタの出力をレンダリングします。シーケンスがシーンストリップを含む場合は、これらもパイプラインの一部としてレンダリングされます。「Do Composite」も有効にされていると、シーンストリップはコンポジターの出力になります。

コンポジティング(Compositing)

コンポジティングノードの出力をレンダリングしてから、すべての画像をコンポジットノードマップに通し、Composite Output ノードに流し込まれた画像を表示します。

ヒント

うへえ！ がさつな妹が PC の電源を切りました、動画レンダリングの真っ最中なのに！

アニメーションがほんとうに単純なもので、30分以内にレンダリングできそうなら別ですが、そうでなければ、冒頭からムービーファイルとして作らず、ロスレスの形式(TGA、PNG、BMP)で切り分けた画像フレームとしてレンダリングするのは良い考えです。こうすると、レンダリングをやり直す必要のある問題が発生しても、すでにレンダリングしたフレームは出力ディレクトリに残るため、簡単に復旧できます。開始(Start)のフレーム番号を再レンダリングの開始番号に変えて、アニメーション(ANIM)をもう一度クリックするだけです。

中盤の数フレームだけを再レンダリングしたいのですが。

はじめはフレーム分割してレンダリングするのもよい案です。数フレームだけにミスがあった場合に、該当のフレームだけを修正して再レンダリングできるからです。その後、Blender のシーケンスエディタやコンポジティングノードを使って、切り分けたフレームからムービーを作れます。

最初のフレームだけがレンダリングされ、Blender が固まります。

アニメーションボタンをクリックして最初のフレームだけがレンダリングされるなら、出力ファイルがメディアプレイヤーによってロックされていないことを確認してください。通常はレンダリング時にコンソールを確認してください。

Quicktime ムービーが作れません。

CreateMovieFile error: -47

Quicktime ムービーストリップが(おそらくVSEが)使用中のため上書きできません。もし VSE で使用中なら、ストリップを削除するか、OSのファイルエクスプローラーを使ってそのファイルを削除してください。

Render Baking

(レンダリングの焼き込み)

Baking、バイクとは一般に、のちに行う他の処理を高速化するために、何かを事前計算することを指します。ゼロからレンダリングを行うと、設定によっては多くの時間がかかります。このため Blender では、選択したオブジェクトのレンダリングの一部を先に「焼いておく/バイクする」ことができます。こうすればレンダリングボタンを押したとき、このオブジェクトの色は再計算する必要がないため、シーン全体が高速にレンダリングされます。

レンダリング結果を焼き込むと、レンダリングされたメッシュオブジェクト表面の2次元のビットマップ画像ができます。オブジェクトの UV 座標をつかって、画像をオブジェクトに再マップすることができます。バイクは個々のメッシュが UV 展開されたときにはじめて完成します。準備や実行に時間がかかる一方で、レンダリング時間を節約します。長いアニメーションをレンダリングするつもりなら、バイクに費やす時間は、長いアニメーションの各フレームをレンダリングする時間に比べれば微々たるものです。

AO や範囲光源からの柔らかい影のような、光と影を強調するような作品で、バイクを使ってください。メインのオブジェクトに AO を焼きこめば、全体のレンダリングで AO を有効にする必要はなくなり、レンダリング時間を節約できます。

Full Render を使うか Textures を使って画像テクスチャを作ってください。バイクしたプロシージャルテクスチャは、さらなるテクスチャペイントの開始点にできます。Normals (法線) は低解像度のメッシュを高解像度のように見せるために使ってください。そのためには細部まで細かく作られた高解像度のモデルを UV 展開して、法線をバイクします。法線マップを保存して、同様に展開した低解像度のメッシュの UV に マッピング (テクスチャ設定) します。低解像度のメッシュは高解像度のメッシュのような見た目になりますが、少ない面/ポリゴンしか持ちません。

利点

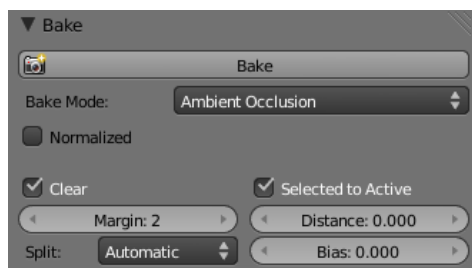
- レンダリング時間を大幅に減らせます
- テクスチャペイントが簡単にできるようになります
- ポリゴン数を減らします
- レンダリングの繰り返しが行き、その回数だけ節約時間が倍になります

欠点

- オブジェクトは UV 展開されている必要があります
- 影を焼きこむと、光源やオブジェクトは影との整合性をとるため動かせなくなります
- 大きなテクスチャ (例: 4096x4096) はメモリを多く使う可能性があり、レンダリングした場合と同じぐらい遅くなります
- UV 展開、焼き込み、ファイル保存、テクスチャの適切なチャンネルへの適用に人の作業と時間が必要です

Options

(オプション機能)



Ambient Occlusion

Bake Mode

Full Render

(すべてのレンダリング)

Specularity/スペキュラー と SSS を除く、すべてのマテリアル、テクスチャ、ライティングを焼き込みます。

Ambient Occlusion

(アンビエントオクルージョン)

AO を World パネルでの指定に従って焼き込みます。シーン内の光源はすべて無視されます。

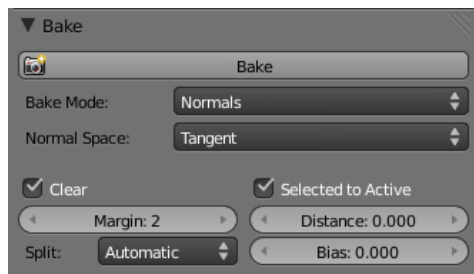
Normalized

マテリアルの設定を使わずに正規化します

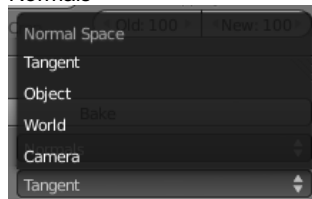
Shadow

(影)

影と光を焼き込みます。



Normals



Normal Space

Normals

(法線)

(特に) tangent/接線 やカメラ空間の法線を RGB 画像に焼き込みます。

Normal Space

法線を別の空間に焼きこむことができます:

Camera space (カメラ)

旧のデフォルトです。

World space (ワールド)

オブジェクトのトランスフォームや変形に左右される、World 座標の法線

Object space (オブジェクト)

オブジェクトのトランスフォームから独立しているが、オブジェクトの変形に左右される、オブジェクト座標の法線

Tangent space (タンジент)

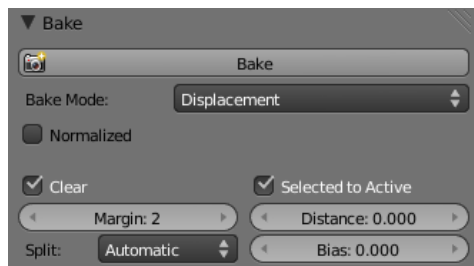
オブジェクトのトランスフォームや変形から独立した、接線空間座標での法線。デフォルトです。たいていの場合正しい選択肢です。その後法線マップはアニメーションのあるオブジェクトにも使えます。

マテリアルに対しても既存の画像テクスチャの Image Sampling (画像のサンプリング) → Normal Map (ノーマルマップ) 設定の隣にあるオプションで同じ空間を選ぶことができます。正しい結果を得るためには、この設定が焼き込みの設定と一致する必要があります。

Textures

(テクスチャ)

陰影を除く、マテリアルとテクスチャの色だけを焼き込みます。



Displacement

Displacement

(ディスプレイスメント)

法線マップの焼き込みに似ていますが、displacement (ディスプレイスメント) マップは Selected to Active オプションを使って、高解像度のオブジェクトを展開済みの低解像度オブジェクトに焼き込むのにも使えます。

Normalized

距離に正規化

これを subsurf や displacement モディファイアと組み合わせて使うときは、焼き込み前に「低解像度」モデルに重めの subsurf モディファイアを一時的に追加する必要があります。これは、subsurf の上に displacement モディファイアを使えば、subsurf されたジオメトリとの相対的な差として保持され

るため、元の基本メッシュよりも displacement が正しくなるということです (元のメッシュは subsurf でかなり歪められる可能性があります)。バイクする間の subsurf の render レベルを高くするほど displacement の正確性が増します。このテクニックは外部レンダラーで使うためのディスプレイメントマップを保存する際にも役立ちます。

Emission

(放射)

マテリアルの Emit や Glow の色を焼き込みます。

Alpha

(アルファ)

マテリアルのアルファ値、透明度を焼き込みます。

Mirror Colors and Intensity

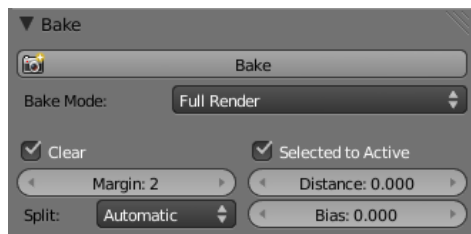
(鏡面色と強度)

Mirror Color と Intensity 値を焼き込みます。

Specular Color and Intensity

(反射光色と強度)

Specular Color と Intensity 値を焼き込みます。



Full Render

Additional Options

(その他のオプション)

Clear (クリア)

有効化すると焼き込み前に画像を消去し、選択した背景色にします (デフォルトは黒です)

Margin (余白)

焼き込んだ結果から UV "island" の境界線を越えてこのピクセル数だけ拡張し、テクスチャの seam をほぐします

Split (分離)

Fixed (定値)

四角面を頂点 (0,1,2) (0,2,3) でふたつの三角面に分けます

Fixed alternate (別の定値)

四角面を頂点 (1,2,3) (1,3,0) でふたつの三角面に分けます

Automatic (自動)

四角面を、バイク時に歪みが最小になるような三角面に分けます

Select to Active (選択→アクティブ)

アクティブオブジェクトに別のオブジェクトの情報を焼き込めるようにします

Distance (距離)

別のオブジェクトの頂点とアクティブオブジェクトの頂点を対応づける距離。Selected to Active でのみ使われます。

典型的な用途は、細部造形のあるハイポリオブジェクトを作ってその法線をローポリオブジェクトに焼き込むというものです。この結果できた法線マップをローポリオブジェクトに用いて、細部造形があるように見せることができます。

Bias (バイアス)

この値を増やすほど、別のオブジェクトの一番近い面ではなく、遠い側の面の頂点を焼き込みます (例: シャツの番号札の焼き込み)。値は Blender 単位です。

レンダリング時にメッシュが見えている必要性

通常のレンダリングでメッシュが見えていない場合、例えばアウトライナーでレンダリングを無効化していたり、DupliVerts 設定が有効化されていたりすると、そのメッシュに焼き込むことはできません。

Workflow

(作業の流れ)

1. 3D ビューウィンドウでメッシュを選択し、UV/面選択モードに入ります
2. [メッシュオブジェクトを UV 展開します](#)
3. UV/Image エディタウィンドウで、新規画像を作るか既存のものを開きます。3Dビューがテクスチャ表示モードになっていれば、これで画像がメッシュにマップされたのが見えるはずですが、すべての面が選択されていることを確認してください。
4. Render コンテキスト最下部にある Bake (バイク) パネルで、目的の種類 (Full Render など) をバイクします。
5. レンダリングが終わると、Blender は元の画像と焼き終えた画像を入れ替えます。
6. 画像を保存します。
7. 画像をメッシュに UV テクスチャとして適用します。「displacement」や「normal maps」用なら [バンプおよび法線マップ](#) を参照してください。「full」や「texture」用なら [テクスチャ](#) を参照してください。
8. 画像を精錬したり、[テクスチャペイント](#) や外部の画像エディタを使って美しくします。

はじめに

レンダリング速度を上げたい状況では、blender にリモートにアクセスしてレンダリングするか、blender をコマンドラインで使うスクリプトを作りましょう。

コマンドラインを使う利点は、X server が不要であるため(Linux の場合)、SSH や telnet でリモートからレンダリングできることです。

引数に関する注意点

空白で区切る必要があります。

```
blender -ba test.blend # 'a'が無視されます
blender -b test.blend -f8 # '8'はフレーム数として扱われず無視されます
```

渡した順に実行されます。必ず **-f** か **-a** を最後に渡してください。

```
blender -b test.blend -f 1 -o /tmp # 出力パスを決める前にレンダリングされ、出力されません
blender -b -o /tmp test.blend -f 1 # blendファイルのロード時に出力先が上書きされ、/tmpに保存されません
blender -b test.blend -o /tmp -f 1 # うまく動作します
```

構文

```
blender [-b <dir><file> [-o <dir><file>] [-F <format>]
[-x [0|1]] [-t <threads>] [-S <name>] [-f <frame>]
[-s <frame> -e <frame> -a]] [[-P <scriptname> [-- <parameter>]]]
```

レンダリングオプション:

```
-b または --background <file>
<file> をバックグラウンドで読み込み (UI非表示のレンダリングによく使われます)

-a または --render-anim
開始フレームから終了フレームまでをレンダリング (両フレームを含む)

-S または --scene <name>
アクティブシーン <name> をレンダリング用に設定

-f または --render-frame <frame>
<frame> をレンダリングして保存
+<frame> 開始フレームの相対指定、-<frame> 終了フレームの相対指定

-s または --frame-start <frame>
開始フレームを <frame> に設定 (-a より前で使ってください)

-e または --frame-end <frame>
終了フレームを <frame> に設定 (-a より前で使ってください)

-j または --frame-jump <number>
フレームのレンダリングごとに進めるフレーム数を設定

-o または --render-output <path>
レンダリングの出力パスとファイル名をセットします
出力パスに blend ファイルからの相対指定をすときは // を使います。
ファイル名に # を使うと frame 番号で置き換わります。番号はゼロ埋めされます。
例: ani_###_test.png → ani_01_test.png
例: test-#####.png → test-000001.png
ファイル名に # を使わない場合、自動的に ##### が末尾に追加されます。
例: blender -b foobar.blend -o //render_# -F PNG -x 1 -a
→ //render_# は //render_##### になり、フレームが //render_0001.pngとして書き出されます

-E または --engine <engine>
レンダリングエンジンを指定します。利用可能なエンジンの一覧は -E help で見られます。

-t または --threads <threads>
レンダリングに使うスレッドの量を指定します [1-64]
0を指定するとシステムのプロセッサの数が使われます
```

ファイル形式のオプション:

```
-F <format>
出力形式をセットします。有効な選択肢は..
TGA IRIS JPEG MOVIE IRIZ RAWTGA
AVIRAW AVIJPEG PNG BMP FRAMESERVER
(コンパイルして blender に含まれる形式で、システムによっては使えないもの)
HDR TIFF EXR MULTILAYER MPEG AVICODEC QUICKTIME CINEON DPX DDS

-x または --use-extension [0|1]
ファイル名のうしろに拡張子を付加するオプションです。
0 は付加せず 1 は付加します。
```

アニメーション再生のオプション:

(play button を押した時使われます。blender は動画プレイヤーのように動作します)

```
-a <options> <file(s)>
<file(s)> を再生します。-b が使われていなければこの方法で実行します。
-p <sx><sy> 左下の隅を <sx>,<sy> に合わせて開きます
-m ディスクから読みます (バッファを使いません)
```

```
-f <fps> <fps-base>      開始時の FPS
-j <frame>                フレームステップ
-s <frame>                開始フレーム
-e <frame>                終了フレーム
```

ウィンドウのオプション:

```
-w または --window-border
   ボーダーつきで開きます(デフォルト)
-W または --window-borderless
   ボーダーなしで開きます
-p または --window-geometry <sx> <sy> <w> <h>
   左下隅を <sx>, <sy> に合わせ、幅を <w>、高さを <h> にして開きます
-con または --start-console
   コンソールウィンドウの開いた状態で開始します(-b がセットされていると無視されます)(Windowsのみ)
--no-native-pixels
   高解像度ディスプレイ(MacBook 'Retina')で、native のピクセルサイズをしません
```

Game Engine 専用のオプション:

```
-g fixedtime      frame落ちせずに 50 hertz で実行します
-g vertexarrays  レンダリングに Vertex Arrays を使います(通常は高速)
-g nomipmap      Texture Mipmapping を使いません
-g linearmipmap  Nearest ではなく Linear Texture Mipmappingを使います(デフォルト)
```

Python のオプション:

```
-y または --enable-autoexec
   Python スクリプトの自動起動を有効にします
-Y または --disable-autoexec
   Python スクリプトの自動起動を無効にします(pydriver と スタートアップスクリプト)(デフォルトでは非標準としてコンパイルされます)
-P または --python <filename>
   指定した Python スクリプトファイルを実行します

--python-text <name>
   指定した Python script text block を実行します
--python-console
   インタラクティブなコンソールを使って Blender を実行します
--addons
   カンマ区切り(空白なし)のアドオンの一覧
```

デバッグのオプション:

```
-d または --debug
   デバッグをオンにします
   * すべての operator call と引数を表示
   * マウスのグラフ操作(デバッガーとのやりとり)を無効化
   * sys.stdin の設定を None にせず、維持します

--debug-value <value>
   開始時にデバッグ値 <value> を設定
--debug-events
   event system についてのデバッグメッセージを有効化
--debug-handlers
   event handling についてのデバッグメッセージを有効化
--debug-jobs
   バックグラウンドジョブの time profiling を有効化
--debug-python
   python についてのデバッグメッセージを有効化
--debug-wm
   window manager についてのデバッグメッセージを有効化
--debug-all
   すべてのデバッグメッセージを有効化(libmvを除く)
--debug-fpe
   浮動小数の例外を有効化
--disable-crash-handler
   crash handler を無効化
```

その他のオプション:

```
--factory-startup
   ホームディレクトリの "startup.blend" を読み込みません

--env-system-datafiles
   BLENDER_SYSTEM_DATAFILES 環境変数をセット
--env-system-scripts
   BLENDER_SYSTEM_SCRIPTS 環境変数をセット
--env-system-python
   BLENDER_SYSTEM_PYTHON 環境変数をセット

-nojoystick
   ジョイスティックのサポートを無効化
-noglsl
   GLSL shading を無効化
-noaudio
   サウンドシステムを None にします
-setaudio
   サウンドシステムを指定のデバイスにします
```

```
NULL SDL OPENAL JACK
```

```
-h or --help
```

```
このヘルプメッセージを表示して終了します
```

```
-v or --version
```

```
Blender のバージョンを表示して終了します
```

```
--
```

```
後続の引数を変更せずに渡して、オプションの実行を終了します。python の sys.argv からアクセスしてください
```

その他のオプション:

```
/?
```

```
このヘルプメッセージを表示して終了 (Windowsのみ)
```

```
--verbose <verbose>
```

```
ログ記録の詳細度を設定
```

```
-R
```

```
.blend 拡張子を登録して終了 (Windowsのみ)
```

```
-r
```

```
.blend 拡張子を登録して終了。メッセージを表示しません (Windowsのみ)
```

実行例

静止画のレンダリング

```
# blender -b file.blend -o //file -F JPEG -x 1 -f 1
```

- **-b**

インターフェイスなしで blender を起動

- **file.blend**

レンダリングする .blend ファイル

- **-o //file**

ディレクトリと保存する画像ファイルの名前

- **-F JPEG**

JPEG 形式

- **-x 1**

ファイル名に拡張子 .jpg を付加

- **-f 1**

フレーム1をレンダリング

動画のレンダリング

```
# blender -b file.blend -x 1 -o //file -F MOVIE -s 003 -e 005 -a
```

- **-b**

インターフェイスなしで blender を起動

- **file.blend**

レンダリングする .blend ファイル

- **-x**

ファイル名に拡張子 .avi を付加

- **-o //file**

ディレクトリと保存するファイルの名前

- **-F MOVIE**

低圧縮で .AVI 動画を保存

- **-s 003 -e 005 -a**

開始フレームを 003 終了フレームを 005 にセット。**重要: -s と -e は順番が逆だと動作しません!**

指定エンジンで Blender を起動する

-E エンジン名 または --engine エンジン名 を使います。

```
# blender --engine CYCLES
```

利用可能なエンジンは次のコマンドで一覧表示されます

```
# blender --engine help または blender -E help
```

出力の例

```
found bundled python: /home/satishg/bin/blender-2.65a-linux-glibc27-x86_64/2.65/python
Blender Engine Listing:
  BLENDER_RENDER
  BLENDER_GAME
  CYCLES[1]      Done
~/bin/blender-2.65a-linux-glibc27-x86_64/blender --engine help
```

プラットフォーム

実際に Blender をコマンドラインから起動する方法は、プラットフォームや Blender のインストールされた場所によって変わります。以下はさまざまなプラットフォームでの、基本的な使い方です。

Windows

コマンドプロンプトを開いて、Blender がインストールされたディレクトリに移動してから、blender コマンドを実行します。

```
# cd c:\<blender installation directory>
# blender
```

Mac OS X

ターミナルアプリケーションを開いて、Blender がインストールされたディレクトリに移動してから、アプリケーションバンドル内にある実行形式ファイルを実行します。次のようなコマンドを使います:

```
# cd /Applications/Blender
# ./blender.app/Contents/MacOS/blender
```

この操作をよく使うなら、単に「blender」と入力するだけでいよいよエイリアスを作ることができます。そのためにはターミナルで次のようなコマンドを(適切なパスを使って)実行します。

```
# echo "alias blender=/Applications/Blender/blender.app/Contents/MacOS/blender" >> ~/.profile
```

これで、新しいターミナルを開けば、次のコマンドが動作します:

```
# blender
```

Linux

ターミナルを開いて、Blender がインストールされたディレクトリに移動し、blender コマンドを実行します。

```
# cd <blender installation directory>
# ./blender
```

PATH 内に Blender をインストールしていれば (linux distribution package を通じてインストールすると通常はそうなります)、簡単になります:

```
# blender
```

Output Options

(出力のオプション)

レンダリング工程のはじめの一步は、出力のオプションを決め、セットすることです。オプションには描画サイズ、フレームレート、ピクセルのアスペクト比、出力する場所、ファイル形式があります。

Dimensions

(寸法)

Resolution(解像度)

レンダリング画像のサイズの設定です。

デフォルトでは SizeX と SizeY は 1920×1080 になり、X、Y の項目を調整して変更可能です。これで、画像全体のサイズを調節します。

Percentage (パーセンテージ) スライダーの値は、現在セットされている解像度を拡大縮小します。最終的な画像と同じ寸法比で、小さなテストレンダリングを行う際に便利です。

Aspect Ratio(アスペクト比)

すぐ下に二つ設定があります。AspX と AspY は、各座標軸に沿ってピクセルの形状を調節します。コンピューター画面のピクセルは正方形なので、デフォルトは 1:1 です。テレビ用短編の制作では、テレビのピクセルは正方形ではないため、このアスペクト比を対象のビデオ規格に合うように変える必要があるかもしれません: ヨーロッパ向けの PAL とアメリカ向けの NTSC があります

ピクセルのアスペクト比について詳しくは、[ビデオ出力](#) をご覧ください。

Border(ボーダー)

フレーム全体ではなく、ビューの一部だけをレンダリングすることができます。Camera View で Border を有効にして CtrlB を押し、矩形をドラッグしてレンダリングしたい範囲を決めます。CtrlAltB はボーダーを無効にするショートカットです。

なお、この機能は Performance にある Save Buffers オプション、および Anti-Aliasing にある Full Sample オプションを無効化します。

Crop(切り落とし/トリム) を有効にすると、レンダリングした画像で Border(ボーダー) サイズの周りに黒い領域を描かず、このサイズで切り取ります。

Frame Range(フレーム範囲)

[アニメーションのレンダリング](#) 用に、Start(開始) と End(終了) フレームをセットします。Step(ステップ) は、タイムライン上の各フレームごとに、前に進めるフレーム数を調整します。

Frame Rate(フレームレート)

[アニメーション](#) 用で、1秒間にいくつフレームが表示されるのかを示す値です。デフォルトはアニメーション向けの標準である 24 フレーム毎秒です。USA television 向けには 29.97 フレーム毎秒を使ってください。

Time Remapping(タイムリマップ)

アニメーションの長さを書き換えるのに使ってください。

Presets

(プリセット/既定セット)

楽をするために一番上のメニューには一般的なプリセットが用意されています(par = Pixel Aspect Ratio、ピクセルアスペクト比)。+と-のボタンで、独自の設定を追加したり削除したりできます:

DVCPRO HD 1080p	1280x1080, 3:2par 24fps
DVCPRO HD 720p	960x720 4:3par 24fps
HDTV 1080p	1920×1080 square pixels 24fps
HDTV 720p	1280x720 square pixels 24fps
HDV 1080p	1440x1080 4:3par 23.98fps
HDV NTSC 1080p	1440x1080 4:3par 29.97fps
HDV PAL 1080p	1440x1080 4:3par 25fps
TV NTSC 16:9	720x480 4:3.3par 29.97fps
NTSC 4:3	720×480 10:11par. 29.97fps
PAL 16:9	720x576 16:11par 25fps
PAL 4:3	720x576 12:11par 25fps

これは既定のセットにすぎません。お使いの PC のメモリが許す限り、希望の解像度をセットしてください。巨大なレンダリング出力を可能にするアイデアやテクニック、ツールについては Render のページをご覧ください。


Output Panel

(出力パネル)

アニメーション用にレンダリングしたフレームの場所や保存する画像の品質を設定します。

File Locations

(ファイルの場所)

デフォルトではアニメーションの各フレームは /tmp ディレクトリに保存されます。変更するには、他の入力欄と同様に、入力欄を ◊ Shift LMB  クリックして新しい場所を入力してください。// を使って新しい .blend ファイルを保存しなければ、// は Blender のインストールフォルダとして解釈されます。

入力欄の右にあるフォルダのアイコンをクリックすると、Blender ウィンドウ描画域がファイルブラウザウィンドウに変わります。このウィンドウはハードディスク内を見て回って、ファイルやディレクトリを選ぶのに非常に便利です。

パス指定

場所を示すパスの指定には、完全な *Microsoft-Windows* 上の *普通のドライブレター* か *マップしたドライブレター* (例: "F:")、*パンくず記法* (例: "./" や "../"), それに "//" (blend ファイルの場所) を使えます。スラッシュ (Unixスタイル) とバックスラッシュ (Windows スタイル) はどちらのプラットフォームでも使えます。省略すると、blender が起動した、現在の一時作業ディレクトリにファイルが保存されます。

File Type

(ファイル形式)

Blender は多くの画像形式をサポートしています。ファイル形式は一覧に、アルファベット順に並びます。

アニメーション CtrlF12 の出力形式は 出力 (Output) パネルの ファイル形式 メニューを使って選びます。ここでたくさんの画像や動画の形式を選べます。静止画をレンダリングするときは、レンダリング後画像を保存する際にファイル形式を選べます。

さまざまな用途を持つたくさんの画像形式があります。ファイル形式によって画像は *可逆 (loss-less)* または *不可逆 (lossy)* な形式で保存されます。不可逆な形式では画質の劣化が生じますが、バイト数は減り、ディスク領域を節約できます。可逆な形式はピクセル単位で正確に画像を維持します。ファイル形式は *静的な (static)* 画像とムービー *クリップ (clips)* に分類することができます。

どちらのカテゴリにおいても規格 (静止画の形式およびクリップのコーデック) があります。これらはプロプライエタリ (一社により開発・管理) である場合も、オープンな規格 (コミュニティやコンソーシアムによる管理) である場合もあります。オープンな規格は一般的に他の特定の企業より長生きし、使用料不要で、ビューワを通じて自由に入手できます。プロプライエタリな形式は特定のビデオカードでのみ動作するかもしれませんが、コーデックが無料であってもビューワが有料であるかもしれません。

Compression

(圧縮)

形式によっては画像を圧縮してディスク使用量を減らせるものがあります。圧縮には可逆 (PNG, ...) と不可逆 (Jpeg, ...) のものがあります。不可逆な形式は個々のピクセル情報を保存しないので、画質は落ちます。他のすべての形式もだいたい同じですが、それぞれ利点と欠点があります。ファイル形式セレクトの下にあるボタンや入力欄から、圧縮を選択します。たとえば Jpeg を選ぶと、圧縮率 (品質:90 がデフォルト) を指定できます。品質を上げるほどディスク容量を使いますが、圧縮変換による画像の歪み (artifacts) は減り、絵の見た目はよくなります。

デフォルトの画像形式は Targa ですが、画像はバッファに蓄えられてから保存されるため、レンダリング後、このメニューを使って保存する前に画像ファイル形式を変えることができます (ご注意: これは静止画の話です。アニメーションのレンダリングには当てはまりません!)。

Channels

(チャンネル)

Blender はデフォルトでカラー (RGB) の画像をレンダリングしますが、白黒 (BW) やアルファチャンネルを伴うカラー (RGBA) も利用できます。ご注意: Output (出力) パネルの拡張子ボタンをセットしない限り、ファイル名に自動的に拡張子が追加されることはありません。したがってファイル保存ウィンドウには、.tga や .png といった何らかの拡張子が明示的に記入されている必要があります。

OpenEXR および **OpenEXR Multilayer** 形式だけは、Z-深度バッファの情報を格納します。**OpenEXR Multilayer** はレンダー階層とレンダーパスをポストプロダクションで合成可能なレイヤーとして格納する、唯一の形式です。

Image Formats

(画像の形式)

BMP	Bit-Mapped Paint、可逆形式、初期のペイントプログラムが使用
Iris	標準的な Silicon Graphics Inc (SGI) の形式。この立派な Unix OS マシンで使われます Portable Network Graphics は、可逆形式でフルカラー画像をサポートすることから、古い GIF に置き換わる規格です。アルファチャンネルをサポートします。
PNG	アルファチャンネルを保存するには RGBA ボタンを有効にしてください。
Jpeg	この仕様を定義したコンソーシアムの名前に由来する Jpeg はオープンな形式で、最小限の劣化で非常によく圧縮率を持っています。RGB 値だけを保存します。画像を再保存すればするほど圧縮され、劣化します。
Jpeg 2000	Jpeg 2000 コーデックを使います。
TARGA and Targa raw	Truevision Advanced Raster Graphics Adapter は簡素なラスターグラフィックス形式で、1984年に作られ、初期の IBM PC で使われました。アルファチャンネルをサポートします。 アルファチャンネルを保存するには RGBA ボタンを有効にしてください。
Cineon	Kodak Cineon camera で作られる形式で、最高位のグラフィックスソフトウェアで使われ、デジタルフィルム向けです。
MOV	Digital Moving-Picture eXchange 形式。Cineon に近いオープンで専門的な形式で、画像のメタ情報を含んでいます。16ビットの圧縮

OpenEXR	<p>縮されないビットマップであり、ファイルサイズは巨大になります。保管用に使われます。</p> <p>OpenEXR 形式で、ひとつのファイルに複数の画像レイヤーを保存できます。各レイヤーはひとつのレンダリングパスを保存します。たとえば shadow(影)、specularity(光沢)、color(色)などです。コーデックセレクタを使うと、MultiLayer ファイルを保存するのに使うエンコーディングを指定できます。デフォルトは ZIP(可逆)です。</p> <p>オープンで非プロプライエタリに拡張された HDR 画像形式です。アルファとZ-深度バッファの情報を保存します。</p> <ul style="list-style-type: none"> 16ビットの形式を使う場合は <i>Half</i> ボタンを有効にしてください。それ以外では 32ビットの浮動小数による高精度のカラー深度が使われます。 Zバッファ(カメラからの距離)の情報を保存する場合は <i>Zbuf</i> ボタンを有効にします。 ディスク容量を節約するには圧縮/復元 <i>CODEC</i> (コーデック) (デフォルトは ZIP)を選びます。 アルファチャンネルを保存するには <i>RGBA</i> ボタンを有効にします。 OpenEXR はとても新しい形式なのでふつうは OS がプレビューをサポートしていません。Previewを有効にすると EXR画像とともに JPG 画像を保存し、もとの画像がどのようなものか素早く簡単に見ることができます。
Radiance HDR	High Dynamic Range 画像形式は、チャンネルごとに 32ビットの浮動小数を使って、(1.0 より明るい光を持つ)画像を保存できます。
TIFF	テレタイプやファクシミリ画像によく使われます。
Frame Server	代替的な出力方式で、Blender が ネットワーク越しにフレームを送ります。フレームがディスク上には非圧縮でおさまりそうになく、外部のビデオエンコーダーを使う場合に便利です。 ドキュメント (eng)

VSE Rendering

(VSE レンダリング)

Rendering to an Image Sequence

(連続画像へのレンダリング)

コーデックでエンコードされたビデオストリップを切り離して再編集すると、いろいろすることが多くなるでしょう。各イメージを再構築するのに使われる内部的なエンコードのアルゴリズムは、1~3フレームごとにフレームを「取り去る」からです。直接 '生の' フレームセットで作業するには、非常に一般的なテックニックとして、ビデオをストリップとしてインポートし、フレームを順に、個別にレンダリングして出力する方法があります。各フレームは独立した画像ファイルとして保存されます (JPG がもっとも一般的です)。

このためには、元のビデオを Add->Movie で読み込みます。ファイル形式の SizeX と SizeY をセットし (元と同じにしても、歪めたり拡大縮小するために変えても構いません)、画像タイプを JPEG にして、品質 (Quality) 設定を調節し、アニメーションパネルで終了フレームをビデオクリップ内にある実フレーム数にセットします。アニメーションをクリックすると、出力パネルの上部で指定した場所に、一連の番号を振られたファイルが出力されます。

これで、ビデオストリップを削除して、代わりに Add->Image を行えます。ディレクトリにある画像すべてを順番通りに取り込むために、ディレクトリ名を右クリックし、画像のあるディレクトリを指定します。こうして、たとえばフレーム 4321 でカットしたとき、2番目のストリップの次のフレームは *また* フレーム 4322 から始まります。

Rendering to Video

(ビデオへのレンダリング)

(ボタンがどこにあるか覚えれば)ばかばかしいほど簡単です:

1. 一連の画像を上述のように追加します。
2. レンダーボタン群の Output(出力)ボックスの上のほうにある、出力ファイルのパスをムービーファイルの保存場所に変えます (例: C:\My Documents\MyMovie)。
3. ファイル形式をムービーファイルの形式 (AVI, MOV, FFMPEG) とコーデックにします。
4. シーケンスの再生時に使われるフレームレートをセットします。アニメーション/再生ボタンの下 (Dimensions(寸法)内)にあります。
5. アニメーションシーケンスにある画像の数を、End(終了)フレームに設定します
6. アニメーションをクリック

単体のムービーファイルが作られ保存されます。ファイル名は、指定したものに開始フレームと終了フレームの番号が足されたものになります (例: MyMovie0000-0250.avi)

Preparing your work for video

(ビデオ用作品の準備)

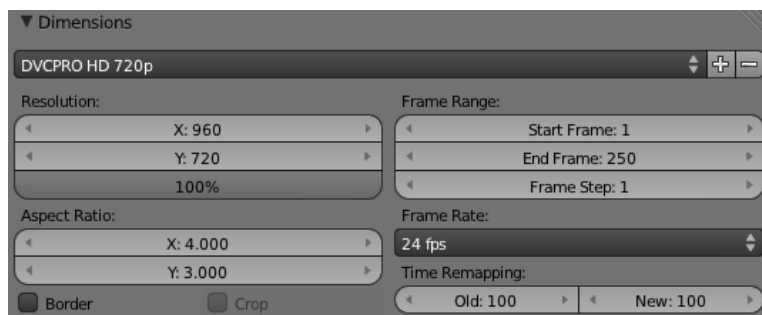
いったんアニメーションの方法を習得すればきっと、お好きなコーデックでエンコードされた素敵なアニメーションを作り始めたいでしょう。また、コミュニティの人たちとインターネットで共有したくなるかもしれません。

遅かれ早かれ、テレビ用、あるいはおそらくDVDに書き出すためのアニメーションを作りたくなるでしょう。以下は特にビデオの準備作業を対象にした、挫折しないための秘訣です。最初の秘訣は、カメラビュー内の白い点線を覚えておくことです！

PC 用にレンダリングすると、*外側*の点線の矩形より内側に描かれたイメージ全体が現れます。テレビ用なら、CRTの電子ビームの走査の構造上、走査線の一部が失われます。カメラビュー内にある *内側*の点線の矩形より内部にあるものは、画面上でも見えることが保証されます。二つの矩形の間にあるものはすべて、ビデオの視聴者が使うテレビによって、見える場合も見えない場合もあります。

Dimensions Presets

(寸法のプリセット)



テレビの規格によってレンダリングサイズが厳密に指定されます。制作に便利のように、Blenderには既定の設定が 11 あります：

Preset	Resolution (X x Y)	Aspect Ratio (X x Y)	Frame Rate
DVCPRO HD 1080p	1280x1080	3:2	24 fps
DVCPRO HD 720p	960x720	4:3	24 fps
HDTV 1080p	1920x1080	1:1	24 fps
HDTV 720p	1280x720	1:1	24 fps
HDV 1080p	1440x1080	4:3	23.98 fps
HDV NTSC 1080p	1440x1080	4:3	29.97 fps
HDV PAL 1080p	1440x1080	4:3	25 fps
TV NTSC 16:9	720x480	40:33	29.97 fps
TV NTSC 4:3	720x486	10:11	29.97 fps
TV PAL 16:9	720x576	16:11	25 fps
TV PAL 4:3	720x576	12:11	25 fps

注意点として、たとえば 1600x1200 の解像度でアニメーションをレンダリングして DVD に書き込むと、テレビで見た時に不明瞭な歯切れの悪い絵になります。実際、DVD 書き込みソフトウェアは画像のサイズを上る解像度のいずれかにあわせなければならなくなり、約4倍のディスク容量とレンダリング時間を無駄にします。

Pixel Aspect Ratio

(ピクセルのアスペクト比)

昔のテレビの画面は、コンピューターモニターが持つような正方形のピクセルを持っていません。ピクセルはいくらか長方形で、コンピューターではきれいに見えないがテレビではきれいに見える、*あらかじめ歪めた* 絵を生成する必要があります。レンダリング時にリサンプリングされ画質の低下を招かないように、ピクセルの正確なアスペクト比を使うことが重要です。

Colour Saturation

(彩度)

ビデオテープとビデオ信号のほとんどは RGB モデルではなく YCrCb モデルベースです：より正確にはヨーロッパでは YUV (PAL)、アメリカ合衆国では YIQ (NTSC) です。後者は前者に非常によく似ているため、少し知識も必要です。

YCrCb モデルは情報を '輝度' または強度 (Y) と、二つの '色度' の信号である赤と青 (Cr と Cb) として送信します。白黒テレビは輝度だけを表示しますが、カラーテレビは色度 (および輝度) から色を再現します。RGB から YCrCb 値を得るには、二つの手順を踏みます (*斜体*の定数はシステム (PAL または NTSC) によって変わります)：

まず、ガンマ補正を行います (*g* 値は NTSC で 2.2、PAL で 2.8)：

- $R' = R^{1/g}$
- $G' = G^{1/g}$
- $B' = B^{1/g}$

それから変換します：

- $Y = 0.299R' + 0.587G' + 0.114B'$
- $Cr = a_1(R' - Y) + b_1(B' - Y)$
- $Cb = a_2(R' - Y) + b_2(B' - Y)$

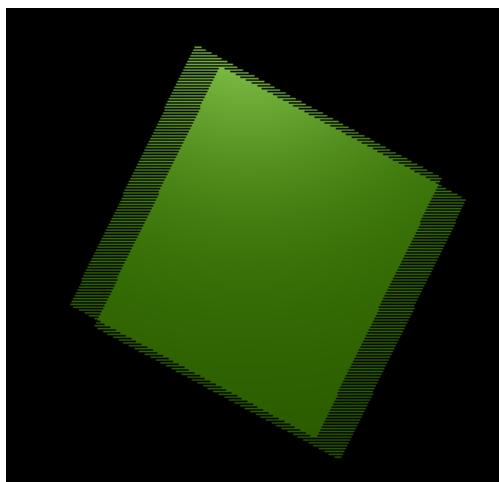
標準的な 24 ビットの RGB 画像は各チャンネルに8ビットを持ちますが、帯域幅を下げるため、また人間の目が色度より輝度に敏感であることを踏まえ、輝度信号は二つの色度信号よりも多いビット数を使って送信されます。このビット拡張の結果、ビデオ内の色の変化はモニターで見慣れたものよりも小さなものにおさえられます。したがってすべての色が正しく表示されているわけではないことを覚えておいてください。

経験から言えば、色をなるべく'灰色がかかったもの'、'非彩色'にとどめてください。これで色の変化を他色との約 80% 以内におさえます。言い換えると、RGB の最高値と最低値の差が、0.8 (範囲が[0-1]のとき)あるいは 200 (範囲が[0-255]のとき)を超えないようにします。

これは厳密なものではなく、0.8 よりいくらか多くても許容できます。ただし、0.0 から 1.0 までのカラーコントラスト範囲を持つ RGB の表示は、コンピュータモニターでは明るく躍動的に見える一方で、ビデオではとても汚く見えるでしょう(彩色過多)。

Rendering to fields

(フィールドへのレンダリング)



フィールドレンダリングの結果

テレビ規格では1秒間あたりのフレーム数を 25 (PAL)または 30 (NTSC)に定めています。画面の発光体は明るさをそれほど長く維持しないため、これは目立ったちらつきの原因になり得ます。

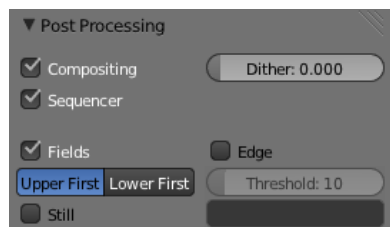
ちらつきを最小限におさえるため、テレビはフレームの描画に、コンピューターの描画方式('プログレッシブ'方式)ではなくハーフフレーム(half-frames)、フィールドを倍のリフレッシュレートで描く方法をとります。したがって PAL では 秒間 50 ハーフフレーム、NTCS では秒間 60 ハーフフレームです。元々は送電線の周波数(ヨーロッパでは 50Hz、US で 60Hz)に縛られていました。

特に各フィールドは "interlace (インターレース/飛び越し)" されます。これはあるフィールドが完全なフレームのすべての偶数番目の走査線を描き、続くフィールドが奇数番目の走査線を描くという意味です。

各フィールド間には無視できないほどの時差があるため(1/50 か 1/60 秒)、フレームを通常の方法でレンダリングして二つのハーフフレームに分ける方法はうまくいきません。動いているものの縁に小刻みなぶれ(jitter)が出ます。

Options

(オプション)



フィールドレンダリングの設定

Fields (フィールド)

フィールドレンダリングを有効にします。Render パネルの *Post Processing* 欄にある Fields ボタンが押されていると、Blender は各フレームを2回にわけて作ります。はじめに偶数行だけをレンダリングし、タイムステップの半分だけ時間を進め、次に奇数行すべてをレンダリングします。PC 画面ではおかしな結果になりますが(図 *フィールドレンダリングの結果*)、テレビでは正しく表示されます。

Upper First (上優先) / Lower First (下優先)

はじめにレンダリングするフレームを偶数番目、奇数番目で切り替えます。

Still

フィールド間の half-frame time step を無効にします (x).

正しいフィールド順序の設定

Blender のデフォルトの設定では偶数フィールドを奇数フィールド *より前に* 作ります。これはヨーロッパの PAL規格に従っています。NTSC では奇数フィールドを先に走査します。

当然ながら、選択を誤るとフィールドレンダリングをまったく使わないよりも結果は悪くなるでしょう。

正しいフィールド順がわからず困っているなら、それを確かめる簡単な方法があります。白い正方形が左から右に動く、黒い背景の短いテストアニメーションをレンダリングします。奇数のフィールド順でひとつ、偶数のフィールド順でもうひとつを作ります。これをテレビ画面で見えます。正しいフィールド順のものは滑らかに見え、そうでないほうはひどい出来になります。この簡単なテストで *何時間もの* レンダリング時間を節約できるでしょう...

フィールドとコンポジットノード

ノードは現時点ではフィールドを認識しません。うまく近傍演算 (blur, vector blur など) を行うには情報が不足しすぎていることが理由のひとつです。解決策はアニメーションをフィールドなしの2倍のフレームレートでレンダリングし、あとから映像を組み合わせてやることです。

Video Files

(ビデオファイル)

以下の形式は、主にレンダリングしたシーケンスを再生可能な動画に圧縮するのに使われます(純粋な音声ファイルを作るのにも使えます)。

コーデック(codec)とは、ビデオを圧縮して DVD におさまるようにしたり、インターネットやケーブルを通じてストリーム出力を可能にしたり、あるいは単に手頃なファイルサイズにしたりするための小さなプログラムです。ビデオのチャンネルを圧縮して容量を節約し、途切れない再生を可能にします。不可逆式(Lossy)のコーデックは画質と引き換えにファイルをより小さくします。H.264 のように、大きな画像に優れているコーデックもあります。コーデックは動画のエンコードおよびデコードに使われます。このため、エンコードを行うマシン(Blender)と対象となるマシンの両方に存在する必要があります。エンコードの結果はコンテナ(container)ファイルに収められます。

数百とは言わずとも数十のコーデックがあります。たとえば XviD、H.264、DivX、Microsoft などがあります。各々長所と短所があり、さまざまな OS やプレイヤーと互換性を持ちます。

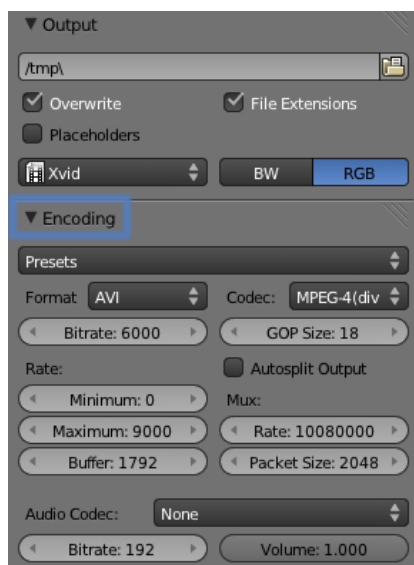
ほとんどのコーデックは RGB または YUV 色空間のみを圧縮しますが、アルファチャンネルを扱えるものもあります。以下は RGBA をサポートするコーデックです:

- animation (quicktime)
- PNG *TIFF
- Pixlet - 可逆式ではありません。おそらく Apple の Mac でのみ使えます
- [Lagarith Loss-less Video Codec\(英語サイト\)](#)

AVI Codec	AVIコーデック圧縮。利用できるコーデックは OS によって違います。AVI codec をはじめに選ぶと、コーデックのダイアログが自動的に起動します。コーデックは AVI Codec 設定にある Set Codec ボタンを使うと直接変更できます。
AVI Jpeg	AVI で Jpeg 圧縮を使います。不可逆で、ファイルは小さくなりますがコーデック圧縮アルゴリズムを使ったときほどは小さくなりません。Jpeg 圧縮はデジタルカムコーダーで使われる DV 形式でも使われます。
AVI Raw	Audio-Video Interlaced (AVI)、フレームは非圧縮
Frameserver	Blender がレンダーファームの一部として フレームリクエスト (frames upon request(eng)) を出します。ポート番号は OpenGL User Preferences パネルで指定されています。
H.264	動画をH.264 コーデックでエンコードします。 高度なエンコード をご覧ください。
MPEG	動画をMPEG コーデックでエンコードします。 高度なエンコード をご覧ください。
Ogg Theora	Theora コーデックで Ogg ファイルとしてエンコードします。 高度なエンコード をご覧ください。 Apple の Quicktime の .mov ファイルです。コーデックがインストールされており、この形式が最初に選ばれると、Quicktime コーデックのダイアログを使えます。 Quicktime Encoding をご覧ください。
QuickTime	QuickTime がインストールされていれば GIF を読めます Windowsと Mac で [QuickTime] がインストールされていれば、Blender は GIF ファイルを読むことができます。GIF の機能(PSDの平坦化、Mac 上での PDF の平坦化他)は QuickTime についてきます。
Xvid	Xvid コーデックでエンコードします。 高度なエンコード をご覧ください。

Advanced Encoding

(高度なエンコード)



H.264、MPEG、Ogg Theora、Xvid のいずれかのコーデックが選ばされると、Encoding パネルが利用できるようになります。FFmpeg を使ってこれらのファイル形式やその他の形式をエンコードするための設定があります。

[FFmpeg\(英語サイト\)](#) は多数の形式でデジタル音声およびビデオの記録、変換、ストリームを行う、フリーでオープンソースのソフトウェアライブラリを集めたものです。この中には複数の他プロジェクトで使われている音声/ビデオコーデックライブラリである libavcodec や、音声/ビデオコンテナの mux および demux ライブラリである libavformat があります。

Video Settings

(ビデオの設定)

使いたいビデオコーデックの選択と圧縮の設定を行えます。圧縮形式の選択については、ファイルサイズ、プラットフォーム間の互換性、再生品質の間に妥協点があります。

[システムコンソール](#) を見ると、エンコード処理の出力の一部を見ることができます。 `blender -d` で Blender を起動すれば、さらなる出力を見ることができます。

DV、SVCD、DVD などの既定セットを使うことができます。そのタイプの出力に最適な設定を選んだものです。もしくは手動で形式 (MPEG-1、MPEG-2、MPEG-4、AVI、Quicktime (インストールされている場合)、DV、H.264、Xvid (インストールされている場合)) を選ぶことができます。Blender がビデオストリームを圧縮するためには、お使いのコンピューターに正しいコーデックがインストールされている必要があります。

Video Formats

(ビデオ形式)

名称	拡張子	概要
MPEG-1	.mpg, .mpeg	ビデオおよび音声向けの、不可逆な圧縮の規格です。VHS 品質の raw デジタルビデオおよび CD 音声を 1.5 M ビット/秒 まで圧縮するように設計されています。
MPEG-2	.dvd, .vob, .mpg, .mpeg	"動画および付随する音声情報の共通符号化" のための規格。不可逆なビデオ圧縮と不可逆な音声データ圧縮方式の組み合わせで、現在利用できる記憶媒体や伝送帯域幅を使った動画の保存や伝送を可能にする、と書かれています。
MPEG-4(DivX)	.mp4, .mpg, .mpeg	MPEG-1 や MPEG-2、および関係する他の規格の多くを取り入れ、新たな機能を追加しています。
AVI	.avi	Resource Interchange File Format (RIFF) の派生で、ファイルのデータをブロック、"チャンク(chunks)"に分割します。
Quicktime	.mov	マルチトラックの形式です。QuickTime と MP4 コンテナ形式は同じ MPEG-4 形式を使えます。ほとんどは QuickTime-only 環境で互換性があります。MP4 は国際規格で、より多くのサポートがあります。
DV	.dv	intraframe ビデオ圧縮方式で、フレームごとにビデオを圧縮するのに離散コサイン変換 (DCT) を使います。音声は圧縮されずに格納されます。
H.264	.avi ("当面")	ビデオ圧縮の規格で、現在もっとも一般的に使われる形式の一つです。録画、圧縮、高解像度ビデオの配布に使われています。
Xvid	.avi ("当面")	MPEG-4 規格に続くビデオコーデックライブラリです。Bフレーム、グローバルおよび1/4動き補償、lumi masking、格子量子化、および H.263、MPEG およびカスタム量子化行列といった ASP 機能を使います。Xvid の一番の競合相手は DivX Pro コーデックです。
Ogg	.ogg, .ogv	フリーで不可逆なビデオ圧縮形式です。Xiph.Org Foundation によって開発され、ライセンス料なしで配布されています。
Matroska	.mkv	オープンでフリーなコンテナ形式の規格です。コンテナ形式とは任意の数のビデオ、音声、画像、字幕トラックをひとつのファイルに保持できるファイル形式です。
Flash	.flv	インターネットで Adobe Flash Player を使ってビデオを配信するのに使われる、コンテナファイル形式です。
Wav	.wav	非圧縮の(または軽く圧縮された)Microsoft および IBM の音声ファイル形式です。不可逆なデータ圧縮を使う高圧縮率のデジタル音声のエンコード形式で、特許で保護されています。デジタル音声

Mp3 .mp3 の圧縮やデジタルオーディオプレイヤー上での音楽の再生および転送において業界標準であり、また民生用の音声ストレージ用音声形式としても一般的です。

Video Codecs

(ビデオコーデック)

名称	概要
None	音声だけのエンコード用。
MPEG-1	(上の ビデオ形式 をご覧ください。)
MPEG-2	(上の ビデオ形式 をご覧ください。)
MPEG-4(DivX)	(上の ビデオ形式 をご覧ください。)
HuffyUV	可逆式ビデオコーデックで、ビデオキャプチャの形式として非圧縮の YCbCr を置き換えるべく Ben Rudiak-Gould 氏によって作られました。
DV	(上の ビデオ形式 をご覧ください。)
H.264	(上の ビデオ形式 をご覧ください。)
Xvid	(上の ビデオ形式 をご覧ください。)
Theora	(上の ビデオ形式 の Ogg をご覧ください。)
Flash Video	(上の ビデオ形式 をご覧ください。)
FFmpeg video codec #1	FFV1 とも言います。可逆式 intra-frame のビデオコーデックです。可変長符号化 (variable length coding) もエントロピー符号化 (entropy coding) 用の算術符号化 (arithmetic coding) もできます。エンコーダーとデコーダーは FFMpeg 内にあるフリーでオープンソースのライブラリ libavcodec の一部です。

Options

(オプション)

Bitrate (ビットレート)

フレームごとのビット数である平均 [ビットレート](#) (品質) を設定します。 [ffmpeg -b:v\(英語サイト\)](#) もご覧ください。

Rate (レート)

ビットレートの調節欄には Minimum と Maximum もあります。

Buffer (バッファ)

[decoder bitstream buffer\(英語サイト\)](#) のサイズ。

GOP Size (GOPサイズ)

[Group of Pictures](#) ごとに含まれる画像の数。0にすると "intra_only" になり [inter-frame\(英語サイト\)](#) ビデオを無効にします。ffmpeg のドキュメントにはこう書かれています: "非常に低いビットレートでのストリーミングには、低いフレームレートと小さな GOP を使ってください。特に、Linux player がそれほど高速でなく、フレームが欠ける場合の RealVideo に当てはまります"

Autosplit Output (出力の自動分割)

ビデオが 2G を超える巨大なものなら、自動分割を有効にしてください。出力ファイルサイズの主制御には GOP または keyframe interlace を使います。通常は数値を上げるほど小さなファイルができますが、再生にはより高性能なデバイスが必要です。

Mux (マルチプレクサ)

[Multiplexing\(英語サイト\)](#) の設定。

Rate

マルチプレクサストリームの最大ビットレート。

Packet Size

(ffmpeg のドキュメントに記述がありません)

標準

コーデックは突飛なビデオサイズをエンコードできません。したがって XY サイズには標準的なテレビサイズ用のプリセットで使われているものだけを使ってください。

Audio Settings

(音声の設定)

選択したコーデックを使って音声がエンコードされます。

音声コーデック (Audio Codecs)

[MP2](#) 不可逆な音声圧縮形式で、ISO/IEC 11172-3 で定義されました。

[MP3](#) (上述 [ビデオ形式](#) の MP3 をご覧ください)

[AC3](#) Audio Codec 3 はドルビーラボラトリーズが開発した音声圧縮技術です。

[AAC](#) "先進的音響符号化 (Advanced Audio Codec)" は規格化された、デジタル音声向けの不可逆な圧縮およびエンコード方式です。MP3

AAU 形式の後継になるべく設計され、通常は同様のビットレートの MP3 よりも音質がよくなります。

Vorbis MP3 や AAC に相当する、オープン標準で高圧縮の形式です。AAC や Windows Media Audio のような他の MDCT 形式に匹敵する複雑な計算を行う一方、これらと同等か高い音質の小さなファイルを作ることから、他の不可逆な音声形式の多くよりも性能が優れていることが過去に証明されてきました。

FLAC フリーの可逆式音声コーデック (Free Loss-less Audio Codec) です。FLAC のアルゴリズムで圧縮されたデジタル音声は元のサイズの 50-60% に抑えられ、復元すると元の音声データとまったく同じコピーになります。

PCM Pulse Code Modulation はサンプリングしたアナログ信号をデジタルで表すのに使われる方式です。コンピュータや各種 Blu-ray、CD および DVD 形式、あるいはデジタル電話システムなどのその他の用途にも使われる標準的な形式です。

Bitrate

各コーデックに対して、動画内の音声のビットレート(品質)を調節できます。この例は 128kbps での MP3 のエンコードです。ビットレートを高めるほどファイルが大きくなりストリームが重くなりますが、音は良くなります。互換性のため、2のべき乗になるようにしてください。

Samplerate

音声の1秒あたりのサンプル数を調節します。デフォルトの 44100 は CD 音声を含む多くのファイル形式の標準で、高品質なサウンドを作ります。

Volume

音声の音量を設定します。

Tips

(ヒント)

選択すべき形式は、画像を使って何を行うかによります。

動画をポストプロセスや特別なエフェクトなしで作るのなら、**AVI-JPEG** か **AVI Codec** を使い、XviD のオープンなコーデックを選びます。VSE に読み込んだ音声とともに動画を出力するのなら、**FFMPEG** を使います。

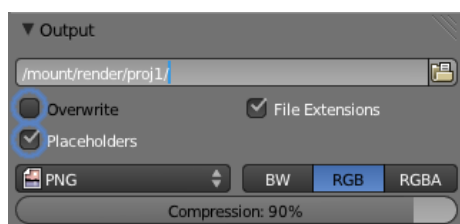
動画にポストプロセスを施すつもりなら **OpenEXR** 画像としてレンダリングされたフレーム一式を使うのが一番です。ファイルを一つだけにしたいのなら、**AVI Raw** を選びます。AVI Raw は巨大ですが、ポストプロセス用に出力の正確な品質を維持します。compositing/sequencing のようなポストプロセスのあとで、圧縮できます。圧縮済みのファイルにポストプロセスを施したいとは思わないでしょう。圧縮による画像の歪み (artifacts) が、ポストプロセスで行った作業を消してしまう可能性があるからです。

なお、ビデオ形式に直接レンダリングしたいとは思わないかもしれません。レンダリング中に問題が起きれば、はじめからすべてのフレームを再レンダリングする必要があります。もしはじめに一連の静止画(デフォルトの PNG、あるいはより高品質な OpenEXR)をレンダリングすれば、[Video Sequence Editor \(VSE\) \(eng\)](#) 内で Image Strip を使ってつなぎあわせることができます。この方法では、以下のことが簡単にできます:

- 問題発生箇所(フレーム)からレンダリングを再開する
- (数分や数時間でなく)数秒で別のビデオオプションを試してみる
- VSE の他の機能を楽しむ。たとえば前のレンダー、音声、ビデオクリップ等から Image Strip を追加します。

Home-made Render Farm

(自家製レンダーファーム)



レンダリングの作業負荷を複数マシンで分担する簡単な方法です:

1. 共有ディレクトリを準備します(例: Windows 共有や NFS マウント)
2. "Overwrite (上書き)" のチェックを外し、"Placeholders (場所の確保)" にチェックします
3. 望むだけの数のマシンを使って、そのディレクトリへのレンダリングを始めます。マシン同士がお互いの足を踏んでしまうことはありません。

Post Processed Effects

Render 設定では、レンダリングの完了後に、レンダリング画像に視覚要素を追加するようなエフェクトを有効にできます。カメラ内で行われるのではなく、画像の上に合成されるものです。

Composited(ノード合成)と Sequence(VSE/シーケンサー) については [出力のオプション \(eng\)](#) で説明されています。

Fields については [動画の出力 \(eng\)](#) に説明があります。

Edge (Toon) Rendering



Toon マテリアルを使ったシーン

Blender のトゥーンシェーダーは塗り色の濃淡に影響を与え、レンダリングをコミック調もしくはマンガ調の見た目にすることができます (図「Toon マテリアルを使ったシーン」)ほんとうのコミックやマンガはふつう筆書きの輪郭線を持つので、トゥーンシェーダーだけでは完成とはいえません。Blender は輪郭線をポストプロセス操作として追加します。

Freestyle

輪郭線の描出にはこの機能の他に、Blender 2.67 から同梱されている Freestyle を使うことができます。詳しくは [Freestyle](#) をご覧ください

Options

(オプション機能)

以下のオプションは Render(レンダー) コンテキストの Post Processing (ポストプロセッシング) パネルにあります。

Edge (辺)

Blender がレンダリング結果から境界線を探し、「縁取り」を足します。

レンダリングを繰り返す前に以下のパラメータの設定が必要です。

Threshold (しきい値)

境界線を描く辺の両側の面が作る角度のしきい値です。0 から 255 の値です。10 にすると背景に対するオブジェクトの輪郭だけに境界線を与えられます。値を高くするとオブジェクト表面の辺にも、初めは尖った辺から境界線が追加され始めます。最大強度では、完全に滑らかではないサブサーフ辺でさえもかすかに境界線が表示されます。

色 (RGB)

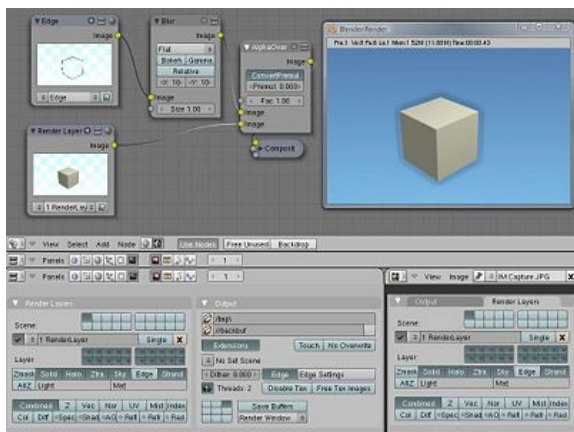
レンダリングされた境界線の色です(デフォルトは黒です)。スウォッチをクリックしてカラーピッカーを表示します

Examples

(例)



トゥーンエッジセットを使って再レンダリングしたシーン



専用のレンダーレイヤーを使って、境界線のレイヤーを分離することができます。境界線のない部分のアルファチャンネルは 0、ある部分は 1 です。境界線のレイヤーを分離することで、ぼかし(blur)、色の変更、マスクなどを行えるようになります。下の図はこの方法を示しています。私はまず、SkyとEdgeレイヤーのみを持つ境界線用のレンダーレイヤーを作りました(Skyを入れたのは、後ほどコンポジットの出力でWorldカラーを取得するためです)。他のレンダーレイヤーでは境界線のレイヤーを省き、通常の画像だけを返すようにします。Outputパネルで幅10の黒い境界線を有効にしました。このレイヤーをBlurノードに通します。続いてAlphaoverノードを使って、ぼかした境界線の上にキューブを合成します。やわらかい影を持つような効果が得られます。なお境界線の画像はすでに1.0のアルファ値を持っているため、Premultiplyがセットされています。

Dithering

(デザ処理)

Ditheringはグラデーション内に帯状のむらができる色と色の間に段ができるのを避けるために、ピクセルをぼかす技術です。帯状のむらはグラデーションが長くなるほど、または勾配がゆるいほど、目につきやすくなります。ディザリングは低ビット深度のグラフィック用に開発されたものなので、使える色の範囲は限定的です。

ディザリングはピクセル値をしきい値や隣接ピクセルと照らし合わせ、適切な色を生成する計算を行うことで動作します。色を並べて、引きの絵で別の色に見えるような知覚効果を作ります。例えば赤と黄色のピクセルを交互に格子状に並べると、画像はオレンジ色に見えます。

Ditherは0から2の範囲の値です。

Stamp

(スタンプ) スタンプパネルには画像スタンプ用のオプション機能があります。スタンプはレンダリングした画像の上にテキストを追加します。なおスタンプ情報は対応している画像形式(現時点で JPEG、EXR、PNG)の画像のメタデータに書き込むこともできます。スタンプのチェックボックスが外れているときも、チェックされた項目についてメタデータがファイルに書き込まれます。

スタンプには次のデータを含むことができます:

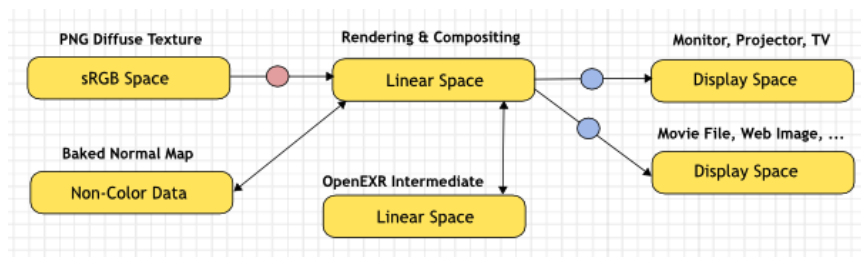
Time	シーンの経過時間とフレーム番号。HH:MM:SS.FF 形式
Date	現在の日時
RenderTime	レンダリング時間
Frame	フレーム番号
Scene	アクティブなシーンの名前
Camera	アクティブなカメラの名前
Lens	アクティブなカメラのレンズ値
Filename	.blend ファイルのファイル名
Marker	直前のマーカーの名前
Seq. Strip	前面にあるシーケンスストリップの名前
Note	自由な註釈
Stamp Text Color	スタンプの文字の色と透明度
Stamp Background	文字の背景の色と透明度
Font Size	文字のサイズ

Blender 2.64: 色管理

[OpenColorIO](#)がBlenderに統合され、これにともない色管理システムが再設計されました。以前のBlenderはリニア(Linear)とsRGBの二つの色空間しか対応していませんでしたが、今はもっと多く対応し、どの色変換を使用すべきかを細かくコントロールできます。

シーンリニア(Linear)色空間

正しい結果を得るには、レンダリング、表示、画像の格納にそれぞれ違う色空間が必要です。レンダリングとコンポジットはシーンリニア色空間で行うのが最適で、これは自然により近く対応し、計算もより物理的に正しい物になります。



サンプルリニアワークフロー

色がリニアということは、現実でフォトンの量を二倍に増やした場合に、その色の値も二倍であるということを意味します。別のいい方をすると、それぞれ二つのライトの内の一つを持つ、二つの写真/レンダリング画像を持っている場合、これらの画像をいっしょに加算合成した結果が、両ライトを照らしてレンダリングした画像や写真と同じになるということです。このような放射分析的なリニア空間はフォトリアスティックなレンダリングとコンポジットに最も適しています。

しかし、これらの値は直接人間の知覚やディスプレイデバイスの動作方法には対応しておらず、さらに画像ファイルは大抵別の色空間で格納されているため、このリニア色空間内や外への正しい変換を注意して行わなければなりません。

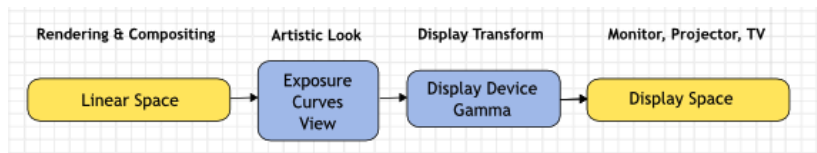
表示

レンダリング画像の正しい表示には表示デバイスの色空間への変換が必要ですが、これが設定できるようになりました。コンピュータのモニターの動作はデジタルシネマプロジェクトやHDTVとは違います。シーン(Scene)プロパティにこれらの設定があります。

デバイス(Device)

画像が表示されているデバイス。大半のコンピュータモニタはsRGB色空間用に調整されており、コンピュータ上で作業する場合は、通常はこのオプションはデフォルトのままにしておく方がいいでしょう。これは主に画像をコンピュータにつなげた他の表示デバイスで見る時や、画像ファイルを他のデバイス上で表示するつもりで出力する時に変更します。

Rec709は一般的にHDTVで使用され、一方、デジタルプロジェクターではXYZとDCI-P3が一般的です。デバイスをなし(None)に設定することで色管理を無効にできます。

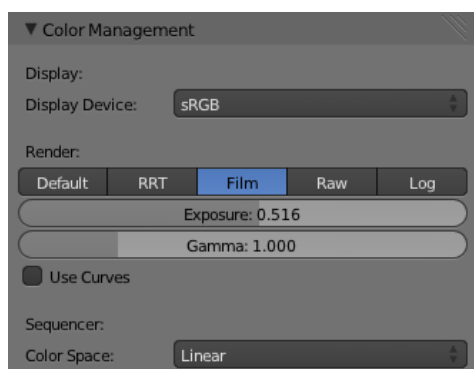


リニアから表示デバイス空間への変換

レンダー

さらにレンダー用につくられた、アーティスティックチョイスがあります。その理由の一つに表示デバイスは色域を完全には表示できず、限られた輝度しか持たないことがあり、色をデバイスの全範囲に適合するよう、色を詰め込むことができます。これに加え、レンダーを特殊な見た目、例えば現実のフィルム上にプリントされたようにするのも便利です。

別の一般的な使用ケースは、レンダリングの検査をしたい場合に、暗い影の中や明るいハイライトの細かい部分を見たり、レンダリングエラーを確認するなどがあります。このような設定は一時的に使用されるだけで、最終レンダーには使用されません。



色管理用のシーン(Scene)設定

ビュー変換(View Transform)

これらは同じ表示デバイス上で画像を見る別の方法です。デフォルトではその表示デバイス用の変換以外は何も追加で変換を行いません。RRTはACES Reference Rendering Transformを使用し、フィルム状の見た目をシミュレートします。フィルム(Film)オプションは別のフィルム

状の見た目です。RawとLogは画像の検査用を意図しており、最終的なエクスポート用ではありません。Rawはどの色空間の変換もない画像を与え、一方Logは非常に暗いまたは明るい領域のない、より「フラット」な画像のビューを与えます。

露出(Exposure)

色空間の変換が行われる前に適用される、画像の輝度への乗数です。

ガンマ(Gamma)

色空間の変換が行われた後に適用される、追加のガンマ補正です。デフォルト sRGB または Rec709 色空間変換にはすでに約2.2へのガンマ補正が含まれているため(RawとLogビューを除く)、これがそれに追加で適用されることに注意してください。

カーブを使用(RGB Curves)

色空間の変換前に画像の色をコントロールするカーブです。

画像ファイル

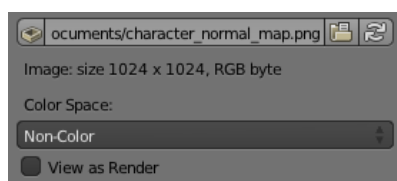
他の場所で色管理を維持するのに心がけておくことは、いつ**画像ファイルの読み書き**を行うかです。PNGやJPEGなどのファイルフォーマットでは通常、表示可能な色空間に格納されており、リニア空間ではありません。これらが例えばレンダー内でテクスチャとして使用されている時、まずリニアに変換する必要があり、Web上での表示用にレンダリング画像を保存する時も、表示空間に変換する必要があります。OpenEXRのような、他のファイルフォーマットは、制作での中間ファイルとして便利のように、リニア色空間で格納されています。

画像ファイルで作業する場合、普通はデフォルトの色空間が正しい物となります。このケースに当てはまらない場合、画像ファイルの色空間は▼画像(Image)で設定できます。手動による変更が必要な一般的な状況は、例えばノーマルマップやディスプレイメントマップの作業やベイクをしている時です。このようなマップでは、実際には色ではなく、単に色としてエンコードされたデータを格納しているだけです。このようなケースではNon Colorを指定してください。

画像(Image)データブロックは常にメモリ内の浮動小数点数(float)バッファにシーンリニア色空間で格納され、一方、メモリ内のバイトバッファとディスク上のファイルでは、この設定により指定された色空間で格納されます。

色空間(Color Space)

ディスク上の画像の色空間。これはファイルフォーマットに依存し、例えばPNGやJPEG画像ではよくsRGBで格納され、一方、OpenEXR画像はリニア色空間で格納されます。ノーマルやパンプ、ステンシルマップのようないくつかの画像では厳密には「色」を格納しておらず、このような値では絶対に何も色空間変換を適用してはいけません。このような画像では色空間はなし(None)を設定してください。



色管理用の画像(Image)設定

デフォルトではレンダリング画像のみレンダービュー変換が適用されて表示・保存されます。これらは Render Result と Viewer Image データブロック、そしてアニメーション(Render Animation)オペレーターにより直接ディスクに保存されたファイルです。しかし、中間物の OpenEXR ファイルに保存されたレンダリング画像の読み込み時では、例えば画像テクスチャもしくはディスプレイメントマップの可能性があるので、Blenderは自動的にこれがレンダリング画像と探知できません。下記の二つの設定で、これがレンダリング画像だと教え、適用してほしい変換を指定する必要があります。

ビューにも適用(View as Render)

Image データブロックを(レンダリング画像に限らず)ビュー色変換、露出、ガンマ、RGBカーブを適用して表示します。リニア OpenEXR ファイルでレンダリングしたフレームを、直接レンダリングした時と同じ状態で見ると便利です。

レンダー色空間で保存(Save as Render)

画像保存オペレーターのオプションで、ビュー色変換、露出、ガンマ、RGBカーブを適用します。これは例えばリニア OpenEXR を PNG や JPEG ファイルに、表示空間で保存するのに便利です。

シーケンサー(Sequencer)

シーケンサー色空間(Sequencer Color Space)

シーケンサーが動作する色空間。デフォルトではシーケンサーは以前のバージョン同様、sRGB 空間で動作しますが、コンポジットノードのようにリニア空間や他の色空間で動作させることもできます。別の色空間では色補正(Color Correction)やクロスフェード(Cross Fade)などの操作で違う結果になります。



同じレンダリング画像の違う見た目と露出

OpenColorIO 設定

Blender に多数の便利な表示デバイスとビュー変換の入った、標準の OpenColorIO 設定が付きましました。使用している参照リニア色空間は、Rec. 709色度とD65白色点のリニア色空間を使用しています。

しかし、OpenColorIO は [複数のアプリケーション間](#)でのユーザエクスペリエンスが一致するよう設計されており、そのために、単一の共有設定ファイルを使用することもできます。Blender は Blender デフォルトの設定ではなく OpenColorIO 設定を読み込むのに、標準の OCIO 環境変数を使用します。ワークフローなどの詳しい設定方法については、[OpenColorIO webサイト\(英文\)](#)を参照してください。

現在、以下の色空間の役割を使用しています。

- *scene_linear*: レンダリング、コンポジット、メモリ内に格納されるすべての Float 精度の画像用に使用されます。
- *default_sequencer*: シーケンサーのデフォルト色空間。指定されていない時は *scene_linear*を使用します。
- *default_byte*: バイト精度の画像とファイル用のデフォルト色空間。指定されていない時は *texture_paint*を使用します。
- *default_float*: Float 精度の画像とファイル用のデフォルト色空間。指定されていない時は *scene_linear*を使用します。

別のリニア色空間はあるものの、標準の Blender 設定も [ACES用\(コードとドキュメント\(英文\)\)](#)にいくつか対応しています。リニア ACES 色空間による EXR ファイルの読み書きでき、RRT ビュー変換を画像の表示にその標準の表示変換で使用可能です。ただし、ACES 全範囲は Rec. 709全範囲に比べて大きいため、最良の結果を得るには、ACES 仕様の設定ファイルを使用すべきでしょう。制作に使用できるようにするには若干調整が必要なものの、OpenColorIO は [ACES 設定](#)を提供しています。

互換性

現存のファイルとの互換性はほぼ維持されています。色管理を有効にしていたファイルは完全に互換性があり、一方、色管理オプションが無効な古いファイルでもほぼ互換性は保たれていますが、頂点カラーとビューの色に違いがあります。

[Developer Documentation](#)

(訳注: 英語記事は空っぽですが [チュートリアル](#) に簡単な説明があります)

Motion Blur

(モーションブラー)

デフォルトでは Blender のアニメーションは、一連の *完全な静止画* としてレンダリングされます。これは現実味に欠けます。なぜなら、'現実世界のカメラ' で撮られた映画のフレームと写真のどちらにおいても、高速移動するものはその動きでぶれている (blurred) ことにより '動いている' ように見えるからです。このような「動きのぶれ/モーションブラー」の効果を得るために、Blender は 現在のフレームと、実在のフレームの合間に数フレームをレンダリングして溶け合わせ、高速移動する箇所が 'ぶれた' 画像を得ます。

The Human Eye

(人間の眼)

我々の脳は両目からの絵を同時に、毎秒約15枚処理しています。これらの絵を脳は cognate し、二つを比較することで動きを知覚します。十分速く動くものがあれば、脳はそれがぶれていると知覚します (桿体細胞の光への反応に少し遅延があること、あるいは脳が絵を重ね、差異を知る過程でどういうわけか溶け合わせて混合物にすることが原因です)。要するに、脳は動きのぶれを *知覚* しません。

In Film

(映画)

ものがびくびくと揺れて見えないように、映画は毎秒のフレーム数 (fps) を単純に倍にして 30 フレームにしました (ヨーロッパでは 24 fps)。したがって、シャッターは基本的に1/30秒間開いており、フィルムがその時間だけ世に露出します。この間に現実世界のものが動くと、そのフレームにはフィルムの露出によって、動いたものが物理的にぶれるか不鮮明になった絵ができます。現像して上演されたとき、我々が見るのは物理的にぶれた絵です。要するに、我々はぶれた絵を *見ています*。

In CG

(CG)

CG では、コンピュータはフレームを描画するとき、どこに何があるかを正確に知っており、その通りに描画します。フレームを追うごとに、ある物体はフレーム1では A に、フレーム2では B に位置します。この二つのフレームを速く見せられると、物体は飛び跳ねて見えます。眼球とフィルムの間にはどこにも、現実世界や映画にあるのと同じぶれは存在せず、差異を見分けることができます。

Motion Blur in Blender

(Blender におけるモーションブラー)

それで、どうすればぶれた CG 画像を作れるのでしょうか。Blender にはモーションブラーを実現するための二つの方法があります:

Sampled Motion Blur

(サンプル抽出によるモーションブラー)

この方法は遅いですが、結果はよくなります。レンダーオプションパネルの、モーションブラー欄で有効にできます。

Motion Samples

各フレームごとに取得するサンプルの数を設定します。

Shutter

シャッターを開いてから閉じるまでの時間をフレーム数で指定します。

Vector Blur

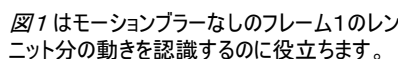
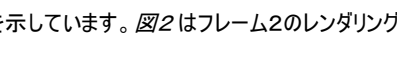
(ベクトルブラー)

[Vector Blur](#) は高速ですが望まない副作用を起こすことがあります。ただし、これは回避できます。ベクトルブラーはコンポジティングでなされる処理で、ぶれの一切ないシーンを、各ピクセルの動き情報を持ったパスと合わせてレンダリングします。この情報には2次元か3次元の方向および強さが記述されます。コンポジターはこのデータを使い、指定方向にある各ピクセルをぶれさせます。

Examples

(実例)

概念をよく把握するために、立方体が毎フレーム1 Blenderユニットだけ均一に右に移動することを考えてみましょう。立方体自身の一辺が2 Blenderユニットしかないため、これはとても高速です。

1 はモーションブラーなしのフレーム1のレンダリングを示しています。2 はフレーム2のレンダリングです。立方体の下にある目盛りは、1 Blenderユニット分の動きを認識するのに役立ちます。

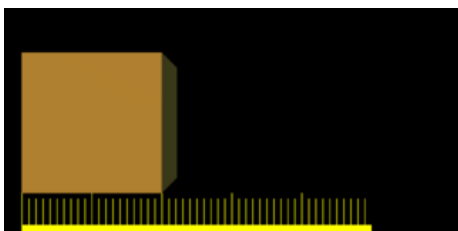


図1. モーションブラーなしで立方体を動かしたときのフレーム1

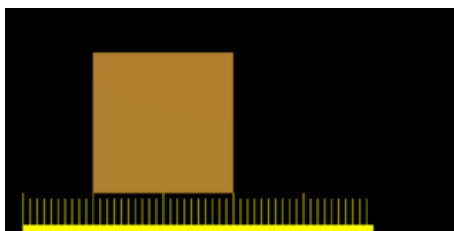


図2. モーションブラーなしで立方体を動かした時のフレーム2

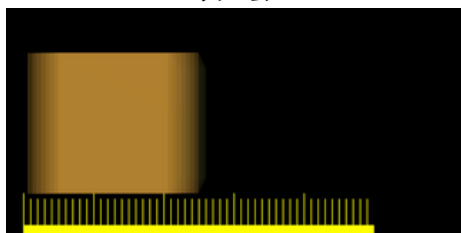


図3. サンプル数8、Shutter=0.5のモーションブラーつきで立方体を動かした時のフレーム1

一方 図3は、モーションブラーがセットされ8つの'中間'フレームが算出されたときのフレーム1のレンダリング結果です。Shutterは0.5にしています。これは、フレーム1から始めて、0.5フレームの区間ごとに8つの'中間'フレームを算出するという事です。立方体の'ぶれ'はすべて、立方体本体の前0.5ユニットと後ろの0.5ユニットで起きていることが非常にはっきりとわかります。

図4と 図5はシャッター値を増やした結果を示しています。1より大きな値はとても'遅い'カメラシャッターを意味します。

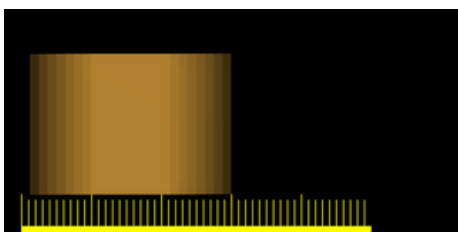


図4. サンプル数8、Shutter=1.0のモーションブラーつきで立方体を動かした時のフレーム1

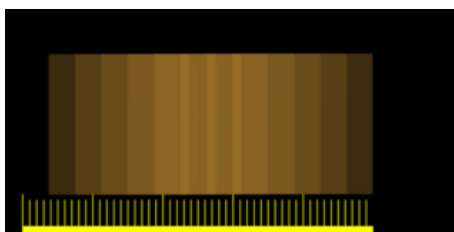


図5. サンプル数8、Shutter=3.0のモーションブラーつきで立方体を動かした時のフレーム1

サンプル数を8ではなく11や16にすることでここで見られるものよりよい結果が得られますが、当然ながらサンプル数と同じだけの'別々の'レンダリングが必要になるため、モーションブラーを使うレンダリングは使わないレンダリングより何倍も時間がかかります。

Hints

(ヒント)

シーン内で何も動いていなくても、モーションブラーが有効なら Blender は実際には'中間'フレームと次のフレームの間で少し、カメラを小刻みに揺らします(jitter)。これは、アンチエイリアシングがオフであったとしても、結果の画像にうまくアンチエイリアシングがかかるということです。モーションブラーによるアンチエイリアシングは同レベルのアンチエイリアシングに相当しますが、普通は遅くなります。

面白いのは、レベル16のアンチエイリアシングが満足な結果を生まないような複雑なシーンでは、アンチエイリアシングとモーションブラーの両方を使うと結果がよくなることです。こうするとフレームごとのサンプル数は'中間'フレームの数と同じになり、5、8、11、16サンプル数が選ばれると、それぞれ25、64、121、256レベルのオーバーサンプリングを実際にもたします。

Optimizing Render Performance

(レンダリング効率の最適化)

「待つ身は長い」は古いことわざですが、レンダリングに時間がかかりすぎたり、さらには途中でクラッシュしたりするのがなぜなのか、疑問に思ったことがあるかもしれません。レンダリングの高速化のために、あるいは複雑なレンダリングをやり遂げるために、できることはたくさんあります。それに「かしこいレンダリング」を行えば、平凡な PC で非常に複雑なシーンをレンダリングできます。以下はレンダリングの高速化のために、果てはシーン描画中のクラッシュ回避のために、すべきことすべきでないことの「トップ10」リストです。選択肢の中には描画品質を下げるものもありますが、試しレンダリング用途であれば気にならないでしょう。

「Malloc returns nil」というメッセージが出たら、メモリアロケータが Blender 用に物理メモリを追加取得しようとして手ぶらで帰ってきたということです。つまり、シーンのレンダリングに使うメモリが足りず、Blender が処理を続行できません。レンダリングを行うには、このページにあるタスクをひとつ以上行う必要があるでしょう。

Hardware Improvements

(ハードウェアの更新)

1. PC (マザーボードや OS) の上限まで RAM を増やす。まもなく Blender は 8GG (ギガギガ) までの物理メモリ (64ビットアドレス空間) を使えるようになりますが、大半の PC は 4G の RAM だけを扱えます
2. CPU をマルチコアマルチプロセッサに更新
3. OpenGL ビデオドライバの更新
4. もっと速いメモリバスを入手
5. マザーボードの上限まで高速なメモリを入手。667MHz のメモリは 800MHz のものより 30% 低速です
6. 家にある使える PC をすべて使って render farm を始める準備をする。もしくは BURP のような render farm を使います

Operating System Configuration

(OS の設定)

1. OS で Blender の処理優先度を上げる
2. OS がメモリスワッピングに使用するスワップファイルの容量を増やす。仮想メモリのページファイルサイズとも呼ばれます。物理メモリサイズまで増やします
3. 64ビット OS に更新 (いまお使いでなければ)
4. ウィルス検知、BOINC、Real、さらには Quicktime などの「活動中でない」もの (更新を調べるのにランダムで起動するため) なども含めた、バックグラウンド処理を終了させるか停止させます。バックグラウンド処理は RAM も使います
5. ネットワーク接続を無効にして、ランダムな ping や更新用データの通信を止めます
6. インターネットラジオを聴くのやめてウェブブラウザを閉じる。特にマルチメディアサイト (音声/動画/ゲーム) は閉じます
7. Word などの、実行中の他のアプリケーションをすべて閉じる
8. すべての常駐プログラム (システムトレイにアイコンがあるもの) とウィジェット、バックグラウンド処理、ウィルス検知プログラムを終了させる

Blender Settings

(Blender の設定)

1. ユーザー設定の、システム & OpenGL タブで最大メモリキャッシュ (Memory Cache Limit) を増やす
2. [最適化された Blender ビルド](#) に更新する。SSE2 をサポートする先進的なチップをお使いなら特に、最適化されたビルドを使います。レンダリング時間が 30% 高速化します
3. 平行投影 (Orthographic) カメラに切り替え、シーンにある「部品」を別々の画像としてレンダリングし、GIMP で貼り合わせる。現実のカメラでパノラマ写真を作る古い方法として、1枚目を撮ったら右に向きを変えて別の写真を撮り、さらにもう一枚を撮るというやり方で、非常に幅広の (浜辺の日没) 風景を3枚ほどの写真に撮影する方法があります。写真を現像したら重ね合わせて、幅広の横長画像を作ります。Blender でも同じことをしましょう。1場面をレンダリングしてファイルに出力したら、シーンの別の場所にカメラを合わせてその場面をレンダリングします。撮影ごとの領域が狭くなるので、カメラの捉えるポリゴン/面の数も少なくなります。あとで貼り合わせられるように、重ね合わせの撮影に使うカメラをいつ動かすか、間違えないようにしてください。GIMP を使いたくない場合は、Blender のコンポジティングノードで Translate (移動) ノードを使って組み合わせることができます
4. レンダリングウィンドウを (Blender 内部ウィンドウにレンダリングするなら Blender も) 最小化。毎フレーム劇的に高速になるという ATI ユーザーの報告があり、これはフレームが増えるほど意味があります
5. 「Big Render」スクリプトを使って画像を細切れにレンダリングし、それを貼り合わせる
6. カスタマイズされたビルドを作る。たとえば、convertblender.c にある check_non_flat_quads への呼び出しをコメントアウトします。20-40% の顕著な違いが出る場合があります

Scene and Specific Objects

(シーンと特定のオブジェクト)

1. 光源をなくす。または、レンダリングしないレイヤーに移動するかレイヤーに限定します
2. 光源の影をいくつかオフにして、1つか2つのメインの太陽光源だけが影を作るようにする。少数の「影のみ」光源は、全光源が影を持つ場合よりも高速にレンダリングできます
3. ray-traced Shadows ではなく buffer Shadows を使う
4. 動かない面に、Full Render バイクを使って影を焼き付け。できたテクスチャを同じメッシュに持って、このマテリアルに対する影を無効にします
5. メッシュを単純化する (ポリゴンを減らす)。カメラに入る頂点が多いほど、レンダリングに時間がかかります
6. 重複を取り除く (Remove Doubles)。または Decimator メッシュ編集機能を使う
7. Subsurf や Multires モディファイアを外す
8. メッシュの裏側を削除 (見えないジオメトリを取り除く)
9. 一度に少しのオブジェクトだけをレンダリングする。常に変更されることのない、常に背景にあるオブジェクトや舞台装置をプロジェクト開始時にレンダリングします
10. 建物を別のレイヤーに置き、レンダーレイヤーを通してレンダリングしないようにする。あとで合成して戻します
11. 上の二つのアイデアを達成しやすくするためカメラを固定する

12. 範囲光源 (Area lights) の使用を避ける
13. マテリアルの陰影処理を外す
14. AO とテクスチャをバイクして、マテリアルから陰影処理を外す
15. スポットライトのクリップ距離を減らす
16. カメラのクリップ距離を減らす
17. world AO を切る
18. Material SSS を切る
19. 使用する画像テクスチャを小さくする。256x256 の画像は 2k 画像が使うメモリの 1% しか使いませんが、最終レンダリングの画質が劣化しないことはよくあります
20. Subsurf を減らす。レベルを上げるたびに面数は前のレベルの4倍になります
21. Multires を減らす
22. 建物などの背景オブジェクトの matte render を行ない、シーン内にはオブジェクトそのものの代わりに、その画像を載せたビルボード (billboard) を置きます。これで頂点や面の数が減ります
23. オブジェクトからリンクしたインスタンスがたくさんあるときは、インスタンス化されている DupliFace を使います。100あっても、Blender が保持するジオメトリは1つだけです (インスタンスそのものが少しメモリを使います)

Render Settings

(レンダリングの設定)

- [Output\(出力\)パネル](#)
 1. Edge レンダリングの無効化
 2. Save Buffers (バッファ保存)
 3. ポップアップではなく [Image Editor](#) ウィンドウにレンダリングします。 [Render Window](#)
 4. マルチコア CPU での複数 スレッド の使用 (複数の 部品 を使います)
- [Render Layers\(レンダーレイヤー\)パネル \(eng\)](#)
 1. 目的のレイヤーだけをレンダリング
 2. すべての光源を単純なスポット1つにしてレンダリング (Light: 欄にスポットの名前を入力)
 3. 1つのマテリアルで上書きしてレンダリング (Mat: 欄に名前を入力)
 4. Z などの必要でないレンダーパスの無効化。もしくは Diffuse などの目的のパスだけをレンダリング
- [Render\(レンダー\)パネル](#)
 1. Shadows を切る
 2. Environment Mapping を切る
 3. Panoramic Rendering を切る
 4. Raytracing を切る
 5. SSS Subsurface Scattering を切る
 6. oversampling/aliasing OSA を切るか下げる
 7. Motion Blur を切るか下げる
 8. 部分別にレンダリングする。これで非力な PC でも巨大な画像をレンダリングできます。マルチコア PC では同様に各部分にスレッドを割り当てます
 9. octree resolution を増やす
 10. 最終的な解像度の縮小サイズでレンダリング (25% など)
 11. Fields レンダリングを切る
 12. Border レンダリングで全体画像の一部をレンダリング
- [Anim\(アニメーション\)パネル](#)
 1. アニメーションのフレーム数を減らす (そして、同じ長さのアニメーションで低いフレームレートを使う)。たとえば3秒のアニメーションで、75枚のフレームを毎秒25フレームでレンダリングする代わりに、30枚のフレームを毎秒10フレームでレンダリングします
- [Bake\(バイク\)パネル](#)
 1. Full Render をバイクする。マテリアルをもとにオブジェクトに着色する UVテクスチャを作って、陰影なしでマテリアルの代わりに使います
 2. Ambient Occlusion だけをバイク
 3. オブジェクト用のテクスチャをバイク
 4. Normals (法線) や Displacement (ディスプレイスメント) のバイクはレンダリング時間を高速化せず、別の目的に使われます
- [Format\(ファイル形式\)パネル](#)
 1. 低解像度でレンダリング。絵が小さいほどレンダリング時間は少なくなります
 2. 高速な CODEC か CODEC 設定を選ぶ
 3. 白黒でレンダリング (BW ボタン)
 4. FFmpeg を使っていれば、Multiplex audio を有効にしない
 5. FFmpeg を使っていれば、Autosplit Output (出力を自動分割する) (Video パネルボタン)
 6. 色だけが必要なら RGB だけをレンダリング。A チャンネル (RGBA ボタン) はメモリ使用量を増やしますが、動画ファイルの保存時には使われません

Multi-Pass Compositing

(マルチパスのコンポジット)

レンダリングの長期化問題に取り組むのに使えるもう一つの戦術は、「ポストプロダクション」の章で解説されている [コンポジット](#) を積極的に利用できるように、作業工程を根底から組み立てることです。この方法では、各場面を、別々にレンダリングできる構成要素にまで分解し、別々にレンダリングされた要素を統合して、完成した動画を作り上げます。以下はその例です:

- カメラが動かないのであれば、背景も動きません。必要なのはひとつのフレームだけです (同じことはフレーム内にある動かないものすべてに当てはまります)。こうした要素は一度生成されれば、必要とされる数のフレームをまたいで、必要な回数だけ再利用できます
- 影とハイライトは、照明の当てられたオブジェクトや影で覆われたオブジェクトと別々に取得できます。intensity (強度)、色、効き目の強さのようなものはあとで、再レンダリングせずに調整できます
- 影を作らない光源を使うことから始めてください (影の計算は大きく時間を削ぎます)。次に「影のみ」の光源を使って (影を作りますが光を差しません)、実際に必要に思えた箇所 だけ影を作ります (ほとんどの場合実際に問題になるのはシーン内に存在する影のいくつかだけで、残りは単に気づかれませんが)
- オブジェクトを別々に扱うことで、扱いにくいライティング条件を避けることができ、別々にレンダリングされたクリップを統合して、結果を微調整でき

ます。

これは非常に一般的な考え方です。たとえば現代的な楽曲収録では常に「マルチトラック」方式を使います。曲のパートをそれぞれ別に隔離した状態で録音し、その後「ミックス」します。「最終ミックス」がさらなる処理工程であるマスタリングを経て、完成した製品が生まれます（実際に、現代的な音声処理ソフトウェアは Blender のノードベースのコンポジットと直接比較できるものです）。

この方法には圧倒的な長所があります：

- 選ぶことができます。何か「少しおかしい」とき、必ずしも一からやり直す必要はありません。
- 実際、締め切りを破るのは「ある場面のちょっとしたことが間違っている」ために、普通は完成しているはずのものを（そっくりそのまま）再レンダリングすることです。これを避けるためにコンポジティングは役立ちます。というのは（理想的には）間違いの見つかった特定箇所だけを再実行しなければならないからです（もしくは、間違った部分を「不要データマスク (garbage matte)」を使って除外して、直したものをその場所に差し込むことがおそらくできます。誰も気づかないでしょう！）。
- 「わかったよ、こいつはほぼ僕が欲しかったものだ。でも今度はこれを 足してあれを削りたいかな」となるかもしれません。コンポジティングベースのやり方でちょうどそういったことができます。さらには 非破壊でできます。言い換えれば、「追加分」（あるいは「マスク」）を別チャンネルの情報として生成することで、全体の「ミックス」内でその影響を微調整できるようになり、あるいは気分が変わってそれを完全に削除することも、元に戻さない変更をすることなく行えるということです。
- 概ね、この段階ではそれまでに「R、G、B、アルファ（透明度）と Z 深度情報」でできたラスタ画像」を操作することで 2 次元的に働くので、一貫して高速です。
- 分離された各レンダリングタスクは単純化されているため、コンピュータはずっと少ないリソースを使って実行できます。
- タスクは複数のコンピュータで分担することができます。処分できずに押し入れに居座っている二世前前のマシンのような、非力な PC でさえも参加できます。
- 「結局のところ、シーンが 説得力を持つために、物理的に正確である必要はありません」。コンポジティングベースのやり方はこれを最大限に利用します。本当に目立った違いを生むような、シーンのはっきりした特徴 (aspect) に焦点を絞ることができます。重要性の低い特徴 (aspect) を（「技術的には正しい」が）意識して選び除外することで、少なからぬ時間を節約することができますし、おそらくそれは気づかれないうでしょう。

もちろん、このやり方はそれ自身の代償なしには存在しません。所持しているマテリアルとその場所、更新の必要性、それにきっちり同じものを作る方法、といったことを正確に追いつけるための、扱いやすいアセット管理システムを考案しなければなりません。綿密に編成されたファイルコレクションにあるオブジェクト、ノード、マテリアル、テクスチャ、シーンを参照するためには、Blender の「ライブラリリンク」機能を理解して使う必要があります。完成した撮影の構成がどうあるべきか、またタスクの内訳はどうあるべきか、あらかじめ明確な見解を持たねばなりません。几帳面に記録を行い、記録を管理しなければなりません。しかししっかりとした製品を完成させるために、これはときに、唯一ではなくとも最良の方法になります。

Distributed Render Farm

(分散型レンダーファーム)

多くの頭脳を結集させれば、レンダリング時間全体を減らすことができます。CPU の割り当てにはいくつかレベルがあります。

Multi-Threading

(マルチスレッド)

レベル 1。マルチコアの CPU があれば、スレッド数を増やすと Blender は その数の CPU をレンダリングの計算に使用します。

Frame Ranges

(フレーム範囲わけ)

レベル 2。LAN に繋がった 別の PC が利用できれば、作業をフレームごとに分割することができます。例えば 200フレームのアニメーションをレンダリングするなら、同程度の性能の PC が5台あれば、1台目にフレーム 1-40、2台目に 41-80、といった具合に分担します。他より遅い PC には、単に少なめにフレームを割り当ててください。LAN レンダリングを行うには .blend ファイルのあるフォルダーを共有ドライブに設定します(テクスチャなどの外部ファイルは Pack (パック/梱包)しておく必要があります)。各 PC で Blender を起動し、この .blend ファイルを開きます。保存は行わずに、開始と終了フレームを設定し、レンダリングを開始します。出力パスに相対パスを利用すれば、レンダリングしたフレームはホスト PC に保存されるでしょう。

Collaborative Rendering

(共同レンダリング)

レベル 3。WAN レンダリングを行うことができます。.blend ファイル(Pack した状態で!)をインターネットを介してメールかファイル共有か Verse-Share して、レンダリングの一部を誰かの PC に行ってもらいます。作業が完了すれば、できたフレームをメールで送ってもらえるでしょう。信頼できる友人がいれば、この方法で協力することができます。

Remote Renderfarms

(遠隔のレンダーファーム)

レベル 4。レンダーファームのサービスを利用できます。[BURP](#) (Big and Ugly Rendering Project) のようなレンダーファームサービスは、組織運営されています。.blend ファイルをメールで送ると、レンダリングは複数台の PC に割り振られます。BURP を例に出したのはこのサービスが無料で、Blender ユーザーである会員の PC を使った BOINC 型のバックグラウンド処理で使用するためです。他のサービスは定期契約型課金や、従量型の課金を行っています。

解説

Blender 内部からのネットワークレンダリング

目的

- 透明性
- 柔軟性

使い方

バージョン 2.6 以降では、ユーザー設定の Addons パネルで「Network Render」を有効化しておく必要があります。

GUI

マスター/Master

一台のマシンでマスターサーバーを開始します。

- Blender を起動し、information ウィンドウのヘッダーにあるドロップダウン(「Scene」の隣)を使ってレンダリングエンジンを Network Render に切り替えます。
- (レンダリングモードを選択済みであることを確認してください) 処理モードとして Master を選びます。
- 「オプションとして」待ち受ける IP address を指定します。port も同様です。サーバーを指定ポートのすべてのネットワークインターフェイスで待ち受けるときは、[default] のままにします。
- Start を押します(空のレンダリングウィンドウが開きます)。レンダリング状況を示すバーにはサーバーの動作が反映されます。
- マスターは、通常のレンダリングのキャンセルのように Esc を押すと停止します。それまでは実行を続けます。

マスターのウェブインターフェイス

開始後マスターは、スレーブとジョブに関するさらなる情報を与えるウェブインターフェイスも提供します。現在、二つのウェブインターフェイスがあります。古いものは次の URL スキームで見られます。http[s]://マスターのIPアドレス:マスターのポート
開発中の jQuery ベースの新しいものは次の URL スキームで見られます。http[s]://マスターのIPアドレス:マスターのポート/html/newui
新インターフェイスに関する全情報は [こちら](#) をご覧ください。

スレーブ/Slave

別のマシンでレンダースレーブを開始します。

- Blender を起動し、レンダリングエンジンを Network Render にします。
- (レンダリングモードを選択済みであることを確認してください) 処理モードとして Slave を選びます。
- (オプションで) マスターサーバーの IP address と port を指定します。ブロードキャストからスレーブにマスターを自動検知させるには、[default] のままにします。
- Start を押します(空のレンダリングウィンドウが開きます)。レンダリング状況を示すバーにはスレーブの動作が反映されます。
- スレーブは、通常のレンダリングのキャンセルのように Esc を押すと停止します。それまでは実行を続けます。

クライアント/Client

あなたのワークステーションからクラスタにジョブを送るには:

- レンダリングする blend ファイルを開きます。サイズ等のレンダリング設定を確認します。
- ファイルを保存します(ここで最後に保存したファイルが送信されます)。
- レンダーエンジンに Network Render を選びます。
- 処理モードとして Client を選びます。
- 次のいずれかを行います:
 - マスターサーバーの IP address および port を指定します。
 - ブロードキャストからクライアントにマスターサーバーを自動検知させるには、address の下にある Refresh ボタンを押します。
- アニメーションジョブをマスターサーバーに発信するには、Send Job を押します。
- 準備ができれば、アニメーションをレンダリングして(Ctrl-F12)、できたフレームを集めます。できたフレームは自動的に「現れ」ますが、進行中のフレームでは少し止まります。
- アニメーションの任意フレームで Render を押すこともできます。マスターから結果が取得されます。
- 簡単に言えば、「ネットワークでアニメーション」を押し、送られてくるフレームを待つだけでいいのです。レンダリング時間の合計はほぼスレーブの数に反比例します(転送時間を除く)。

ひとつのシステム上で、マスター、クライアント、スレーブを実行できます。

コマンドライン

- 前述のとおり、マスターを設定します。「Start Service」を押す代わりに、ファイルを保存します(例: master.blend)。
- スレーブの設定も同様に保存します(例: slave.blend)。
- 以下のようにしてマスターとスレーブを開始し、バックグラウンドレンダリングを行います:
 - `blender -b master.blend --addons netrender -a -noaudio -nojoystick`
 - `blender -b slave.blend --addons netrender -a -noaudio -nojoystick`
- マスターとスレーブは Ctrl-C で停止できます。スレーブをマスターより先に止めることをお勧めします。

オプション `--addons netrender` は、ネットワークレンダーのアドオンが読み込まれていることを保証します。オプション `-noaudio -nojoystick` は任意です。警告表示を避けるためだけに指定しています。

その他

最終結果がローカルのレンダリングとまったく同じになるように、完全なマルチレイヤーのレンダリング結果が使われます。もとのファイル内でこれを出力として指定する必要はありません。スレーブ上で自動的に行われます。

テスターは [IRC \(#blendercoders\)](#) が電子メールで **theeth** と連絡をとるよう呼びがかかっています。

設定



NetRender as a Render Engine

レンダーエンジンのドロップダウンは、Blender ウィンドウの一番上にある Info ウィンドウにあります。ネットレンダー機能を使うには、ここで Network Render を選びます。

マスター/Master



マスターの設定

- Network Settings (ネットワークの設定)
 - **Start Service:** マスターサーバーを開始します。
 - **Path:** マスターがジョブファイル、処理結果、ログやその他のものを保存する場所。 *master_<pid>* という名前で新規ディレクトリを作ります。 *<pid>* にはマスターサーバーの process ID が入ります。あとでマスターを再開できるように、フォルダのルートに "blender_master.data" という名前のファイルが保存されます。
 - **Server Address:** マスターが待ち受けるネットワークインターフェイスのアドレスです。 *[default]* は利用できるネットワークインターフェイスをすべて待ち受けます。
 - **Port:** マスターの待ち受けポート。
 - **SSL:** スレーブやクライアントとの接続に SSL (https) を使います。有効にすると、SSL の証明書と鍵を指定する二つの項目が新たに表示されます。自己署名証明書か、comodo や verisign のようなサードパーティ提供の証明書を使えます。トラストチェーン (chain of trust) があれば同じファイル内に入れることができますが、証明書は先に入れる必要があります。証明書、トラストチェーン、鍵は PEM ファイルとして渡さなければなりません。
 - **Open Master Monitor:** ブラウザでウェブベースのマスターモニターを開きます。
- Master Settings (マスターの設定)
 - **Broadcast:** マスターのアドレスとポートをローカルネットワークでブロードキャストします (10秒ごと)。
 - **Force Dependency Upload:** ローカルにある既存の依存ファイルはクライアントのファイルに一致しても使わず、代わりにクライアントの依存ファイルをマスターにアップロードさせます。
 - **Clear on exit:** マスターが停止したとき、 *Path* 内に作られたディレクトリを取り除きます。有効にすると、何らかの理由でマスターの処理が止められたとき、あとでマスターが再開しないようにします。

スレーブ/Slave



スレーブの設定

- Network Settings (ネットワークの設定)
 - **Start Service:** スレーブノードを開始します。
 - **Path:** スレーブがジョブファイル、処理結果、ログを保存する場所。 *slave_<id>* という名前で新規ディレクトリを作ります。 *<id>* にはマスターサーバーによって割り当てられた Slave ID が入ります。
 - **Server Address:** マスターの待ち受けアドレス。
 - **Port:** マスターの待ち受けポート。

- **Refresh:** マスターのブロードキャストを受信し、アドレスとポートを決めます(最大で20秒かかります)。
 - **Open Master Monitor:** ブラウザでウェブベースのマスターモニターを開きます。マスターのアドレスが正しいとき有効になります。
- Slave Settings(スレーブの設定)
 - **Tags:** スレーブに割り当てられたタグの、セミコロン区切りの一覧です。ジョブの持つタグをスレーブがすべて持たない限りは、そのスレーブにジョブは割り当てられません。
 - **Clear on exit:** スレーブが止められたとき、*Path* に作られたディレクトリを取り除きます。
 - **Generate thumbnails:** スレーブ上のレンダリング結果のサムネイルを作ります(無効にするとマスターの要求があり次第作られます)。
 - **Output render log on console:** レンダリングのサブプロセスからのログを、マスターに送られるレンダーログだけでなく、標準出力にも出力します。
 - **Threads:** スレーブがレンダリングに使うスレッド数。

クライアント/Client



クライアントの設定



スレーブとジョブの一覧

- Network Settings(ネットワークの設定)
 - **Path:** クライアントが一時的なレンダリングの結果ファイルを保存する場所です。
 - **Server Address:** マスターの待ち受けアドレス
 - **Port:** マスターの待ち受けポート
 - **SSL:** マスターとのやりとりに SSL (https) を使います。
 - **Refresh:** マスターのブロードキャストを受信し、マスターのアドレスとポートを決めます(最大20秒かかります)。
 - **Open Master Monitor:** ブラウザでウェブベースのマスターモニターを開きます。マスターのアドレスが正しいとき有効になります。
- Job Settings(ジョブの設定)
 - **Animation on network:** 現在開いているファイルをジョブとしてマスターに送り、結果を待ちます(レンダリングが他の場所で行われること以外は、通常のアニメーションレンダリングのように動作します)
 - **Send job:** 現在開いているファイルをジョブとしてマスターに送ります。返された job ID が *current job ID* になります。
 - **Bake on network:** シーン内のポイントキャッシュやパーティクルシステムを使ってすべてのモディファイアとともにバイク処理のジョブを送ります。
 - **Send current frame job:** 現在開いているファイルを、現在のフレームだけがレンダリングされるようにして、ジョブとしてマスターに送ります。返された job ID が *current job ID* になります。
 - **Name:** ジョブの名前。[*default*] は blend ファイル名を使います。
 - **Category:** ジョブのカテゴリで、オプションです。マスター上のジョブもカテゴリによって負荷分散されます。
 - **Tags:** スレーブに割り当てられたタグの、セミコロン区切りの一覧です。ジョブの持つタグをスレーブがすべて持たない限りは、そのスレーブにジョブは割り当てられません。
 - **Engine:** このジョブのレンダリングに使うレンダリングエンジン。
 - **Priority:** ジョブの優先度。優先度は、マスターがジョブを X 個のジョブであるかのように数えさせるための乗数です(たとえば優先度1と優先度2のジョブの釣り合いをとると、それぞれ、作業量の33%と66%を受け取ります)。
 - **Chunks:** ジョブチャンクの一部としてスレーブに割り振られるフレーム数。
 - **Save Before Job:** ジョブとして割り振られる前に、現在のファイルをディスクに強制的に保存します。
- Slaves Status(スレーブの状態)
 - **List:** マスターに接続している全スレーブの一覧。
 - **Refresh:** マスターからのスレーブ情報を最新化します。
 - **Remove:** 選択したスレーブをブラックリストに移動します。
- Slaves Blacklist(スレーブのブラックリスト)

- **List:** ブラックリストに登録した全スレーブの一覧。
- **Remove:** 選択したスレーブをブラックリストから取り除きます。
- Jobs (ジョブ)
 - **List:** マスター上の全ジョブの一覧。
 - **Refresh:** マスターからのジョブ情報を最新化します。
 - **Remove:** マスターからジョブを取り除きます。
 - **Remove All:** マスターから全ジョブを取り除きます。
 - **Get Results:** 選択中のジョブから利用できる全フレームを取得します。結果は マルチレイヤー EXR 情報として現在の出力先ディレクトリにダウンロードされます。

物理演算のバイク処理

物理演算のバイク(bake)は最近 Netrender に追加された機能です。シーン(またはモディファイア、あるいはパーティクルシステム)で使われている各ポイントキャッシュについて、バイクジョブの配信をサポートします。

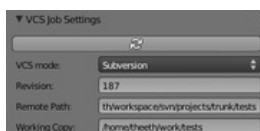
ポイントキャッシュはそれぞれ、スレーブ上で独立してバイクされます。バイク順や依存度は現在サポートされていません。

結果はウェブインターフェイスの該当ジョブのページから、zip ファイルとしてのダウンロードだけができます。ダウンロードができれば unzip し、ファイルに関連付けられた blendcache フォルダに結果を置き、バイクしたモディファイアとパーティクルシステム用にディスクキャッシュを有効にします(この手順はある時点で自動で行われているはず)。

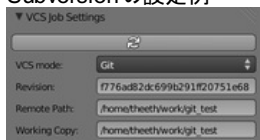
ポイントキャッシュのバイク時に出力されるテキストは、ログファイルにパイプするには恐ろしく向かず、それほど有益なものではありません。したがってジョブのログファイルから多くの情報は得られないでしょう。この変更にはバイクコードを直接変える必要があります。

他の種類の物理演算(fluid 等)のバイクは、いずれサポートされるでしょう。

バージョン管理ジョブ



Subversion の設定例



Git の設定例

VCS (バージョン管理システム)をジョブタイプとして使うことで、ネットレンダラーが通常使用する依存システムを無視し、代わりにバージョンづけシステムに依存できます。より組織化された製作では、依存エラーやディスク消費量、ジョブの割り振り時間を最小限にすることから、VCS を使うのは通常は良い考えです。

現時点で、サポートされているバージョン管理システムは Subversion (svn) と Git の二つだけです。新規追加は比較的容易で、要望があれば行います。

VCS を一つ選んだら、システム固有の設定を3つ指定する必要があります:

- **Revision:** 特定バージョンの認識に使われる文字列 (svn: revision, git: commit hash)。
- **Remote path:** ファイルをダウンロードできるリモートパス (svn: server url, git: スレーブがチェックアウト可能な remote repository path)。ジョブファイルはすべて、このフォルダかサブフォルダになければなりません。
- **Working copy:** 作業コピーのルートフォルダ。ここにリモートファイルがダウンロードされます。同じファイルを何度もダウンロードしないように、ジョブ間で維持されます。ジョブがバージョン管理システムから、特定ファイルの新たなリビジョンを必要とするときにだけ変更されます。

Refresh ボタンは知り得る範囲で最善の設定を探します。

覚書および既知の不具合

- ノード間に共有ネットワーク空間を必要としません。
- さまざまなファイルを割り振ることができ、結果はすべて独立して取得できます(ファイルを閉じてあとで取り出したい場合は、割り振り後にファイルを保存してください)。
- ネットワークエラーの管理はほとんどないため、マスターを最初に閉じると壊れます。不正なアドレスを入力した場合も同じです。
- 同一ファイル名による多くの依存の問題: https://projects.blender.org/tracker/index.php?func=detail&aid=25783&group_id=9&atid=498

はい、現在のワークフローが理想から遠いことは わかっていません、本格的な render farm の視点からは特に。Matt が私を鞭打ってもっとよいものを提案してくれることを期待しています。ローカルレンダーのように「ネットワーク上でアニメ」を押すだけで自動的にネットワークに割り振って結果を待つようにしたいです。すべての「本格的な」機能は随意選択式にすべきです。

負荷の分散/Load Balancing

基本的な負荷分散は、10秒ごとに各ジョブのクラスタの平均使用量を計算することで行われます。次に割り振られるジョブは使用量の一番低いもの(使っているスレーブ数が最小のもの)です。ジョブの優先度は約数の働きをします。優先度2のジョブは、ジョブが1つだけではなく2つあるかのように、クラスタの一定の割合を使います(たとえば優先度1のジョブとスレーブを共有する優先度2のジョブはそれぞれ、処理能力の33%と66%を使うでしょう)。さらに、例外と最優先ルールがあります:

例外

- ひとつのジョブは、それが唯一のジョブでない限りは、全体の N% より多いスレーブを使えません。これで、遅いジョブが速いジョブを停止させないようにします。現在は 75% にセットされていますが、カスタマイズ可能にすべきです。

最優先(基準)

- 割り振られたフレーム数が N より小さいもの(新しいジョブ優先)。エラーを早期に把握するためのものです。
- 最後の割り振りから N 分より長く経過したもの。優先度の高いジョブが他を停止させないようにします。

To do

- メモリからジョブを送る
- 可視性についてレンダーエンジンの選択に依存しない
- "熟練の" レンダー管理
- もっとうまく定義された通信プロトコル
- ノード上でのシミュレーション計算 (cloth、smoke、...)を行なって、その後割り振ってレンダリングするためにポイントキャッシュをサーバーに送るオプション
- アップロード時にテクスチャをパックする
- 単一フレームをタイルとして割り振る

技術的詳細

期限切れです。コードを読んでここに情報を載せてください。

機能一覧

- ファイルに代えてパスをサポート
- client-server-slave: ジョブを特定ノードに制限
- client-server-slave: ノードマシンの状態を表示
- client-server-slave: エラーログのマネージャーへの報告 (ノードからのすべての `stdout` と `stderr`)
- ジョブのキャンセル
- エラーフレームの再開始
- Windows でクラッシュレポートの無効化
- 1つ以上のフレームを一度に割り振り(フレームのシーケンス)
- リセット後、フレームでエラーのあるスレーブをブラックリスト登録
- Multiple paths on job announce
- 全ファイルを捕捉するまでジョブを遅らせる
- フレーム範囲の制限(フレーム範囲で必要とされるときだけポイントキャッシュファイルを送る)
- マスターにログの一部を送信
- TODO: スレーブにネットワークパス上の結果をコピーさせる
- TODO: client-master: ジョブをアーカイブに入れる(ソースファイルと結果の複製)
- TODO: master-slave: スレーブの性能を基準にしたジョブ制限

API 機能のウィッシュリスト

以下はネットレンダーの改善に必要に思える blender コードの一覧です。いくつかはバグで、いくつかはいつか搭載すべき(してほしい)機能です。

- API access to jobs, to be able to run masters and slaves in the background as well as render job notifiers on the client.
- Render result from multilayer image in memory
- Render and load tiles in render results

- [YafRay \(eng\)](#)
- [LuxRender \(eng\)](#)
- [Cycles \(eng\)](#)
- [Freestyle \(eng\)](#)

Cycles レンダーエンジン

Cycles は、Blender 2.61 以上で利用可能な新しいレンダーエンジンです。まだ開発中で、従来の多くの機能に対応しつつ、インタラクティブ性と使いやすさを主としたレンダーエンジンにすることを目的としています。[開発者ドキュメント\(英文\)](#)もあります。

使い方

Cycles はデフォルトで利用可能なアドオンの一つとして同梱されています。Cycles を使用するには、トップヘッダ(Infoヘッダ)で、アクティブレンダーエンジンとしてセットしなければなりません。一旦設定した後は、3Dビューエディタで、描画モードを Rendered に設定すれば、インタラクティブレンダリングが始まります。このレンダーはマテリアルやオブジェクトの編集が行われれば更新し続けます。

[チュートリアル\(英文\)](#)

Reference

- [カメラ](#)
- [マテリアル](#)
 - [サーフェス](#)
 - [ボリューム](#)
 - [ディスプレイメント](#)
- [ワールド](#)
- [ランプ](#)
- [ノード](#)
 - [シェーダ](#)
 - [テクスチャ](#)
 - [その他](#)
 - [OSL シェーダ](#)
- [ニアレンダリング](#)
- [ライトパス](#)
- [インテグレータ](#)
- [ノイズ低減](#)
- [レンダーパス](#)
- [テクスチャ編集](#)
- [GPU レンダリング](#)

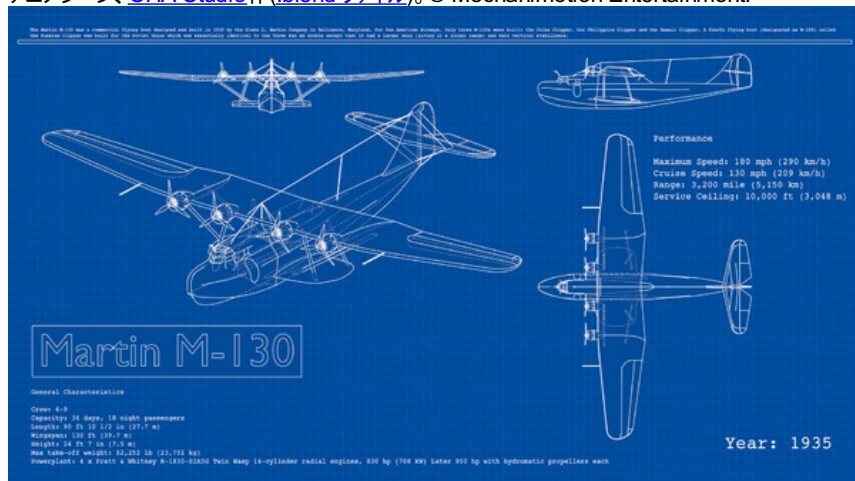
FreeStyle とは？

Freestyle はエッジ(境界)とラインベースのノンフォトリリスティック(NPR)レンダリングエンジンです。メッシュデータと Z 深度情報を頼りに、選択されたエッジタイプ上に線を描画します。様々なラインスタイルを追加することで、アーティストック風(「手描き」や「ペイント風」など)またはテクニカル風(きっちりした線)の表現を作成できます。

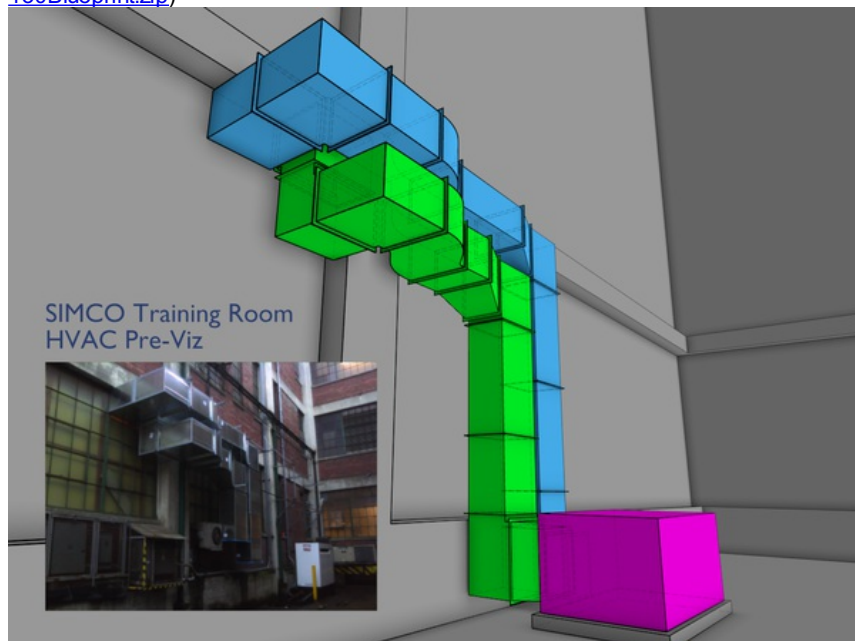
[Python スクリプティング\(英文\)](#)と[パラメーター編集モード\(Parameter Editor\)](#)の二つの操作モードがあり、パワフルで多様なラインスタイルと成果を可能にします。日本の毛筆(big brush)やアニメ風(cartoon)、製図(blueprint)、深度による幅(thickness-with-depth)のようなラインスタイルが、Python によりあらかじめスクリプト化されています。パラメーターエディターモードは破線などの機能を直観的に編集でき、複数のラインタイプとエッジ定義を簡単に設定できます。その上さらにラインスタイルモディファイアが導入されており、無限の可能性がります！



アニメーション、OHA Studio作([blend ファイル](#))。© Mechanimotion Entertainment.



1935年の Martin M-130 の設計図レンダリング、LightBWK氏作。CC0。注意：非常に重いです！ Blender のストレステスト用に作成されたため、クラッシュする可能性があります。(File:[M-130Blueprint.zip](#))



HVAC Pre-Viz, Lee Posey氏作。CC0 ([File:HVACPreViz.zip](#))

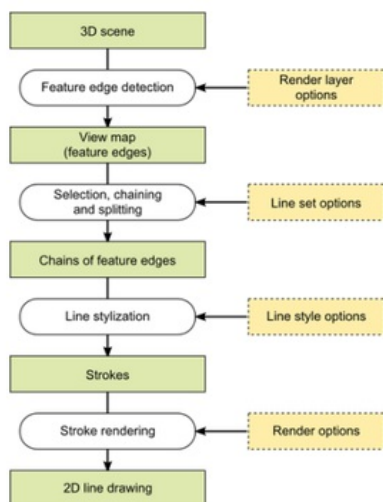


Kitchen, Vicente Carro氏作。© AnigoAnimation

他にも多くの作品があります：http://wiki.blender.org/index.php/Dev:Ref/Release_Notes/2.67/FreeStyle#Freestyle_Artwork_Showcase

ワークフロー

1. FreeStyle をプロパティ(Properties)エディタ → レンダー(Render) タブ → FreeStyle パネルのチェックボックスで有効にします。FreeStyle は Blender 内部レンダラでのみ利用可能であることに注意してください。
2. Freestyle の設定は、新しい レンダーレイヤー(Render Layers) タブにあります。
3. 一つのレンダーレイヤーにつき、一つのビューマップ(Viewmap)を持つことができます。ビューマップはエッジ検知設定 (クリース角度(Crease Angle)、カリング(Culling)のON/OFF、面のスムーズさ(Face Smoothness)のON/OFF、マテリアル境界(Material Boundaries)のON/OFF、詳細オプションの球半径(Sphere Radius)とKr 派生イプシロン(Kr Derivative Epsilon)を有しています。
4. 一つのビューマップにつき複数のラインセット(Line Set)を持つことができます。
5. ラインセットは、シーンを元にしたラインの中から、どのエッジタイプ(Edge Type)と選択(Selection)がレンダリングされるかをコントロールします。
6. 各ラインセットは一つのラインスタイル(Line Style)を使用します (複数のラインセットで共有することもできます)。
7. ラインスタイルはリンクしているラインセットの色、アルファ、幅、その他のレンダリング方法を Freestyle に伝えます。



Freestyle のビューマップとプロセスのブロック図

既知の制限と問題

- FreeStyle は Blender内部レンダラでのみ利用可能です。
- 多くのメモリが必要です。レンダーレイヤー内のすべてのメッシュオブジェクトは一度に読み込まれるためです。
- 面のあるメッシュオブジェクトのみ対応しています。以下のようなメッシュは無視されます。
 - ワイヤ(Wire)マテリアルのメッシュ面。
 - 完全に透過するマテリアルのメッシュ面。
- 半透明の面は不透明な面として処理されます。
- まだ面の交差によるエッジは検知されません。
- レイヤーマスク(Layer mask)は Freestyle では動作しません。
- Freestyle のレンダリング結果には Z 深度情報がありません。
- パノラマ(Panorama)カメラでは動作しません。

Cycles レンダーエンジン

Cycles は、Blender 2.61 以上で利用可能な新しいレンダーエンジンです。まだ開発中で、従来の多くの機能に対応しつつ、インタラクティブ性と使いやすさを主としたレンダーエンジンにすることを目的としています。[開発者ドキュメント\(英文\)](#)もあります。

使い方

Cycles はデフォルトで利用可能なアドオンの一つとして同梱されています。Cycles を使用するには、トップヘッダ(Infoヘッダ)で、アクティブレンダーエンジンとしてセットしなければなりません。一旦設定した後は、3Dビューエディタで、描画モードを Rendered に設定すれば、インタラクティブレンダリングが始まります。このレンダーはマテリアルやオブジェクトの編集が行われれば更新し続けます。

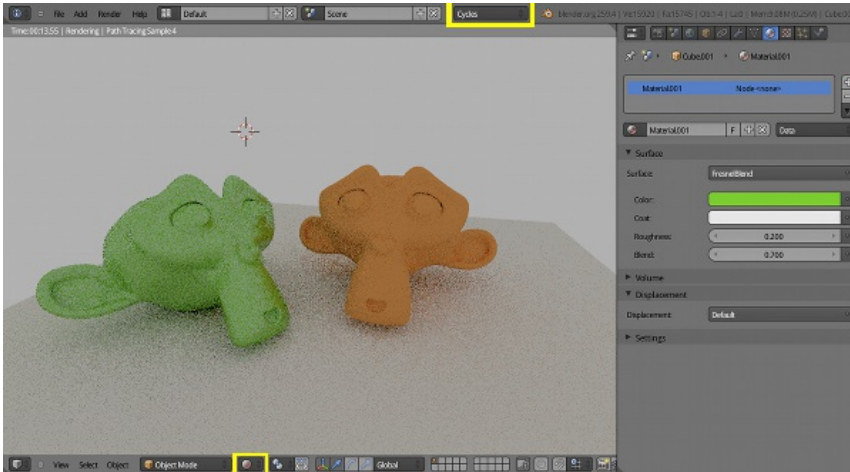
[チュートリアル\(英文\)](#)

Reference

- [カメラ](#)
- [マテリアル](#)
 - [サーフェス](#)
 - [ボリューム](#)
 - [ディスプレイメント](#)
- [ワールド](#)
- [ランプ](#)
- [ノード](#)
 - [シェーダ](#)
 - [テクスチャ](#)
 - [その他](#)
 - [OSL シェーダ](#)
- [ニアレンダリング](#)
- [ライトパス](#)
- [インテグレータ](#)
- [ノイズ低減](#)
- [レンダーパス](#)
- [テクスチャ編集](#)
- [GPU レンダリング](#)

使い方

Cycles はデフォルトで利用可能なアドオンの一つとして同梱されています。Cycles を使用するには、トップヘッダ (Infoヘッダ) で、アクティブレンダーエンジンとしてセットしなければなりません。一旦設定した後は、3Dビューエディタで、描画モードを Rendered に設定すれば、インタラクティブレンダリングが始まります。このレンダーはマテリアルやオブジェクトの編集が行われると更新し続けます。



皆さんの GPU がレンダリングに使用可能か、どれくらい使えるかを知りたい時は、[GPU レンダリング](#)をお読みください。

チュートリアル

- [Introduction to Cycles \(Blender Guru\)](#)



- [Introduction to the Cycles Render Engine \(Blender Cookie\)](#)



- [How to create hoar frost in Cycles \(BlenderDiplom\)](#)



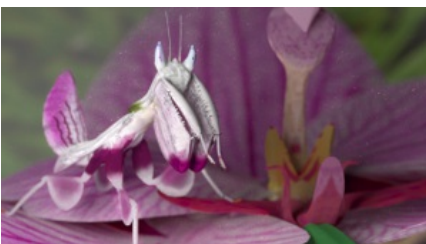
- [Create a Realistic Water Simulation \(Blender Guru\)](#)



- [Cycles Lighting, Materials and Textures Tips and Tricks \(BlenderDiplom\)](#)



- [Advanced Organic Material Setup with Maps \(BlenderDiplom\)](#)



- [Rendering a Scene of Wooden Barrels \(Blender Cookie\)](#)



- [Rendering Nuts and Bolts with Cycles \(Free 3D Tutorials\)](#)



- [Create a Realistic City \(Blender Cycles\)](#)

[300px](#)

- [Make a Procedural Starfield \(Blender Cycles\)](#)

[300px](#)

- [Cycles Material Nodes](#)
- [Cycles Tutorial Series](#)
- [Cycles Complete Overview](#)

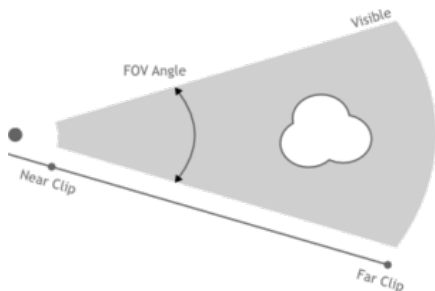
カメラ

カメラはパノラマレンダリングに未対応なことを除き、従来の Blender のカメラと完全に互換性があります。

パースペクティブ (Perspective)

Lens Size and Angle

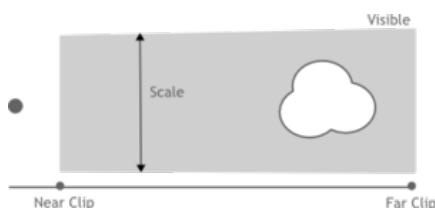
視野角度を操作します。



正射投影 (Orthographic)

Scale

画像に投影されるオブジェクトの大きさを操作します。



被写界深度 (Depth of Field)

Aperture Size

光が入射する、カメラの開口サイズ(レンズ半径と呼ばれることもあります)。0の場合、すべてのオブジェクトに焦点が合い、大きい値でオブジェクトが焦点距離から外れ、ピントが合なくなります。現実のカメラのアパーチャーサイズは、恐らく焦点距離と絞り値という言葉で記されており、その関係は次のとおりです：

$$\text{アパーチャーサイズ} = \text{焦点距離} \div (2 \times \text{絞り値})$$

Aperture Blades

もしこの設定が3以上なら、多角形のアパーチャーが円の代わりに使用され、レンダリング画像のぼやけたハイライトの形状に影響します。

Aperture Rotation

Aperture Blades の回転角度。

Focal Distance

オブジェクトに完全に焦点が合う距離。使用されるカメラからの距離として、代わりにオブジェクトを指定することもできます。

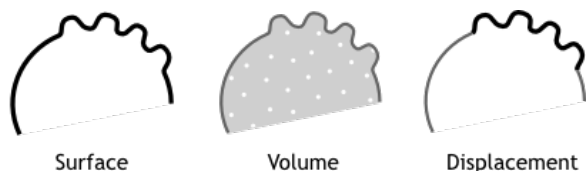
Clipping

Clip Start and End

オブジェクトを直接見ることのできる範囲。この範囲外のオブジェクトもすべて画像に間接的に影響し、遠くの光のバウンスもクリッピングされません。OpenGL レンダリングではクリッピング距離の上下限値を設定することが、十分な精度のラスターライゼーションを行うために重要でした。レイティングではこの事であまり苦労することはなく、もっと極端な値の設定でも安全です。

マテリアル

マテリアルはメッシュ、カーブ、および他のオブジェクトの見た目を定義するものです。マテリアルは3種類のシェーダから成り立っています。メッシュのサーフィス(surface 表面)の見た目、メッシュ内部のボリューム(volume かたまり)、メッシュ表面のディスプレイメント(displacement 置換)です。



サーフィスシェーダ

サーフィスシェーダはメッシュ表面での光の効果を定義するものです。一つまたはそれ以上のBSDF(双方向散乱分布関数)が、入射した光が反射するのか、屈折するのか、吸収されるのかを特定します。

Emissionはサーフィスから光が放射される度合いを規定します。どんなサーフィスでも光源とすることができます。

ボリュームシェーダ

ボリュームシェーダは現在独立に開発されています。

サーフィスシェーダが光を反射も吸収もしない場合、ボリュームの中に入ってくることになります。ボリュームシェーダが指定されていない場合は、光はそのままメッシュの反対側へ通り抜けます。

ボリュームシェーダが定義されている場合、このシェーダは光がメッシュの内部を進むに従ってどのような影響を受けるかということを記述しています。光は散乱したり、吸収されたり、ボリューム上のある点から放射されたりします。

マテリアルはサーフィスシェーダとボリュームシェーダを両方持つこともできるし、どちらか一方だけを持つこともできます。両方を定義することは、ガラス、水、氷などを表現するのに便利かもしれません。表面にガラスや光沢のあるシェーダを使い、入射した光のうちある程度は吸収されるようにするといでしょう。

ディスプレイメント

表面の形状と内部のボリュームはディスプレイメントシェーダによって置き換えることができます。それにより、テクスチャを使ってメッシュの表面に細部の表現を与えることができます。

設定によって、置換は仮想的なものとすることもできます。置換された場合の質感をサーフィスの法線に与えるこの手法は、バンプマッピングとして知られています。また、真の置換と仮想の置換を組み合わせることもできます。

エネルギー保存

マテリアルのシステムは物理ベースのレンダリングを想定して構築されており、そのマテリアルがどう見えるかということと、どのようなアルゴリズムでそれをレンダリングするかということははっきりと分けられています。これにより、リアルな結果とバランスの取れたライティングを得ることがより容易になっています。ただし、いくつか覚えておかなくてはならないことがあります。

グローバルイルミネーションを使うときにマテリアルが上手に働くためには、物理学用語で言うと、エネルギーが保存されなくてはなりません。つまりマテリアルは入射したよりも多くの光をはね返してはいけません。この性質は厳しく強制されるわけではありませんが、カラーが0.0から1.0の範囲にあって、かつBSDFがMix Shaderノードで合成されるだけの場合、自動的にそのようになります。

しかし、そのようにしないこともできます。それには1.0よりも高いカラー値を使うか、Add Shaderノードを使います。ただし、様々な照明のもとでのマテリアルの振る舞いを予測しやすくするためにこのようなことをする場合は、気をつけなくてはなりません。光が反射するたびに増幅され、エミッターのようになってしまう可能性があります。

ディスプレイメント

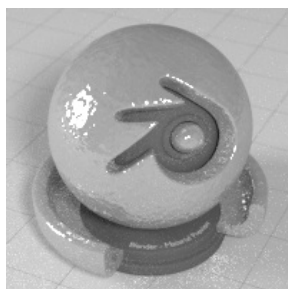
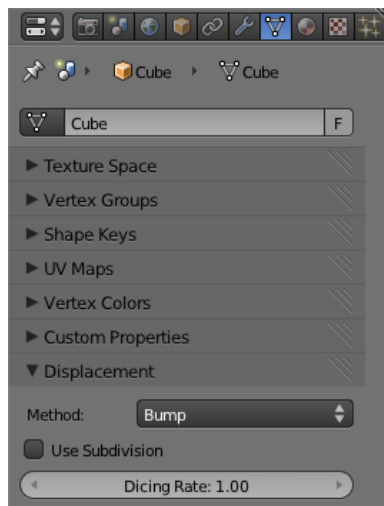
実装は完了していません。 [experimental feature](#)を参照。

表面の形状と内部のポリウムはディスプレイメントシェーダによって置き換えることができます。それにより、テクスチャを使ってメッシュの表面に細部の表現を与えることができます。

種類

設定によって、置換は仮想的なものとすることもできます。置換された場合の質感をサーフィスの法線に与えるこの手法は、バンプマッピングとして知られています。また、真の置換と仮想の置換を組み合わせることもできます。

- **True Displacement:** メッシュの頂点はレンダリングの前に置換されます。メッシュを実際に修正します。最も高い品質の結果が得られますが、メッシュが十分に再分割されている必要があります。結果として最もメモリ使用量が多くなります。
- **Bump Mapping:** サーフィスシェーダを計算する際に本当の法線ではなく修正した法線を使います。これは真のディスプレイメントの代わりとなる簡単な手法ですが、あくまで近似に過ぎません。物体のシルエットは不正確で、ディスプレイメントによってサーフィス自身に投影されるはずの影も現れません。
- **Displacement + Bump:** 二つの手法を組み合わせます。粗いメッシュを置換し、そこにバンプマッピングを施してディテールとします。



Bump Mapped
Displacement

再分割

実装は完了していません。 [experimental feature](#)を参照。

緻密なディスプレイメントをするにはメッシュが小さなポリゴンに再分割されている必要があります。これはモディファイアによっても行えますが、レンダーエンジンに再分割させることもできます。

サーフィス

サーフィスシェーダはメッシュ表面での光の効果を定義するものです。一つまたはそれ以上のBSDF(双方向散乱分布関数)が、入射した光が反射するのか、屈折するのか、吸収されるのかを特定します。

Emissionはサーフィスから光が放射される度合いを規定します。どんなサーフィスでも光源とすることができます。

用語

- **BSDF** : bidirectional scattering distribution function(双方向散乱分布関数)の略です。サーフィスにおいて光がどのように反射および屈折するかを定義します。
- **Reflection**(反射) : 光をサーフィスに対して入射してきたのと同じ側に跳ね返します。
- **Transmission**(透過) : 光をサーフィスの反対側へ通します。
- **Refraction**(屈折) : 光を透過する際にその方向を変化させます。

BSDFのパラメータ

物理ベースでないレンダラーとの主な違いは、光源からの直接光の反射と、他のオブジェクトからの間接光の反射の区別がなく、単一のBSDFで処理されるということです。これはやや可能性が制限されるということですが、全体的にいえば、少ないパラメータの調整で一貫性のある光景をレンダリングするのに有益であるとわれわれは信じています。

BSDFで光沢を表現するには、粗さ(**roughness**)パラメータによって反射の鋭さをコントロールします。0.0は完璧に鋭く、1.0では非常にソフトになります。hardnessやexponentパラメータと比べた場合の利点は、値の範囲が0.0から1.0までの範囲に収まっているために、より線形的なコントロールがしやすいということ、またテクスチャによる操作がしやすいということです。両者の関係はおおまかにいうと $roughness = 1 - 1/hardness$ です。

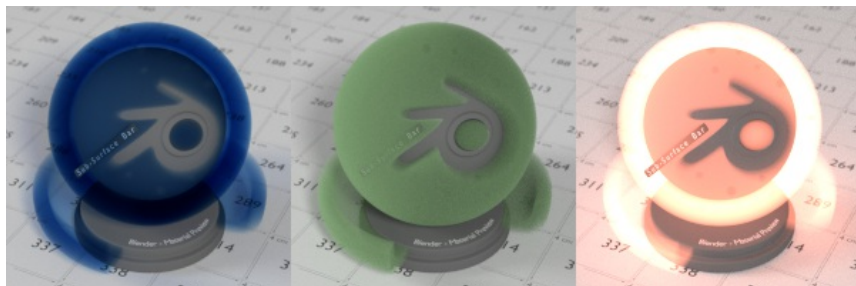
ボリューム

ボリュームシェーダは現在独立に開発されています。

サーフィスシェーダが光を反射も吸収もしない場合、ボリュームの中に入ってくるようになります。ボリュームシェーダが指定されていない場合は、光はそのままメッシュの反対側へ通り抜けます。

ボリュームシェーダが定義されている場合、このシェーダは光がメッシュの内部を進むに従ってどのような影響を受けるかということを記述しています。光は散乱したり、吸収されたり、ボリューム上のある点から放射されたりします。

マテリアルはサーフィスシェーダとボリュームシェーダを両方持つこともできるし、どちらか一方だけを持つこともできます。両方を定義することは、ガラス、水、氷などを表現するのに便利かもしれません。表面にガラスや光沢のあるシェーダを使い、入射した光のうちある程度は吸収されるようにするといでしょう。



The Volume Shader

テクスチャ編集

3Dビューの描画タイプ、UVマッピング、テクスチャペイントは、Cyclesを有効にしていると少し違った動作になります。UVマップにはイメージテクスチャが直接適用されず、イメージテクスチャノードの追加によって適用されます。

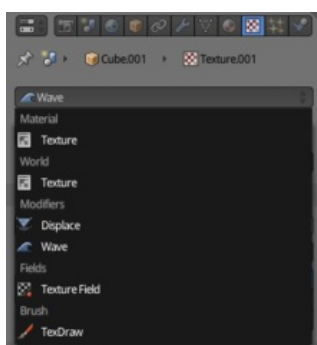
3Dビューの描画タイプ

Textureの描画タイプは、別の三つのタイプに置き換えられます。

- *Texture*: テクスチャの編集、ペイント、マッピングに適したモードです。照明はsolidと同様です。よってこれはBlender内臓のTextured solidモードに似ています。描画されるテクスチャはマテリアルのアクティブなイメージテクスチャノードです。
- *Material*: 単純化されたバージョンのすべてのマテリアル要素がGLSLシェーダを用いて描画されます。照明はレンダリング時に近く、これもテクスチャの編集、ペイント、マッピングに適していますが、さらにそれがマテリアルとして適用されたときの状態を見ることができます。
- *Rendered*: このモードでは、出力に選択されたエンジンに応じて3Dビューがレンダリングされます。二つのオフラインレンダラーとゲームエンジンを使え、ライティングも最終的な状態と同じです。



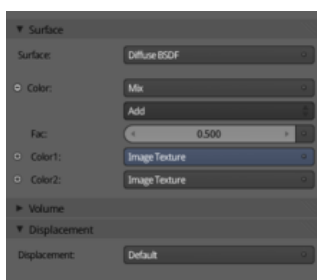
テクスチャのプロパティ



テクスチャのプロパティではテクスチャをリストから選べます。これはワールド、ランプ、マテリアル、モディファイア、ブラシ、物理フィールドのすべてのテクスチャノードが含まれます。

シェーディングノードに使えるのはCyclesテクスチャです。その他のノードにはBlenderテクスチャも使えますが、これは将来の開発で変更されます。

ペインティングとUV編集



テクスチャペイントモードで使われるのはアクティブなテクスチャノードです。これはノードエディタまたはテクスチャプロパティで選べます。またこのノードはマテリアルプロパティでは青くハイライトされます。

UVマッピングにはメッシュプロパティで指定されたアクティブなUVマップが使われます。イメージエディタでイメージを登録した場合も、アクティブなイメージテクスチャノードに影響します。

ワールド

ワールドの環境は単色または物理的な空モデルの照明を任意のテクスチャに放射できます。

サーフィスシェーダ

サーフィスシェーダは環境からシーンに対して放射される光を定義します。ワールドのサーフィスはシーンから非常に離れた距離があるかのようにレンダリングされ、環境とオブジェクトの間で双方向的な作用は起こりません。オブジェクトに対して一方的に光が照射されます。使うことができるシェーダは Background ノードだけで、color と strength の値がライトの強さを決めます。

ボリュームシェーダ

独立に開発中です

ヒント

直接的に目に見える背景と、照明効果として間接的に表れる背景にそれぞれ別のものを使うと便利ことがあります。それを簡単に行うには Mix ノードを追加し、Blend Factor を Is Camera Ray にします。1 つ目の入力カラーが間接的なカラーに、2 つ目が直接見えるカラーになります。

Background シェーダは現在のところ重点サンプリングされません (not importance sampled)。すなわち、背景の明るい領域により多くのサンプルを使うということをしません。そのため、たとえばいくつか明るい点があるテクスチャの場合、レンダリングの収束が遅くなります。テクスチャにぼかしをかけるとノイズが減り、収束が速くなるかもしれません。

ランプ

背景、およびemissionシェーダを適用したオブジェクトとならび、シーンに照明を追加する方法としてランプがあります。前二者との違いは、ランプはレンダリングされたイメージ中で直接見えることはないということ、照明専用のオブジェクトとしてより扱いやすいということです。

Type

現在Point、Area、Sunがサポートされています。SpotとHemi lampsはサポートされておらず、それぞれPointとSunとしてレンダリングされます。しかし、将来的には動作するようになるかもしれません。よって互換性のためにそれら2種は使わないでおくほうが良いです。

Size

Blender単位でのランプの大きさ。大きくすると影やシェーディングがやわらかくなります。

Cast Shadow

これを無効にすると、ランプの光はオブジェクトにさえぎられません。光線を光源まで追跡する必要がないのでレンダリングが速くなります。

Point Lamp

すべての方向に均等に光を放射します。サイズをゼロより大きくすると球状のランプになり、影とシェーディングがやわらかくなります。

Area Lamp

正方形または長方形の領域からランバート分布で光を放射します。

Sun Lamp

指定した方向に光を放射します。ランプの位置に関係なく、常にシーンの外側、無限遠に位置付けられます。距離による減衰はまったくありません。

シーンの内部に存在するのではないので、光の強さは他とは異なる単位を使って表しています。その数値は大抵の場合他の照明よりも低くするべきです。

ノード

マテリアル、照明、および背景は、すべてシェーディングノードのネットワークを使って定義します。このノードは数値、ベクトル、カラー、シェーダを出力します。

シェーダ

ノードを構築するときに理解すべき重要な概念としてシェーダソケットがあります。サーフィスとボリュームのシェーダからの出力はシェーダであり、それはカラーというよりも光の作用として記述されます。

ノードとして使用できるシェーダはいくつかの種類があります。

- **BSDF**シェーダ: オブジェクトのサーフィスにおける光の反射、屈折、吸収を記述します。
- **Emission**シェーダ: オブジェクトのサーフィスまたはボリューム内から放射される光を記述します。
- **Volume**シェーダ: ボリューム内での光の散乱を記述します。
- **Background**シェーダ: 環境からの光の放射を記述します。

それぞれのシェーダはカラーを入力としてシェーダを出力します。これらはMixシェーダ、Addシェーダを使って合成または加算できます。ほかの処理は許されていません。結果として出力されたシェーダはレンダーエンジンによって使われ、直接光またはグローバルイルミネーションにおける光の作用が計算されます。

テクスチャ

テクスチャタイプのそれぞれは、Cyclesにおいてノードに相当します。テクスチャ座標および様々なパラメータを入力として、カラーまたは数値を出力します。テクスチャデータブロックは必要ありません。かわりにノードグループを使ってテクスチャを再使用できます。

UVマッピングとテクスチャペイントにはイメージテクスチャノードが必要です。アクティブにしたイメージテクスチャノードはTexture描画モードにおいて描画され、テクスチャペイントモードでペイントすることもできます。

すべてのノードのデフォルトのテクスチャ座標系はGenerated座標系ですが、イメージテクスチャだけはUV座標系がデフォルトになっています。それぞれのノードはテクスチャマッピングおよび結果としてのカラーを変化させるオプションをもっています。そのオプションはtextureプロパティで編集できます。

その他

ジオメトリデータ、テクスチャ座標、レイヤーシェーダ、および非物理ベースの処理のためのノード。

シェーダノード

BSDF

Diffuse(拡散)

LambertianモデルおよびOren-Nayarモデルによる拡散反射。

Color input

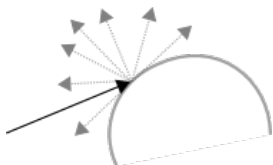
サーフィスの色。物理学的にいえば、それぞれの光の波長が反射または透過される確率。

Roughness

サーフィスのroughnessが0.0だと標準的なLambertian反射になります。数値を高くするとOren-Nayar BSDFが使われます。

BSDF output

Diffuse BSDFシェーダ。



Diffuse Shader



The Oren Nayar
Shader(Roughness 1.0)

Translucent(半透明性)

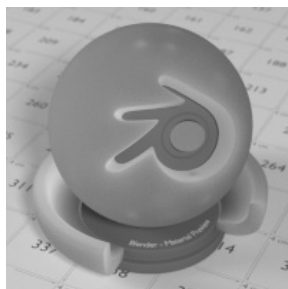
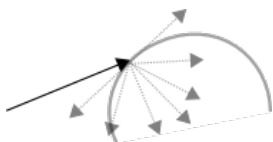
Lambertianモデルによる拡散透過。

Color input

サーフィスの色。物理学的にいえば、それぞれの光の波長が透過される確率。

BSDF output

Tranlucent BSDFシェーダ。



Translucent Shader

Glossy(光沢)

微小面分布 (microfacet distribution)による光沢反射。金属や鏡などに使います。光沢は多数の光点を生む傾向があります。

Distribution

使用する微小面分布。Sharpでは鏡のように完璧に鋭い光沢になります。BeckmannとGGXはRoughness inputを使ってぼやけた反射を表現できます。

Color input

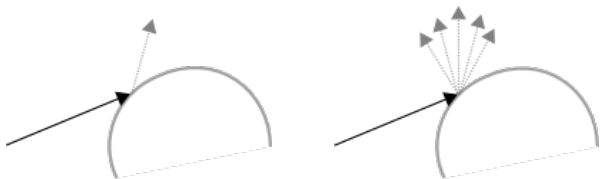
サーフィスの色。物理学的にいえば、それぞれの光の波長が反射される確率。

Roughness input

反射の鋭さに影響します。0.0で完璧に鋭く、高い数値ではやわらかい反射になります。

BSDF output

Glossy BSDFシェーダ。



A Rough Glossy Material



A Sharp Glossy Material

Transparent(透明性)

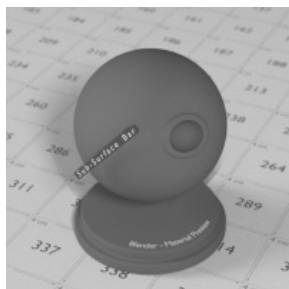
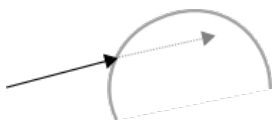
屈折を伴わない透明シェーダ。光はそこに何も無いかのようにまっすぐにサーフィスを透過します。たとえばアルファマップなどに便利です。このシェーダはライトパスに対して他のシェーダとはやや異なった影響を与えます。真っ白なシェーダの場合のみ完璧に透明になることに注意してください。

Color input

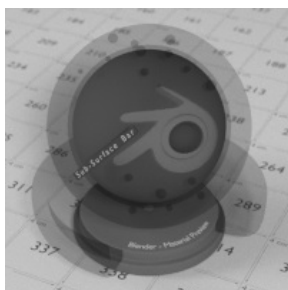
サーフィスの色。物理学的にいえば、それぞれの光の波長が遮断または透過される確率。

BSDF output

Transparent BSDFシェーダ。



The Transparent Shader



A darker color will make the transparent shader less transparent.

Glass(ガラス)

ガラスのように、サーフェイスに対して傾いた角度の光線の反射と屈折を合成します。transparentシェーダと同じく、真っ白な場合のみ透明になります。Glassシェーダは集光(caustics)によりノイズを生みやすいです。

Distribution

使用する微小面分布。Sharpでは鏡のように完璧に鋭い屈折になります。BeckmannとGGXはRoughness inputを使って粗いガラスを表現できます。

Color input

サーフェイスの色。物理学的にいえば、それぞれの光の波長が透過される確率。

Roughness input

屈折の鋭さに影響します。0.0で完璧に鋭く、高い数値ではやわらかい屈折になります。

IOR input

屈折率。光の方向が変わる度合いを定義します。1.0ではtransparentのようにまっすぐに透過し、数値を高くするとより屈折します。

BSDF output

Glass BSDFシェーダ。



Rough values will make the glass frosty.



Sharp Glass

Velvet(ベルベット)

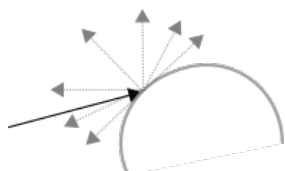
布などの素材に適したベルベット反射。現在のところ、Velvetシェーダは光点が目立つ傾向があります。

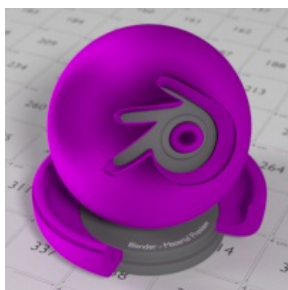
Color input

サーフェイスの色。物理学的にいえば、それぞれの光の波長が反射される確率。; Sigma input: まだ解説はありません。

BSDF output

Velvet BSDFシェーダ。





The Velvet Shader

Emission (放射)

Lambertianモデルによる放射。マテリアルおよびランプのサーフィスの出力に使います。

Color input

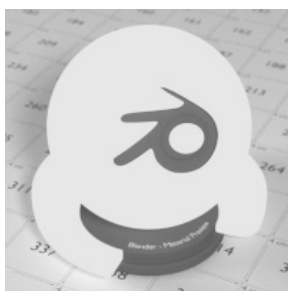
放射される光の色。

Strength input

放射される光の強さ。PointとAreaに関しては単位はワットです。マテリアルに関しては、1.0にするとColor inputで指定したカラーそのものと同じ明るさになります。つまりその場合、陰影付けされていないような状態になります。

Emission output

Emissionシェーダ。



An Emissive material. It has a brightness of 1, making it perfectly shadeless.

Background (背景)

背景からの光の放射。これはワールドのサーフィスにのみ使うべきです。その他の場合は無視されます。

Color input

放射される光の色。

Strength input

放射される光の強さ。

Background output

Backgroundシェーダ。

Holdout (抵抗)

Holdoutシェーダはコンポジット作業に便利です。このシェーダが適用された領域は、イメージ上ではアルファ値が0の「穴」になります。

Holdout output

Holdoutシェーダ。



The black area is a region with zero Alpha.

Mix and Add (合成と加算)

階層化されたマテリアルに使い、Fac inputをたとえばBlend Weightノードと接続できます。

Shader inputs

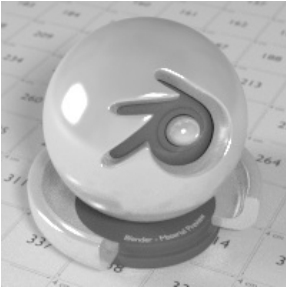
合成するシェーダ。たとえばFacソケットでそれぞれ確率を指定した二つの入射光など。

Fac input

二つのシェーダを合成する際の割合。0では第一のシェーダとまったく同じになり、1では第二のシェーダと同じになります。

Shader output

合成されたシェーダ。



For example, a mix of a glossy and a diffuse makes a nice ceramic material.

テクスチャノード

Image (イメージ) テクスチャ

イメージファイルをテクスチャとして使います。

Image Datablock

イメージソースとして使われるイメージデータブロック。現在のところ、Blender でサポートされているイメージがすべてCyclesで使用できるわけではありません。特定のgenerated、packedイメージまたはアニメーションは現在サポートされていません。

Color Space

入力されたイメージのカラースペース。sRGBまたはリニアです。通常の画像の場合はsRGBにするべきですが、バンプまたはアルファマップの場合はリニアのほうが値としてふさわしいでしょう。

Vector input

テクスチャ座標。このソケットに何も接続されていない場合、アクティブなUVレイヤーのUVマップが使われます。

Color output

イメージのRGBカラー。

Environment (環境) テクスチャ

環境マップイメージをテクスチャとして使います。イメージは緯度／経度 ('latlong') 形式であるとみなされます。

Image Datablock

イメージソースとして使われるイメージデータブロック。現在のところ、Blender でサポートされているイメージがすべてCyclesで使用できるわけではありません。特定のgenerated、packedイメージまたはアニメーションは現在サポートされていません。

Color Space

入力されたイメージのカラースペース。sRGBまたはリニアです。

Vector input

テクスチャを参照するときのテクスチャ座標。このソケットに何も接続されていない場合、Z軸を上にして環境にマッピングされます。

Color output

イメージのRGBカラー。

Sky (空) テクスチャ

手続き型の太陽および空テクスチャ。

Sun Direction

太陽の方向ベクトル

Turbidity

空の曇り具合。

Vector

テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Color output

テクスチャカラー出力

Noise (ノイズ) テクスチャ

手続き型pelinノイズテクスチャ。Blender内臓のCloudsに似ています。

Vector input

テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Scale input

全体的なテクスチャのスケール。

Detail input

ノイズの細部の量。

Distortion input

ゆがみの量。

Color output

テクスチャカラー出力。

Fac output

テクスチャ強度出力。

Wave Texture

Procedural bands or rings texture with noise distortion.

Type

Bands or Rings shaped waves.

Vector input

テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Scale input

全体的なテクスチャのスケール。

Detail input

ノイズの細部の量。

Distortion input

ゆがみの量。

Color output

テクスチャカラー出力。

Fac output
テクスチャ強度出力。

Voronoi(ボロノイ) テクスチャ

ボロノイセルを生成する手続き型テクスチャ。

Type
IntensityまたはCellsの出力。

Vector input
テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Scale input
全体的なテクスチャのスケール。

Color output
テクスチャカラー出力。

Fac output
テクスチャ強度出力。

Musgrave(マズグレイブ) テクスチャ

高度な手続き型ノイズテクスチャ。

Type
Multifractal、Ridged Multifractal、Hybrid Multifractal、fBM、Hetero Terrain。

Vector input
テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Scale input
全体的なテクスチャのスケール。

Detail input
ノイズの細部の量。

Dimension input
未定

Lacunarity input
未定

Offset input
未定

Gain input
未定

Color output
テクスチャカラー出力。

Fac output
テクスチャ強度出力。

Gradient(勾配) テクスチャ

グラデーションテクスチャ。

Type
LinearまたはSphericalグラデーション。

Vector input
テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Color output
テクスチャカラー出力。

Fac output
テクスチャ強度出力。

Magic テクスチャ

サイケデリックなカラーテクスチャ。

Depth
反復の数。

Vector input
テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Distortion input
ゆがみの量。

Color output
テクスチャカラー出力。

Fac output
テクスチャ強度出力。

Checker Texture

チェック模様のテクスチャ。

Vector input
テクスチャ座標。ソケットに何も接続されていない場合、Generated texture座標が使われます。

Color1/2 input

チェック模様の色。

Scale input

全体的なテクスチャのスケール。

Color output

テクスチャカラー出力。

Fac output

テクスチャ強度出力。

その他のノード

Value (数値)

スカラー値を他のシェーダに対して入力します。

Value
数値出力。

RGB

RGBカラーを他のシェーダに対して入力します。

Color
RGBカラー出力。

Geometry (ジオメトリ)

現在シェーディングしている点のジオメトリ情報。ベクトルの座標系は *World Space* です。ポリウムシェーダに対しては *position* と *incoming vector* のみ使用できます。

Position
シェーディングする点の位置。

Normal
(スムーズ化やバンプマップ後の)サーフィスの法線。

Tangent
サーフィスのタンジェント。

True Normal
ジオメトリまたは元の法線。

Incoming
シェーディングする点から視線に対する方向ベクトル。

Parametric
サーフェス上のシェーディング点のパラメータ座標 (U, V, 0.0)。

Backfacing
面が裏側から見られているなら1.0、表からなら0.0。

Texture Coordinates (テクスチャ座標)

代表的なテクスチャ座標系。テクスチャノードのVectorによく使われます。

Generated
変形していない状態のメッシュの頂点位置から自動的に作られる座標系。アニメーション中もサーフィスに貼りついています。変形していない状態のメッシュのパウンディングボックスに対して0.0から1.0の範囲で決められます。

UV
アクティブなUVレイヤーのUVテクスチャ座標。

Object
オブジェクト空間の座標系。

Camera
カメラ空間の座標系。

Window
シェーディングする点のレンダリング画面上での位置。左から右、および下から上に向かって0.0から1.0の値になります。

Reflection
鋭い反射の方向。環境マップによく使われます。

Attribute (属性)

オブジェクトまたはメッシュの属性を受け取ります。現在、UVマップと頂点カラーレイヤーはこの方法で名前によって受け取れます。レイヤーと属性は追加される予定です。また内部で使われている属性、たとえばP (position)、N (normal)、Ng (geometric normal)なども受け取れます。ただしこれらについてはもっと便利なノードがあります。

Name
属性の名前。

Color output
属性から補間されたRGBカラー。

Vector output
属性から補間されたXYZベクトル。

Fac output
属性から補間されたスカラー値。

Mapping (マッピング)

座標を変換します。テクスチャ座標に修正を加えるために使われます。

Translation
ベクトル変換。

Rotation
XYZ軸に関するベクトルの回転。

Scale
ベクトルのスケール。
Vector input
変換されるベクトル。
Vector output
変換されたベクトル。

Layer Weight(レイヤーウェイト)

重み付けをした出力。Mix Shaderノードでシェーダを合成するときによく使われます。

Blend input
入力された第一のシェーダと第二のシェーダをブレンドします。
Fresnel output
フレネル反射のウェイト。たとえばプラスチックの質感を再現するためにdiffuseとglossyをレイヤー合成する場合などに使えます。これはFresnelノードに似ていますが、それと違って入力が0.0から1.0なので、こちらのほうが便利がある場合もあるでしょう。
Facing output
第一のシェーダと第二のシェーダの割合。サーフィスが画面に対して正面に向いている状態から傾いた状態になるにしたがって、第一のシェーダから第二のシェーダに移り変わります。

Fresnel(フレネル)

フレネル反射。光がどの程度屈折して、どの程度反射されるかを計算します。出力されたウェイトはMix Shaderノードでレイヤー合成するのに使えます。これはサーフィスの法線方向と視線の方向の角度に応じて変化する値です。

IOR input
マテリアルの屈折率。
Fresnel output
フレネルウェイト。その点において光が透過せずに反射する確率を示します。

Light Path(ライトパス)

どのような光に対してシェーダが実行されているかを調べるノード。非物理ベースの処理をする際に便利です。それぞれのタイプについての詳細は[Light Paths](#)のドキュメントを参照。

Is Camera Ray output
1.0ならcamera rayに対してシェーディングが実行されています。0.0ならされていません。
Is Shadow Ray output
1.0ならshadow rayに対してシェーディングが実行されています。0.0ならされていません。
Is Diffuse Ray output
1.0ならdiffuse rayに対してシェーディングが実行されています。0.0ならされていません。
Is Glossy Ray output
1.0ならglossy rayに対してシェーディングが実行されています。0.0ならされていません。
Is Singular Ray output
1.0ならsingular rayに対してシェーディングが実行されています。0.0ならされていません。
Is Reflection Ray output
1.0ならreflection rayに対してシェーディングが実行されています。0.0ならされていません。
Is Transmission Ray output
1.0ならtransmission rayに対してシェーディングが実行されています。0.0ならされていません。

Light Paths

レイタイプ

レイタイプは4つのカテゴリに分けることができます。

- Camera: カメラから一直線に来るレイ
- Reflection: サーフェス上の反射により生成されたレイ
- Transmission: サーフェスを通過した伝搬(透過)により生成されたレイ
- Shadow: 透過影用に使用されたレイ

Reflection と transmission のレイは、さらに以下のプロパティを持つことができます。

- Diffuse: ディフューズ反射もしくは伝搬(透過)により生成されたレイ
- Glossy: グロッキーなスペキュラ反射もしくは伝搬により生成されたレイ
- Singular: 完全にはっきりした反射もしくは伝搬により生成されたレイ

Light Path ノードは計算中のシェーディングのレイのタイプを調べるのに使用できます。



バウンスコントロール

光のバウンスの最大数を手動でコントロールすることができます。理想では無限にすべきですが、実際にはもっと少ない反射数で充分、もしくは早く収束させるため、一部の光の相互作用を故意に省略することができるでしょう。

Diffuse Reflection、Glossy Reflection、Glossy Transmission のバウンス数も個別にコントロールできます。

光の最大(maximum)バウンス数より小さい最小(minimum)バウンス数を指定した時、Lights paths は確率的に打ち切られます。この時、最小より長いパスは、その画像の照明にあまり寄与しないと予測されればランダムに止められます。

それでもまだ同じ画像に収束はしますが、レンダリングが速くなる一方、ノイズが多くなる可能性があります。

代表的なノイズ源がコースティクスで、これは一つの Glossy バウンスに続く複数の Diffuse バウンスです(カメラからスタートしたと仮定)。これらを完全に無効にするオプションが利用可能です。

透過

transparent BSDF シェーダには特別な処理が行われます。レイが通過するとき、まるで何もそこにはないかのように光がまっすぐ進みます。

transparent BSDF を通過する時、レイタイプは変化しません。

アルファパス出力も、transparent BSDF では違いがあります。他の transmission BSDF の場合、光の方向を変更するため不透明とみなされます。これでは Alpha Over 合成に使用できません。しかし transparent BSDF ではそれが可能です。

Transparent の最大バウンス数は、他のバウンス数とは別にコントロールされます。また、Transparent バウンスの確率的停止に使用することもでき、たくさんの透過の層のレンダリングに役立つでしょう。

意味的にはレイがまるで形状に衝突しないかのようなのですが、レンダリングパフォーマンス的には透過段階ごとにシェーダの実行とレイの追跡が必要になることに注意してください。

Ray Visibility

オブジェクトには特定のレイタイプを不可視に設定することができます。

- Camera
- Diffuse reflection
- Glossy reflection
- Transmission
- Shadow

これは例えば、エミット Mesh をカメラのレイから不可視にすることなどに使用できます。デブプリケータの可視設定は継承され、もし親オブジェクトがいくつかのレイタイプから隠ぺいされていれば、子も同様にこれらから隠ぺいされます。

パフォーマンスの点からいえば、これらのオプションの使用は、同じエフェクトを行うシェーダノードセットアップを使用するより効率的です。あるレイから見えないオブジェクトは、すでにレイ・トラバースルからスキップされ、少ないレイ・キャストとシェーダの実行をもたらします。

インテグレータ

インテグレータは照明効果を計算するアルゴリズムです。Cyclesは現在1種類のインテグレータをサポートしています。直接光サンプリングをとまなうパス追跡インテグレータです。これは様々なセッティングの照明に対して動作しますが、集光(Caustics)およびいくつかの複雑な状態にはあまり適していません。

シーンの設定

Samples

Render Samples

最終的なイメージのピクセル一つ一つについて、パスを追跡する回数。回数が多いほどよりノイズが少なく正確な結果になります。

Preview Samples

3Dビューにおけるサンプル数。

Seed

乱数発生の種類。数値を変えるとノイズパターンが変わります。

Bounces

Max Bounces

光のバウンスの最大回数です。品質を追求するならばこの値は最大に設定するべきです。しかし実際には、レンダリングを速めるために低い数値にするのもよいでしょう。1に設定すると直接光のみになります。

Min Bounces

それぞれのライトパスにおける光のバウンスの最小回数です。回数がこれを超えたパスは、イメージにあまり寄与していない場合ロシアンルーレット方式で停止されます。数値を高くするとノイズが減りますが、レンダリング時間がかなり遅くなる可能性もあります。

Diffuse Bounces

拡散バウンスの最大回数。

Glossy Bounces

光沢バウンスの最大回数。

Transmission Bounces

透過バウンスの最大回数。

Transparency

Transparency Max

透明バウンスの最大回数。

Transparency Min

透明バウンスの最小回数。それ以降はロシアンルーレット方式で停止します。

Transparent Shadows

直接光サンプリングについて、サーフィスの透明度を影にも使います。

Caustics

No Caustics

パスの追跡は十分なサンプル数があれば集光をレンダリングできますが、現実的にはノイズが多すぎて効果的ではないかもしれません。このオプションは集光の再現を全面的に無効にします。

マテリアルの設定

Sample as Lamp

デフォルトでは、光を放射するマテリアルをもったオブジェクトは直接光と間接光両方のサンプリング法を用います。しかし場合によっては、いくつかのマテリアルは直接光を無効にしたほうが全体的なノイズが減る可能性があります。そのためにはこのオプションを無効にします。これは他の光源と比べて少ない量の光を放射している大きなオブジェクトにおいて特に有用です。

ワールドの設定

Sample as Lamp

デフォルトではワールドからの光は間接光サンプリングのみで計算されます。しかし複雑な環境マップの場合これはノイズが多くなる可能性があります。BSDFのサンプリングにおいて環境マップの明るい点を検出するのが容易でないことがあるからです。このオプションを有効にすると、ワールドの背景はランプとしてサンプリングされ、明るい部分には自動的に多くのサンプル数が与えられます。

Map Resolution

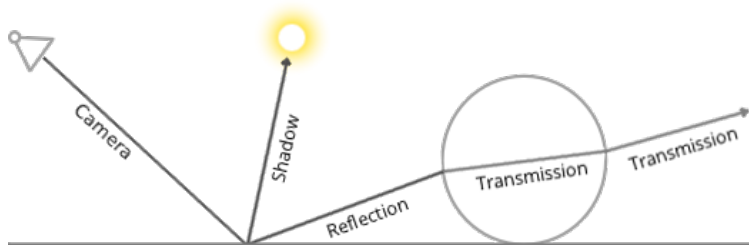
Sample as Lampが有効の場合、この値で重点マップのサイズを指定します(解像度x解像度)。レンダリングが始まる前に、ワールドシェータをグレースケールイメージとして「ベイク」することで重点マップが作られます。そのマップは、背景のどの部分が明るい(よって多くのサンプルを割り当てるべきである)かを決めるのに使われます。解像度を高くするとより正確なサンプリングがされますが、セットアップの時間と使用メモリが増大します。

ノイズの削減

最終レンダリングを実行するとき、可能な限りノイズを減らすのは重要なことです。これから、多くの物理法則を無視したトリックの話をしませんが、納得のいく時間内でアニメーションをレンダリングし終える為には極めて重要なことです。ノイズの違いをご覧になるには、サンプルイメージをクリックして拡大してください。

パス・トレーシング Path tracing

Cyclesは次に起こるイベントの見積もりを利用しつつパス・トレーシングを使用します。そしてそれは、集光の様な光のエフェクトの類をレンダリングするのが苦手です。しかし、他のレンダリング・アルゴリズムと比べて、より大きなシーンを詳細にレンダリング出来るという利点があります。例えばフォトンマップをメモリに格納しておく必要がありません。また、例えば bidirectional path tracing (双方向軌跡追跡)と比較すると、イメージキャッシュを必要に応じた使用する場合において、光線を比較的に整合性がとれた状態に保つことが出来ます。

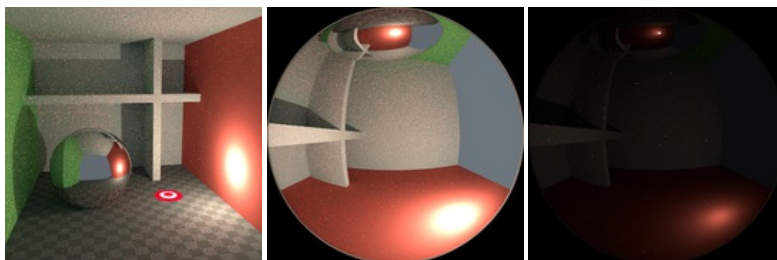


現実世界で起こる事とは逆の事を行われます。光源からシーンへ、そしてカメラへという方向ではなく、むしろ、カメラからシーンへ、そして光源へと向かって光線を追跡します。このやり方では、カメラに突入してこない光線を無駄に計算しなくてもよい、という利点があります。しかし同時に、どの光線がライティングに多く貢献しているのかを見つけるのが困難である、ということでもあります。

以下のドキュメントで詳細を知ることが出来ます。[Light Paths](#)、[Integrator](#)

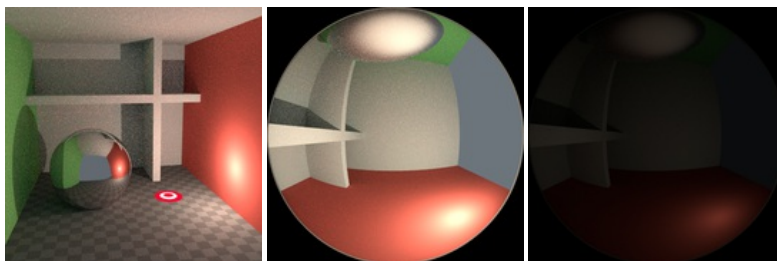
ノイズはどこから来るのか

ノイズはどこから来るのか理解するために、このシーンを例に説明します。ある特定の位置に突入してくる光線を追跡する事は、それはまさに diffuse (拡散光) シェーダーが行う事なのです。このサーフェスで反射し飛び出す光線を知る為に、全てのこれらのピクセルからの平均色を知る必要があります。球体の艶のハイライトと、光源が近くの壁に落とす明るい小領域に注目してください。これらのホットスポットは、このイメージの他の部分より100倍明るく、このピクセルのライティングに対して顕著に貢献するでしょう。



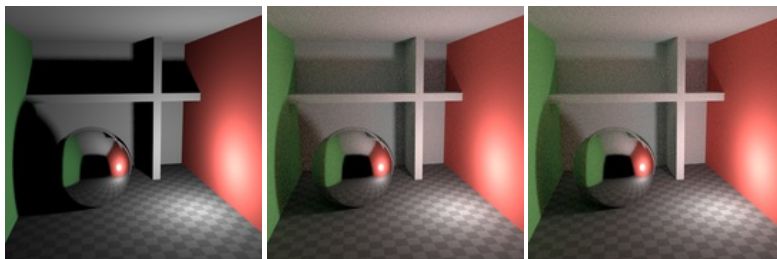
ランプはレンダラーにとって既知の光源で、見つけるのはそう難しくはないでしょう。しかし、その光源により引き起こされる艶のハイライトはまた別の(光源と成り得る為、)問題となります。path tracing を用いて出来る最良の事は、全ての重要な明るい小領域を見つける為に、かいつまみながら半球全体にわたってランダムに光線を配分できる事でしょう。もしいくつかのピクセルと引き換えに、いくつかの明るい小領域を見逃しても、他のもうひとつと引き換えにそれをちゃんと見つけますが、その事はノイズの原因になります。サンプルを多く採れば採るほど、全ての重要な光源をカバーできる確率は高まります。

いくつかのトリックを用いてノイズを減らすことが出来ます。もし明るい小領域をボカせば、より暗くなり大きくなりますし、見つけ易くノイズも少なくなります。この事は完全に正確な同等な結果を与えてくれる事はないでしょうが diffuse もしくは soft glossy reflection を見渡せば、十分似ている結果が見えます。以下は、Filter Glossy と Smooth Light Falloff を用いた例です。



跳ね返り Bounces

現実世界では光は、その速度が非常に速いので、膨大な回数跳ね返ります。実際には、跳ね返りが多いほど、多くのノイズが入ります。Limited Global Illumination (制限された大域照明) プリセット異なるシェーダー向けのより少ない跳ね返りの様なものを使用するのが良いかもしれません。Diffuse サーフェスは典型的に、より少ない跳ね返りでレンダリングをやってのけることが出来ます。一方、艶は少し多目で、ガラスのような Transmission シェーダーは最も多く跳ね返りを必要とします。



同じく重要な事は、値1.0のコンポーネントを持つシェーダー色を使わない事、またはそれに近い事をしない事です。最大でも0.8以下に保ち、ライトの方を明るくして下さい。現実世界ではサーフェスが完全に全ての光を反射するような事は稀です。しかし、もちろん例外はあります。ガラスはほとんどの光線を通しますから、より多くの跳ね返りが必要です。色要素に高い値を入れるとノイズが入りがちです。なぜなら、そうした場合の光源の強度は、それがサーフェスで跳ね返る際にそれほど減少しないからです。

集光と艶フィルター Caustics and Filter Glossy

集光(Cautics)は蛍のようなノイズを引き起こす原因としてよく知られています。ソフトな艶や拡散反射を通して見るスペキュラーハイライトを、発見するのが難しい場合に発生します。拡散反射の後ろにある艶を全体的に無効にする、[No Caustics](#)オプションがあります。多くのレンダリングエンジンは典型的にデフォルトで無効になっています。

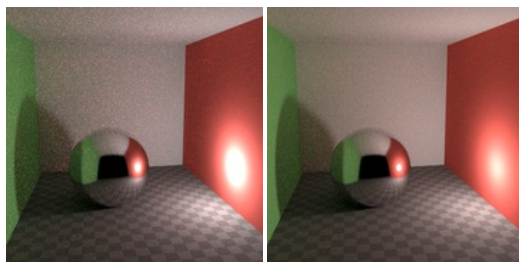


どうい方法を採用したとしても、集光を使わない事は、光の欠落を招きますし、シャープな艶の反射がソフトな艶の反射から跳められるような場所で起こるケースには、依然として対応しません。正確性を確保するのにコストがかかりますが、そのようなケースでのノイズを減らす為に、次のオプションが存在します。[Filter Glossy](#)

上のイメージはデフォルト設定を示しています。集光無しで、filter glossyを1.0に設定しています。

ライトの減衰 Light Falloff

現実世界での真空中での光は常に $1/(distance^2)$ の割合で減衰します。しかしながら距離がゼロに近づくにつれ、この値は無限大に近づき、イメージの中で非常に明るい領域を得ることになります。これらは間接照明にとってかなり問題になります。このような小さいけれども非常に明るい小領域に出くわす確率は低いので、ほとんど起こりません。これが、蛍型ノイズが発生する典型例です。

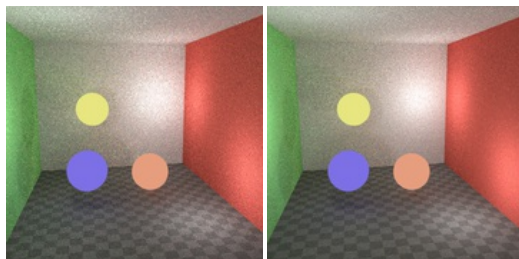


この問題を減らすために、[Light Falloff node](#) はライトが付近のサーフェスを照らす場合の最大の強度を、減少させるために使用される、スムーズ要因を持ちます。上のイメージはデフォルトの減衰と Smooth 値が1.0のときの状態を示しています。

ランプとして採取 Sample as Lamp

自己発光シェーダーを伴うマテリアルは、[Sampled as a Lamp](#)で設定できます。この事は、ランダムにあちこちに跳ね返る光線が最終的にそこへ至るというよりも、それらのマテリアルに向かって直接に光線を送ってもらう、という事を意味します。非常に明るいメッシュ光源に対してであれば、これは顕著にノイズを減らすことが出来ます。しかしながら、発光がそれほど明るくない場合、「この方法でこのマテリアルを見つける事が重要である他のもっと明るい光源」からのサンプル群を、このマテリアルは取り除いてしまうでしょう。

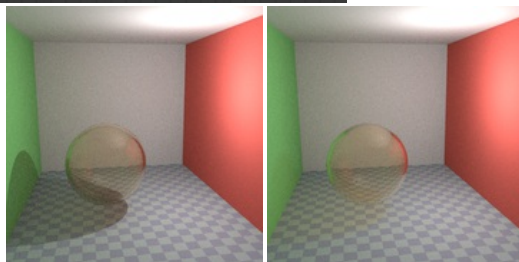
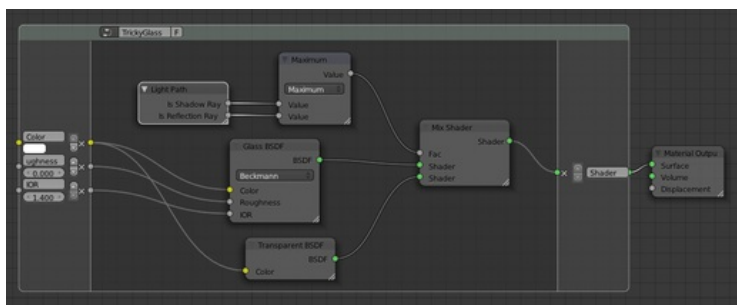
ここでの最適な設定は推測が困難です。試行錯誤が必要かもしれませんが、とにかく発光しているオブジェクトだけが局所的に関係しているということは、ほぼ明らかです。一方、ランプとして使用されているメッシュは、このオプションを有効にすることを要求するでしょう。これは、発光している球体がライティングにほとんど影響を与えないという例です。そして、この右のイメージは、Sample as Lamp を無効にしたことにより、微かに少ないノイズでレンダリングされています。



ワールドの背景は、[Sample as Lamp](#) オプションも持っています。これは、スムーズではなく、小さな明るい小領域を含む環境マップにとってかなり便利です。このオプションはその時に、とある事前処理において、特定の明るい小領域を決定するでしょう。そして、それらに直接光線を送ります。またここで、このオプションを有効にすることで、他のより重要な光源から採取されたサンプル群を、もし必要なければ、取り除くかもしれません。

ガラスと半透明の影 Glass and Transparent Shadows

集光を無効にすると、ガラスは影を失うでしょう。また、艶フィルターを用いると、それらは非常にソフトになるかもしれません。直接見える場合には **Glass BSDF** を使用し、間接的に見える場合は **Transparent BSDF** を使用するというようなガラスシェーダーを作成することが出来ます。Transparent BSDF は、半透明な影がサーフェスをまっすぐ通して光源を見つける為に使用され得ます。そして、正しい色つきの影を出すことができます。しかし集光は出ません。Light Path ノードはその二つのシェーダーのどちらかを、いつ使用するかを決定します。

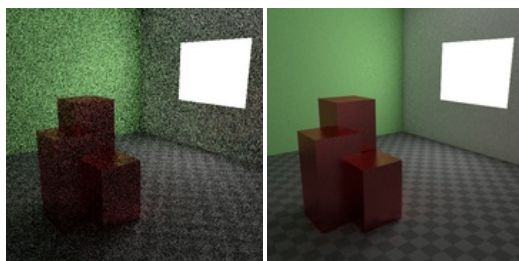


上では、ガラスの透明性トリックの為にノード設定について見ることが出来ます。左の例では、集光の欠落による強すぎる影がレンダリングされています。右の例ではそのトリックを使用しています。

窓型ライト Window Lights

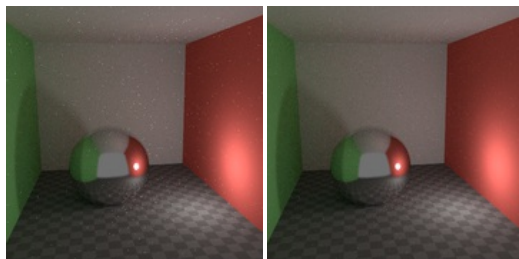
開いたドアや窓を通して室内へ入ってくる太陽光をレンダリングする場合、integrator がそれらに対して、それらの通り道を見つけてあげる事は非常に困難です。そこで、開いている窓の内側に配置された発光シェーダーを伴う平面オブジェクトで代用することが出来ます。そうすれば、integrator はどの方向から光線を発射されているか知ることができます。カメラ視線に対しては、このメッシュ光源を不可視にすれば、窓の外の景色が見えなくなることはありません。これは、そのオブジェクトのカメラ視線の可視性を無効にするか、マテリアル内でガラスシェーダーと発光シェーダーを切り替えることにより実現できます。

下の二つのイメージは同じレンダリング時間がかかっています。二番目の例は、窓の中にメッシュ光源を置いたものです。



蛍ノイズの固定 Clamp Fireflies

先に出た全てのトリックを用いて、蛍ノイズは申し分なく駆逐できるでしょうが、依然起り得ます。それに対して、任意のピクセルに影響を及ぼす個々の光線サンプルの強度の値を、**Clamp setting integrator** を用いて、任意の最大値で固定できます。もし非常に低く設定すると、ハイライトが欠落するかもしれません。これは、ブルームやグレアのようなカメラエフェクトを保全する為には有効かもしれません。

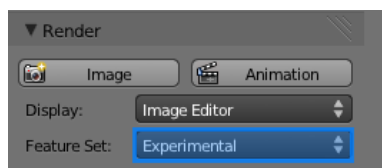


"

実験的機能

Cycles のいくつかの機能はまだ完成していませんが、テスト用にすでにビルドに入っています。これらの機能は動作しなかったり、Blender がクラッシュしたり、後のバージョンで挙動が変更されるかもしれません。

デフォルトでは非表示になっていますが、Renderプロパティで、Feature Set を Experimental に変更することで利用可能になります。



現在実験的機能とみなされている物:

- OpenCL デバイス
- ディスプレイスメント
- サブディビジョン

GPU レンダリング

はじめに

GPU レンダリングは CPU の代わりにグラフィックカードを使用してレンダリングさせることができるようにします。最近の GPU は非常に多くの処理を行うよう設計されているため、これによりレンダリングのスピードアップが期待できます。一方、限られたメモリにより、複雑なシーンのレンダリングにいくつか制限があったり、表示とレンダリングに同じグラフィックカードを使用した時のインタラクティブ性に問題があります。

Cycles は二つの GPU レンダリングモードを持っています。CUDA は nVidia のグラフィックカードで望ましい方法で、OpenCL は AMD/ATI のグラフィックカードでのレンダリングに対応する意味であります。ただしその実装は実験段階で、現在アンビエントオクルージョンのような画像のレンダリングにのみ対応しています。

Userプリファレンスによる設定

GPU レンダリングを使用するには、Userプリファレンスに行き、System タブにある、Compute Device(s)を選択します。次に、各シーンで Render プロパティ内で使用する CPU もしくは GPU レンダリングを設定できます。

CUDA

NVidia CUDA は、**NVidia** のグラフィックカードでの GPU レンダリングに対応しています。GTX 2xx (shader model 1.3)以降のグラフィックカードに対応していますが、*GTX 4xx もしくは GTX 5xx(shader model 2.x)のカードの使用を推奨します*。これらのみがスピードアップが期待でき、初期のカードでは大抵、単にCPUを使用するよりも遅いからです。

Cycles はすべての OS において、最新のドライバのインストールが必須です。皆さんの OS にあったバージョンの Blender をダウンロードしてください。つまり、64ビットの OS では64ビット版 Blender をダウンロードしなくてはなりません。

[CUDA カードとシェーダモデルリスト](#)

古いカード

Mac と Linux では実行時に、公式に対応していないカード用のカーネルのコンパイルができます。GeForce 8xxx、9xxx カードは公式リリースには含まれていませんが、[Experimental\(実験的\)機能](#) を有効にすることで動作するかもしれません。

そのためには、[CUDA toolkit version 4.0](#) (64ビット版)以降をインストールする必要があります。他のバージョンも動くかもしれませんが、非対応です。初回のレンダリングの終了後、(Blender は)カーネルを皆さんの GPU アーキテクチャ用にコンパイルしないとけません。Cycles は標準的な GPU カーネルに比べ非常に複雑であるため、コンパイルは恐らく40秒から数分かかり、グラフィックカードモデルによっては、メモリ使用量が約2GBにもなるでしょう。

OpenCL

まだ実装は完了しておらず、[実験的\(Experimental\)機能](#)とされています。

OpenCLを使用したレンダリングはまだ完全には対応していません。ただし随時作業中であり、もっと多くのグラフィックカードに対応できるでしょう。AMD OpenCL コンパイラにより私たちのフル・カーネルがコンパイルされないため、現在、シンプルなクレイレンダリングのみ対応しています。

初回のレンダリング終了後、(Blender は)カーネルを皆さんの GPU アーキテクチャ用にコンパイルしないとけません。Cycles は標準的な GPU カーネルに比べ非常に複雑であるため、コンパイルは恐らく40秒から数分かかり、グラフィックカードモデルによっては、メモリ使用量が約2GBにもなるでしょう。

OpenCL version 1.1以上が必要です。

よくある質問と答え

レンダリング中、Blender が無反応になるのはなぜ？

グラフィックカードがレンダリング中、ユーザインターフェイスを再描画できなくなり、インタラクティブに反応しなくなります。私たちはこの問題を、GPUからコントロールをできるだけ頻繁に奪回することで回避しようとしていますが、特に重いシーンでは完全にスムーズなインタラクティブ性は保証できません。これはグラフィックカードの真の解決策が存在しない制限の一つですが、将来的には若干改善できるかもしれません。

できるなら、二つ以上のGPUを導入し、一つを表示用、もう一つをレンダリング用として使用するのがベストでしょう。

なぜCPU上でレンダリングされたシーンがGPU上ではレンダリングされないのでしょうか？

おそらく原因は色々ありますが、もっとも一般的なのはみなさんのグラフィックカード上に十分なメモリがないことでしょう。現在、グラフィックカードのメモリ量にあったシーンのみレンダリングでき、これは通常 CPU より少ないです。例えば、8k、4k、2k、1kの画像テクスチャでは、それぞれ256MB、64MB、16MB、4MBのメモリを占めることに注意してください。

私たちは GPU メモリより大きなシーンに対応したシステムを追加する予定ですが、これはすぐには追加されません。

複数の GPU をレンダリングに使用できますか？

まだ対応していませんが、そう遠くないうちに追加することを計画しています。

複数の GPU では利用可能なメモリ量は増えますか？

いいえ。各 GPU は自身が保有するメモリのみアクセス可能です。

Nvidia と AMD, CUDA と OpenCL ではどれがレンダリングが高速ですか？

現在、nVidia + CUDA がレンダリングが高速です。こうなる根本的な理由はなく、CUDA 固有の機能も使用していません。コンパイラがもっと成熟すれば、大きなカーネルにもっとよく対応できると思われます。OpenCL については作業を行うフルカーネルがまだないため、対応はまだ途中で、あまり最適化もされていません。

エラーメッセージについて

Unsupported GNU version! gcc 4.5 and up are not supported!

Linux では、GCCのバージョンによってこのエラーが出ることがあります。

/usr/local/cuda/include/host_config.h から以下の行を削除してください。

```
#error -- unsupported GNU version! gcc 4.5 and up are not supported!
```

CUDA Error: Invalid kernel image

このエラーが Windows 64ビットで出た場合、32ビット版ではなく、64ビットビルドの Blender を使用してるかを確認してください。

CUDA Error: Out of memory

これは大抵の場合、シーンを GPU 上に格納するためのメモリが足りないことを示します。現在、グラフィックカードのメモリ量にあったシーンのみレンダリング可能で、これは通常 CPU より少ないです。詳細は前述してあります。

The NVIDIA OpenGL driver lost connection with the display driver

...異常な Windows のタイムアウト制限の所為で続行が不可能になっています。

GPU を表示とレンダリングの両方に使用している場合、Windows は GPU がレンダリングを計算できる時間を制限します。特に重いシーンでは、Cycles は非常に多くの GPU 時間を使うことがあります。▼Performance の Tile Size を小さくすれば、この問題が軽減されるかもしれませんが、真の解決策は、表示とレンダリングに別々のグラフィックカードを使用することです。

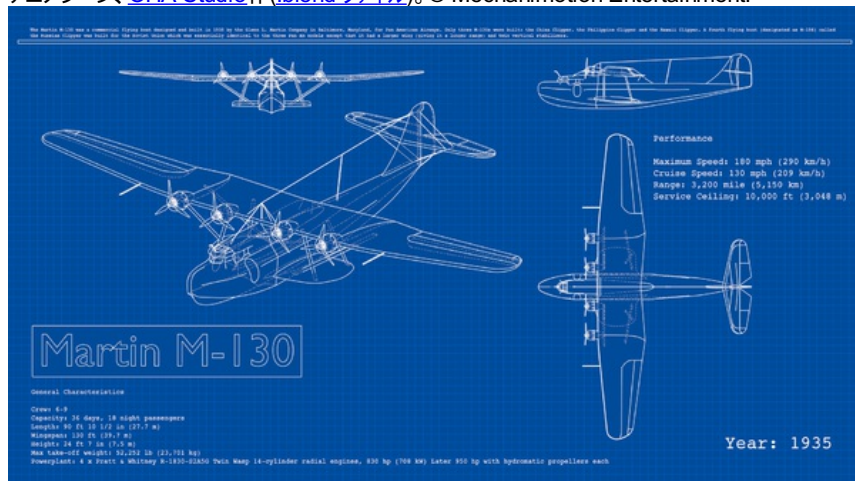
FreeStyle とは？

Freestyle はエッジ(境界)とラインベースのノンフォトリリスティック(NPR)レンダリングエンジンです。メッシュデータと Z 深度情報を頼りに、選択されたエッジタイプ上に線を描画します。様々なラインスタイルを追加することで、アーティストック風(「手描き」や「ペイント風」など)またはテクニカル風(きっちりした線)の表現を作成できます。

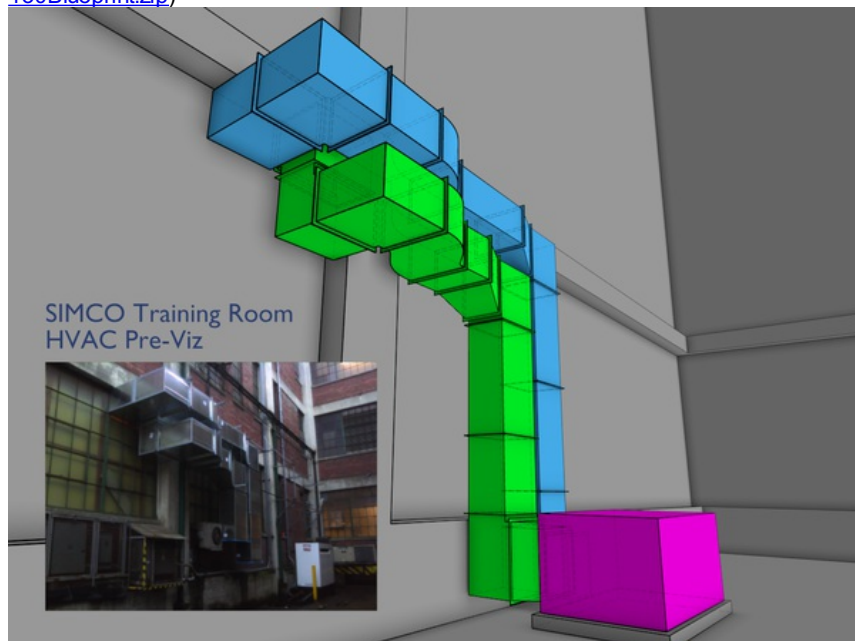
[Python スクリプティング\(英文\)](#)と[パラメーター編集モード\(Parameter Editor\)](#)の二つの操作モードがあり、パワフルで多様なラインスタイルと成果を可能にします。日本の毛筆(big brush)やアニメ風(cartoon)、製図(blueprint)、深度による幅(thickness-with-depth)のようなラインスタイルが、Python によりあらかじめスクリプト化されています。パラメーターエディターモードは破線などの機能を直観的に編集でき、複数のラインタイプとエッジ定義を簡単に設定できます。その上さらにラインスタイルモディファイアが導入されており、無限の可能性がります！



アニメーション、OHA Studio作([blend ファイル](#))。© Mechanimotion Entertainment.



1935年の Martin M-130 の設計図レンダリング、LightBWK氏作。CC0。注意：非常に重いです！ Blender のストレステスト用に作成されたため、クラッシュする可能性があります。(File:[M-130Blueprint.zip](#))



HVAC Pre-Viz, Lee Posey氏作。CC0 (File:[HVACPreViz.zip](#))

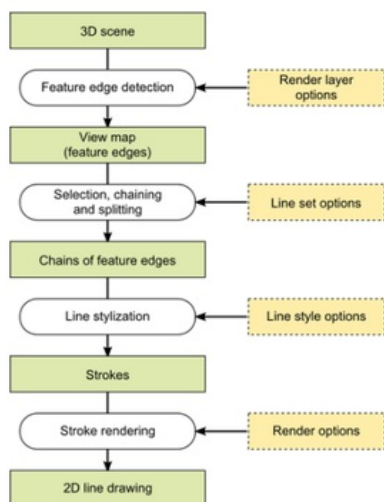


Kitchen, Vicente Carro氏作。© AnigoAnimation

他にも多くの作品があります : http://wiki.blender.org/index.php/Dev:Ref/Release_Notes/2.67/FreeStyle#Freestyle_Artwork_Showcase

ワークフロー

1. FreeStyle をプロパティ(Properties)エディタ → レンダー(Render) タブ → FreeStyle パネルのチェックボックスで有効にします。FreeStyle は Blender 内部レンダラでのみ利用可能であることに注意してください。
2. Freestyle の設定は、新しい レンダーレイヤー(Render Layers) タブにあります。
3. 一つのレンダーレイヤーにつき、一つのビューマップ(Viewmap)を持つことができます。ビューマップはエッジ検知設定 (クリース角度(Crease Angle)、カリング(Culling)のON/OFF、面のスムーズさ(Face Smoothness)のON/OFF、マテリアル境界(Material Boundaries)のON/OFF、詳細オプションの球半径(Sphere Radius)とKr 派生イプシロン(Kr Derivative Epsilon))を有しています。
4. 一つのビューマップにつき複数のラインセット(Line Set)を持つことができます。
5. ラインセットは、シーンを元にしたラインの中から、どのエッジタイプ(Edge Type)と選択(Selection)がレンダリングされるかをコントロールします。
6. 各ラインセットは一つのラインスタイル(Line Style)を使用します (複数のラインセットで共有することもできます)。
7. ラインスタイルはリンクしているラインセットの色、アルファ、幅、その他のレンダリング方法を Freestyle に伝えます。



Freestyle のビューマップとプロセスのブロック図

既知の制限と問題

- FreeStyle は Blender内部レンダラでのみ利用可能です。
- 多くのメモリが必要です。レンダーレイヤー内のすべてのメッシュオブジェクトは一度に読み込まれるためです。
- 面のあるメッシュオブジェクトのみ対応しています。以下のようなメッシュは無視されます。
 - ワイヤ(Wire)マテリアルのメッシュ面。
 - 完全に透過するマテリアルのメッシュ面。
- 半透明の面は不透明な面として処理されます。
- まだ面の交差によるエッジは検知されません。
- レイヤーマスク(Layer mask)は Freestyle では動作しません。
- Freestyle のレンダリング結果には Z 深度情報がありません。
- パノラマ(Panorama)カメラでは動作しません。

コアオプション



Freestyle コアオプション

Freestyle をプロパティ (Properties)エディタのレンダー (Render) プロパティで有効化すると、以下のオプションが表示されます。

ライン幅(Line Thickness)

ベースライン幅を定義する二つのモードがあります。

絶対(Absolute)

ライン幅がユーザが指定したピクセル数になります。デフォルト値は**1.0**。

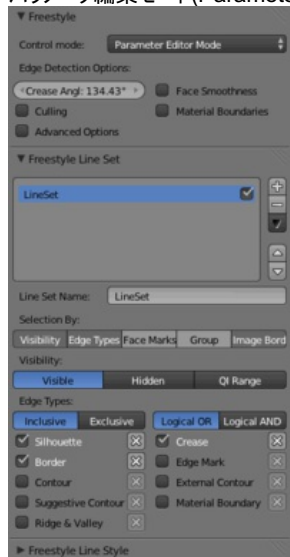
相対的(Relative)

単位ライン幅を**480**ピクセルに対する現在の画像の縦の解像度の比率でスケールリングします。例えば、画像の高さが**480**の場合、単位ライン幅が**1.0**ですが、**720**では**1.5**、**960**では**2.0**になります。

ライン幅(Line Thickness)

絶対(Absolute)のみ: ベースライン幅(ピクセル)。デフォルトは**1.0**。

パラメータ編集モード(Parameter Editor)



パラメータ編集モード (Parameter Editor)

Freestyle パラメータ編集モード(Parameter Editor)はユーザーフレンドリーなインターフェイスでラインセットとラインスタイルを設定できます。

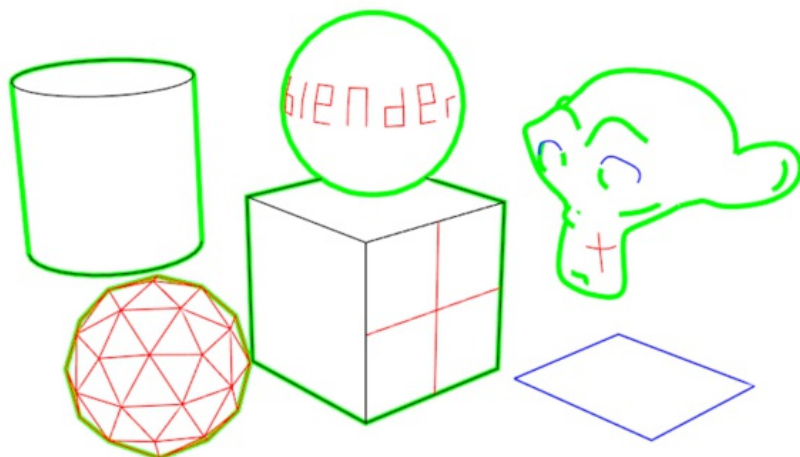
[ラインセット\(Line sets\)](#)は、Freestyle が探知したどのエッジ(縁)を 実際使用(レンダリング)するかをコントロールします。

[ラインスタイル\(Line styles\)](#) は、選択されたエッジをレンダリングする方法をコントロールします。

一つのビューマップ(View map)は(そしてレンダーレイヤーも)、複数のラインセットを持つことができ、各ラインセットは一つのラインスタイルにリンクされます。

ラインセット

ラインセットは Freestyle により検知されたライン(エッジ・縁)を取捨選択します。これらは設定された[ラインスタイル](#)を使用してさまざまな方法を通じてレンダリングされます。



いくつかの基本的なエッジタイプの例。LightBWK 氏作 ([File:EdgeType.zip](#))

可視性による選択

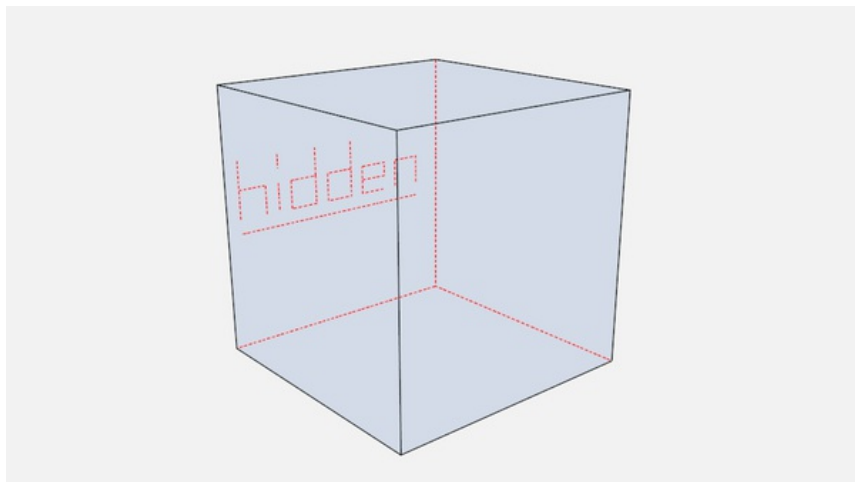
可視性(Visibility)によるエッジ選択には3つの方法があります。

可視(Visible)

どのサーフェスにも隠されていないラインのみレンダリングします。

不可視(Hidden)

少なくとも1つ以上のサーフェスに隠されたラインをレンダリングします。



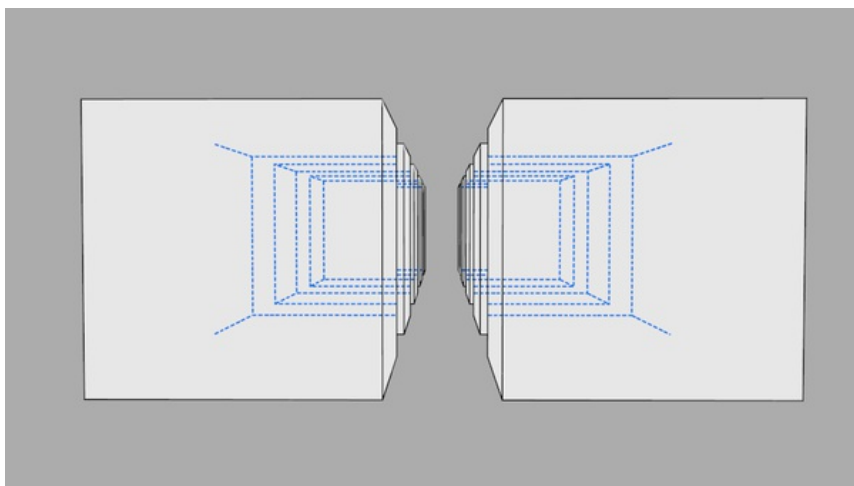
可視と不可視エッジの概念の実証例。LightBWK 氏作 ([File:HiddenCreaseEdgeMark.zip](#))

QI範囲(QI Range)

QIとは *Quantitative Invisibility*(*定量的不可視*)の略語です。与えられた範囲内の数のサーフェスによって隠されたラインがレンダリングされません。

開始(Start)と終了(End)

QI範囲(QI Range)のみ、ラインがレンダリングされる、遮蔽サーフェスの最小/最大数。



QI範囲の概念実証デモ。開始: 3、終了: 7。LightBWK 氏作([File:QI-Range.zip](#))

エッジタイプによる選択

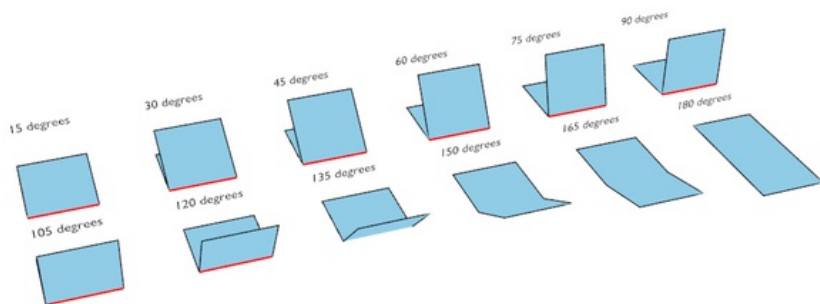
エッジタイプは形状からラインを選択する基本アルゴリズムです。パラメータ編集(Parameter Editor)を使用する場合、レンダー出力を得る為に最低一つのエッジタイプを選択しないといけません、複数のエッジタイプを一つのラインセット内で組み合わせることもできます。また、エッジタイプは横の×ボタンをクリックすることで、計算から除外することもできます。

シルエット(Silhouette)

閉じた形状のオブジェクトの周囲のシルエットを描画します。大抵の場合、有機的なオブジェクトに適しています(Suzanneや球など)が、箱などのシャープな縁には不向きです。開いている円柱や平面のような開いているメッシュオブジェクトはレンダリングできません。出力はKr 派生イプシロン(Kr Derivative Epsilon)ビューマップ設定(訳注: 詳細オプション内)の影響を受けます。

クリーズ(Crease)

隣り合う面の角度がビューマップ(▼Freestyle)のクリーズ角度(Crease Angle)設定より大きいエッジのみ表示します。



クリーズ角度(Crease Angle)が121°での概念実証例。LightBWK 氏作。
([File:CreaseAngle.zip](#))

ボーダー(Border)

ボーダーは縁が開いた/閉じていないメッシュ用で、開いた円柱では上下のエッジは開いており、平面では周囲すべてが開いています。Suzanneの眼窩は開いた縁です。開いた縁はすべてラインがレンダリングされます。メッシュ構造に依存します。

辺マーク(Edge Marks)

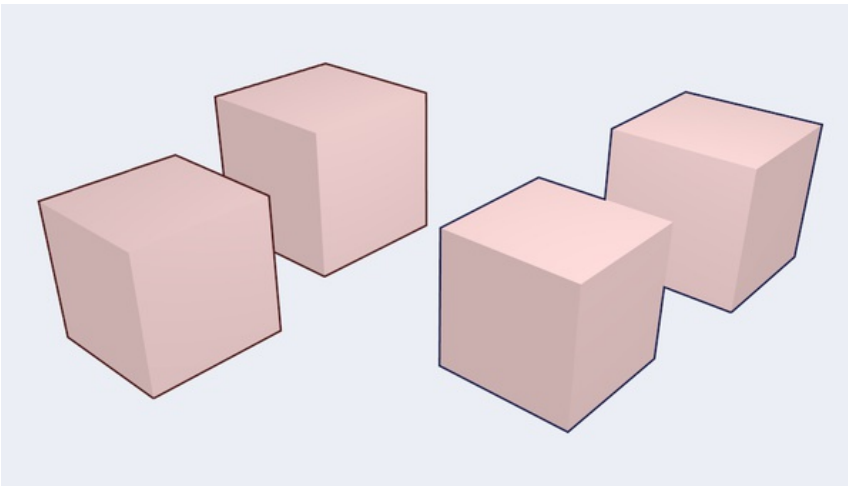
マークされた辺をレンダリングします。詳細は[後述](#)します。

輪郭(Contour)

外部の縁と内部の開いた境界を描画します。

外部輪郭(External Contour)

輪郭線を描画しますが、外側の縁のみです。



左の二つ: 輪郭(Contour)、右の二つ: 外部輪郭(External Contour)

示唆的輪郭(Suggestive Contour)

ビューがシフトした時、メッシュの輪郭を形成するラインを描画します。ビューマップ(訳注: ▼Freestyle→詳細オプション)のKr 派生イプシロン(Kr Derivative Epsilon)と、球半径(Sphere Radius)設定に依存します(詳細はこちらを参照してください(英文): [File:Manual-2.6-Render-Freestyle-PrincetonLineStyle.pdf](#))。

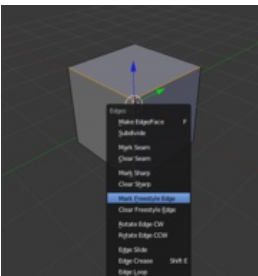
マテリアル境界(Material Boundary)

同じオブジェクト上の二つのマテリアルの出会う場所のラインを描画します。▼Freestyle のビューマップ設定内で有効化する必要があります。

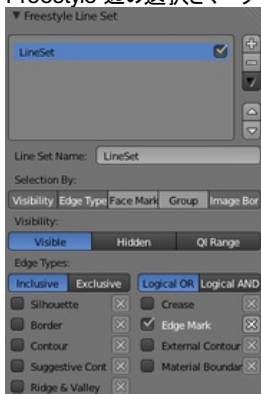
尾根と谷(Ridge & Valley)

尾根と谷を描画します。▼Freestyle→詳細オプション→球半径(Sphere Radius)設定に依存します。

辺マーク(Edge Marks)



Freestyle 辺の選択とマーク



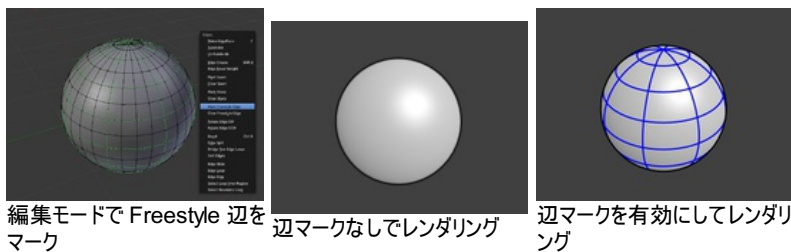
ラインセット(Line Set)パネルの辺マーク(Edge Mark)設定

UV展開用の「シーム」や辺分離モディファイア用の「シャープ」と同様、編集モード(Edit Mode)で、「Freestyle辺 (Freestyle Edges)」をマークできます。マークされた辺は辺マーク(Edge Mark)を ON にすればレンダリング可能になります。

以下のように行います。

- メッシュを選択し、⇄ Tabで編集モード(Edit Mode)に入ります。
- マークしたい辺(Edge)を選択します。
- CtrlEを押し、Freestyle辺をマーク(Mark Freestyle Edge)を選択します。

辺マーク(Edge marks)は特定のメッシュの辺に線を描画したい時に便利です。下記に辺マークの使用方法を解説します。



編集モードで Freestyle 辺をマーク

辺マークなしでレンダリング

辺マークを有効にしてレンダリング

左側の図は編集モード(Edit Mode)での球を示しています。緑の線が辺マークです。右側の図は辺マーク有効時のレンダリング画像です。

辺マーク有効時、すでにマーク済のラインは常にレンダリングされます。図には黒い輪郭線と辺マークにより作成された青い線が表示されています。

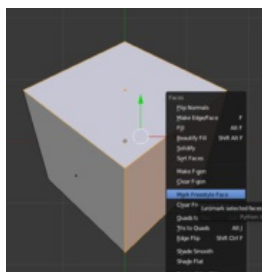
辺マークに適する物:

1. ほぼ平らな平面上のマークをレンダリングする必要があり、他のエッジタイプが何もラインを探知できない場合。
2. エッジレンダリングを完全にコントロールしたい場合。多くの場合四角い形状のエッジ用。
3. ベースメッシュのプレビュー用にベースメッシュ全体をマークしてレンダリングする場合。

辺マークに適さない物:

1. 外側の縁が丸い物(代わりに輪郭(Contour)や外部輪郭(External Contour)、シルエット(Silhouette)を使用してください)。

面マーク(Face Marks)による選択



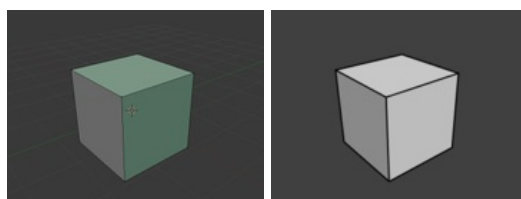
Freestyle 面のマーク

面マーク(Face Marks)を設定するには:

- メッシュを選択し、⇄ Tabで編集モード(Edit Mode)に入ります。
- マークしたい面(Face)を選択します。
- Ctrl+Fを押し、Freestyle面をマーク(Mark Freestyle Face)を選択します。

面マークは1つのメッシュの複数のエリアからラインを除去するのに便利です。

この例では、デフォルトの立方体の二つの面を左の画像のようにマークしています。右側は面マークを有効化していないレンダリング画像です。



マークした面

レンダリング出力



面マーク(Face mark)オプション

ラインの選択は包括的(Inclusion)と面マーク(Face Mark)オプションでコントロールできます。

包括的(Inclusive)/独占(Exclusive)

ラインセットで設定された面マークの状態にマッチする辺を含める、または排除します。

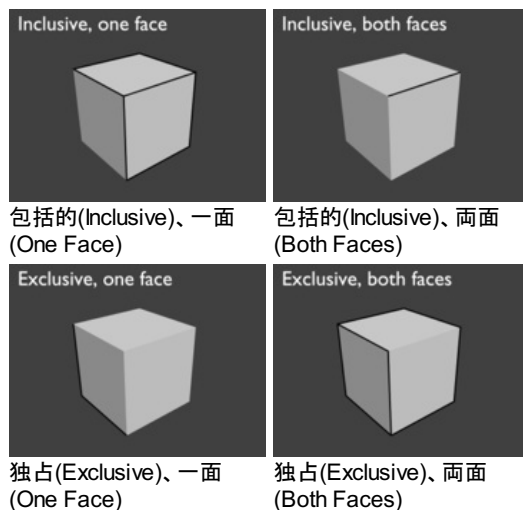
一面(One Face)

隣接する面の両方のうち一つがマークされている辺をすべて(非)選択します。

両面(Both Faces)

隣接する面の両方がマークされている辺をすべて(非)選択します。

以下の画像は組み合わせの結果を示しています。



グループによる選択

属するグループによって、ライン計算にオブジェクトを含めたり除外することができます。

グループ(Group)

使用するオブジェクトグループ名。

包括的(Inclusive)/独占(Exclusive)

このラインセットにこれらのオブジェクトのラインを含める、または除外します。

画像境界(Image Border)による選択

有効時、Freestyle はライン計算に画像境界内の形状のみを考慮します。これにより、レンダリング時間が短縮しますが、形状がカメラ視野を出入りする場合、問題が継続的に増加します。

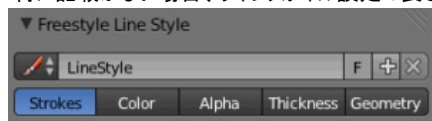
ラインスタイル & モディファイアー

Freestyle では、ラインスタイル(Line Style)設定が次の5つの面でラインセットの見た目を定義します：[ストローク\(Stroke\)](#)、[カラー\(color\)](#)、[アルファ\(alpha\)](#)、[厚さ\(thickness\)](#)、[ジオメトリ\(geometry\)](#)。これらすべてが多様なスタイルのレンダリングを可能にします(図面、ラフスケッチ、アニメ、東洋の水墨画など)。

たくさんのラインスタイルを好きだけ作成でき、さらにラインスタイル名の隣のドロップダウンメニューから選択することで、複数のラインセットにラインスタイルを再利用することもできます。

長さの単位

特に記載がない場合、ラインスタイル設定の長さはすべてピクセル単位です(相対的または絶対のいずれか、[コアオプション](#)で指定)。



ラインスタイル UI



ラインスタイルデモ [File:LineStyle.zip](#)

ストローク(Stroke)



ストロークラインスタイル(Stroke Line Style)

ストローク(Strokes)は最終的にレンダリングされるラインですが、これらをさらに調整することができます。例えば、しきい値による長い/短いストロークの削除や、複数のラインを一つのストロークへチェーン化、あるいは角度を元に一つのストロークを複数に分けたり、破線パターンなど…。

チェーン化(Chaining)

デフォルトでは、ラインセットから取得されたラインはすべて一緒にチェーン化されます。二つの基本的なチェーン化方法があります。

プレーン(Plain)

デフォルトのチェーン化方法。シンプルなチェーンを作成します。

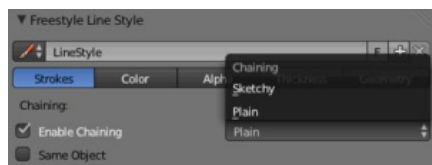
スケッチ(Sketchy)

このチェーン化オプションはスケッチ風の複数のストロークにより、特徴エッジのチェーンを生成できます。基本的には単一ストロークではなく、回数(Rounds)個のストロークを生成します。これはラインスタイルにランダムを使用するモディファイアーを使用している場合にのみ便利です！

回数(Rounds)

スケッチ風ストロークの回数を指定します。

また、チェーン化はラインごとにレンダリングをOFFにすることができます。ラインセットの正確な表示に依存するラインスタイルに便利です。



チェーン化(Chaining)

分割(Splitting)

Freestyle のチェーンラインは以下をチェックすることで分割できます。

マテリアル境界(Material Boundary)

あるマテリアルから他のマテリアルへと移る境界の、特徴エッジのチェーンを分割します。

最小2D角度(Min 2D Angle)と最大2D角度(Max 2D Angle)

2D角度が最小しきい値より上(または最大しきい値より下)の特徴エッジのチェーンを分割します。



分割(Splitting)

2D長さ(2D Length)

チェーンが与えられた値より長い時に分割します。

■(D)1/□(G)1/■(D)2/□(G)2/■(D)3/□(G)3

チェーンを破線パターンで分割します(「■(D)」は線(Dash)、「□(G)」は空白(Gap)。[下記](#)も参照してください)。

選択(Selection)



選択(Selection)

また、最小2D長さ(Min 2D Length)より長い、または最大2D長さ(Max 2D Length)より短いチェーンのみ選択(レンダリング)することもできます。

先端(Caps)

3つのタイプの中から線の先端を選ぶことができます。

直線(Butt)

平たい先端。ぴったり線の終端になります。



線の先端

丸め(Round)

線の終端を中心とする半円。

正方形(Square)

線の終端を中心とする正方形(よって円同様に、線の終端は計算値より少し延長されて描画されます)。

破線(Dashed Line)



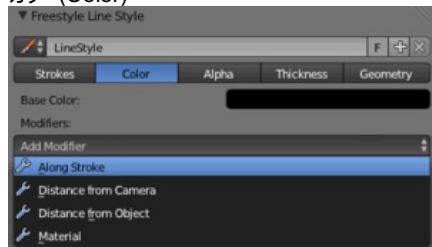
破線のUI

破線(Dashed Line)チェックボックスをONにすると、実線と空白の3つの組の長さを指定できます。実線(■、D)の値は線部分のストロークの長さ、空白(□、G)の値は二つの実線の間隔を指定します。

もし空白が0の場合、対応する実線はたとえ0以外の値でも無視されます。

実線が別々のストロークとして処理されるということは、色やアルファ、幅のモディファイアー、そしてライン先端を適用可能ということを意味します。

カラー(Color)



ラインスタイルカラー UI

このタブではストロークの色をコントロールします。

ベースカラー(Base Color)

このラインスタイルのベースカラー。

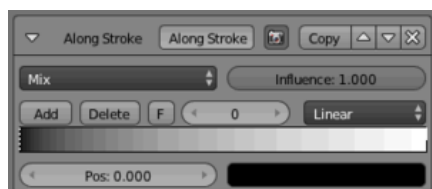
モディファイアー (Modifiers)

4つのカラーモディファイアーが利用可能で、通常の方法でベースカラーとミックスすることができます(詳細は[ミックス\(Mix\) コンポジットノード\(英文\)](#)を参照してください)。Blender の他のモディファイアースタック同様、上から下に適用されていきます。

影響(Influence)

このモディファイアーの結果が現在の色に影響する量。

ストローク追従 (Along Stroke)

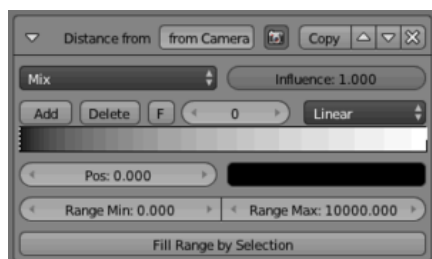


ラインスタイルカラーのストローク追従(Along Stroke)モディファイアー

ストローク追従(Along Stroke)モディファイアーは、各ストロークの長さに従ってマップされたカラーランプで、ベースカラーを新しい色に変更します。別の言い方をすると、カラーランプを各ストロークに沿って適用します。

設定は Blender 標準のカラーランプ(Color Ramp)以外ありません！

カメラからの距離 (Distance from Camera)



ラインスタイルカラーのカメラからの距離(Distance From Camera)モディファイアー

カメラからの距離(Distance from Camera)カラーモディファイアーはアクティブカメラからの距離をパラメーターとして、ベースカラーを与えられたカラーランプの新しい色に変更します。

最小範囲(Range Min)と最大範囲(Range Max)

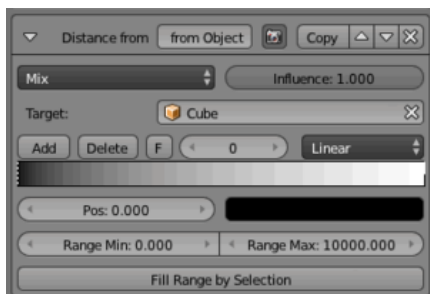
「カラーランプ」にマッピングする「カメラからの距離」の範囲。もしそのストロークの現在位置がアクティブカメラから最小範囲(Range Min)以下の距離であれば、カラーランプの最初の色を取り、逆にカメラから最大範囲(Range Max)以上離れていれば、カラーランプの最後の色を取ります。これらの値は現在のシーンの単位であり、ピクセルではありません！

選択から範囲設定(Fill Range by Selection)

最小・最大範囲の値を現在選択中のオブジェクトとカメラ間の距離から設定します。

他の設定は標準の Blender のカラーランプの物です。

オブジェクトからの距離 (Distance from Object)



ラインスタイルカラーのオブジェクトからの距離 (Distance From Object)モディファイアー

オブジェクトからの距離(Distance from Object)カラーモディファイアーは、与えられたオブジェクトとの距離をパラメーターとして使用し、ベースカラーを与えられたカラーランプの新しい色に変更します。

ターゲット(Target)
距離を測るオブジェクト。

最小範囲(Range Min)と最大範囲(Range Max)

「カラーランプ」にマッピングする「オブジェクトからの距離」の範囲。もしそのストロークの現在位置がターゲットから最小範囲(Range Min)以下の距離であれば、カラーランプの最初の色を取り、逆にターゲットから最大範囲(Range Max)以上離れていれば、カラーランプの最後の色を取ります。これらの値は現在のシーンの単位であり、ピクセルではありません！

選択から範囲設定(Fill Range by Selection)

最小・最大範囲の値を現在選択中のオブジェクトとターゲット間の距離から設定します。

他の設定は標準の Blender のカラーランプの物です。

マテリアル(Material)

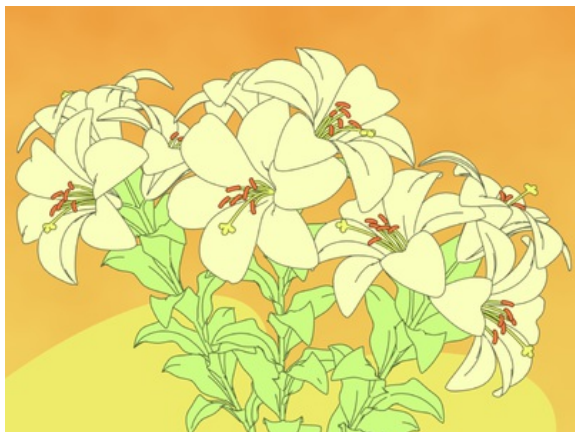


ラインスタイルカラーのマテリアル(Material)モディファイアー

マテリアル(Material) カラーモディファイアーはベースカラーをストロークの下にある現在のマテリアルから持ってきた色に変更します。

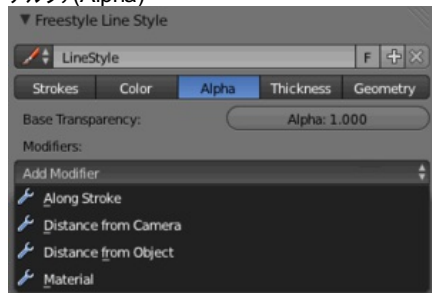
マテリアルの様々なプロパティを使用可能で、その多くは単一コンポーネント(要は白黒の結果)です。この場合、カラーランプでグレースケール値を色付きにマップするのに使用することもできます。

ストローク(Stroke)タブのマテリアル境界(Split by Material)オプションとともに使用していれば、ストロークに沿ってその結果がマテリアル間でブレなくなります。



マテリアルモディファイアーデモ。T.K.氏作[File:Lilies Color Material.zip](http://file.lilies-color-material.zip)

アルファ(Alpha)



ラインスタイルアルファ UI

このタブではストロークのアルファ(透過)をコントロールします。

ベース透過度(Base Transparency)

このラインスタイル用のベースアルファ。

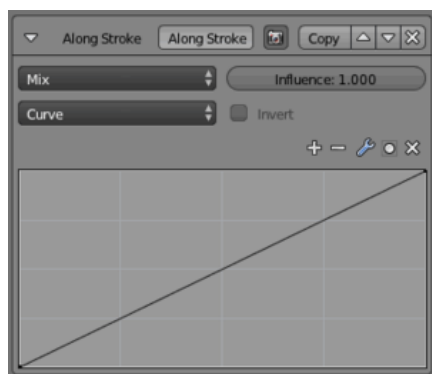
モディファイアー

4つのアルファモディファイアーが利用可能で、通常の方法のサブセットを使用しベースアルファとミックスできます(詳細は[ミックス\(Mix\) コンポジットノード\(英文\)](#)を参照してください)。Blender の他のモディファイアースタック同様、上から下に適用されていきます。

影響(Influence)

このモディファイアーの結果が現在の透過度に影響する量。

ストローク追従(Along Stroke)



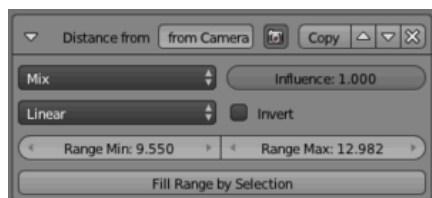
ラインスタイルアルファのストローク追従(Along Stroke)モディファイアー

ストローク追従(Along Stroke)モディファイアーは、各ストロークの長さに従ってマップされたリニア(Linear)のグラデーション、またはカスタムカーブ(Curve)で、ベースアルファを新しい物に変更します。別の言い方をすると、選択されたグラデーションを各ストロークに沿って適用します。

マッピング(Mapping)

リニア(Linear)グラデーション(**0.0**から**1.0**。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。

カメラからの距離(Distance from Camera)



ラインスタイルアルファのカメラからの距離(Distance From Camera)モディファイアー

カメラからの距離(Distance from Camera)モディファイアーはアクティブカメラからの距離をパラメーターとして使用し、ベースアルファをリニア(Linear)のグラデーション、またはカスタムカーブ(Curve)の新しいアルファに変更します。

マッピング(Mapping)

リニア(Linear)グラデーション(**0.0**から**1.0**。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。

最小範囲(Range Min)と最大範囲(Range Max)

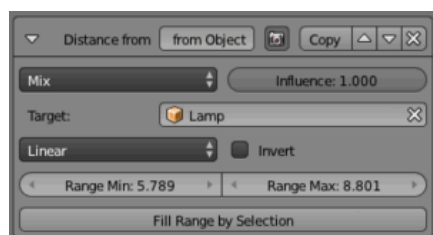
「マッピング内のアルファ」にマッピングする「カメラからの距離」の範囲。もしそのストロークの現在位置がアクティブカメラから最小範囲(Range

Min)以下の距離であれば、マッピングの最初のアルファを取り、逆にカメラから最大範囲(Range Max)以上離れていれば、マッピングの最後のアルファを取ります。これらの値は現在のシーンの単位であり、ピクセルではありません！

選択から範囲設定(Fill Range by Selection)

最小・最大範囲の値を現在選択中のオブジェクトとカメラ間の距離から設定します。

オブジェクトからの距離(Distance from Object)



ラインスタイルアルファのオブジェクトからの距離(Distance From Object)モディファイアー

オブジェクトからの距離(Distance from Object)モディファイアーは、与えられたオブジェクトとの距離をパラメーターとして使用し、ベースアルファをリニア(Linear)のグラデーション、またはカスタムカーブ(Curve)の新しいアルファに変更します。

ターゲット(Target)

距離を測るオブジェクト。

マッピング(Mapping)

リニア(Linear)グラデーション(0.0から1.0。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。

最小範囲(Range Min)と最大範囲(Range Max)

「マッピング内のアルファ」にマッピングする「オブジェクトからの距離」の範囲。もしそのストロークの現在位置がターゲットから最小範囲(Range Min)以下の距離であれば、マッピングの最初のアルファを取り、逆にターゲットから最大範囲(Range Max)以上離れていれば、マッピングの最後のアルファを取ります。これらの値は現在のシーンの単位であり、ピクセルではありません！

選択から範囲設定(Fill Range by Selection)

最小・最大範囲の値を現在選択中のオブジェクトとターゲット間の距離から設定します。

マテリアル(Material)



ラインスタイルアルファのマテリアル(Material)モディファイアー

マテリアル(Material)モディファイアーはベースアルファをストロークの下にある現在のマテリアルから持ってきた色に変更します。

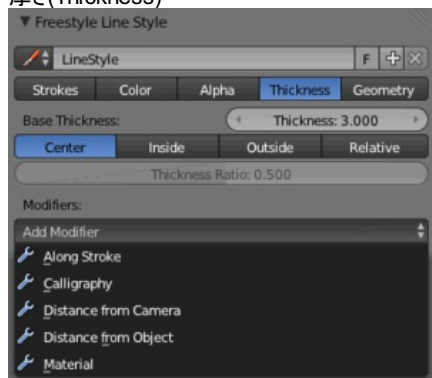
マテリアルの様々なプロパティを使用可能で、いくつかはマルチコンポーネント(RGBの結果など)です。この場合、中間値が使用されます。

マッピング(Mapping)

リニア(Linear)グラデーション(0.0から1.0。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。反転オプションなしのリニアグラデーションは「何もしない」と同じであることに注意してください。これはマテリアルの元の値がすでに[0.0, 1.0]の範囲であるためです...

ストローク(Stroke)タブのマテリアル境界(Split by Material)オプションとともに使用していれば、ストロークに沿ったマテリアル間でその結果がぼやけなくなります。

厚さ(Thickness)



ラインスタイル厚さ(Thickness)UI

このタブでストロークの幅をコントロールできます。

ベース幅(Base Thickness)

このラインスタイルのベース幅。

幅の位置(Thickness Position)

オリジナル(バックボーン)ストローク形状からのストローク幅の位置をコントロールします。以下の4つが選択できます。

中心(Center)

ストローク形状の左右が均等になるよう幅を分割します。

内部(Inside)

オブジェクト境界の内側にストロークが描画されます。

外側(Outside)

オブジェクト境界の外側にストロークが描画されます。

相対(Relative)

下の幅の割合(Thickness Ratio)内に**0.0**(内側)から**1.0**(外側)の範囲の数値で相対位置を指定します。

オブジェクト境界に関するエッジタイプ用であるため、幅の位置オプションはシルエット(Silhouette)とボーダー(Border)エッジタイプのストロークにのみ適用されます。他のエッジタイプのストロークは常に中心(Center)オプションで描画されます。

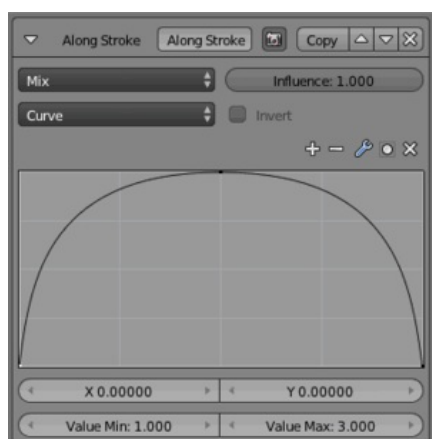
モディファイアー

5つの厚さモディファイアーが利用可能で、通常の演算方法のサブセットを使用し、ベース幅とミックスできます(詳細は[ミックス\(Mix\) コンポジットノード\(英文\)](#)を参照してください)。Blender の他のモディファイアースタック同様、上から下に適用されていきます。

影響(Influence)

このモディファイアーの結果が現在の幅に影響する量。

ストローク追従 (Along Stroke)



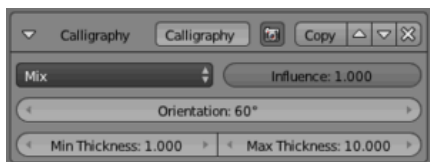
ラインスタイル厚さのストローク追従(Along Stroke)モディファイアー

ストローク追従(Along Stroke)モディファイアーは、各ストロークの長さによってマップされたリニア(Linear)のグラデーション、またはカスタムカーブ(Curve)で、ベース幅を新しい物に変更します。別の言い方をすると、選択されたグラデーションを各ストロークに沿って適用します。

マッピング(Mapping)

リニア(Linear)グラデーション(**0.0**から**1.0**。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。

カリグラフィー(Calligraphy)



ラインスタイル厚さのカリグラフィ(Calligraphy)モディファイアー

カリグラフィ(Calligraphy)モディファイアーはカリグラフィ用の幅広ペンを模倣します。ストロークの方向を元に、違う幅を生成します。

座標系(Orientation)

絵の縦軸からの仮想描画ツールの角度(方向)。例えば**0.0°**でペンが縦軸に平行になり、縦軸では最大幅のストロークに、横軸では最小幅のストロークになります。

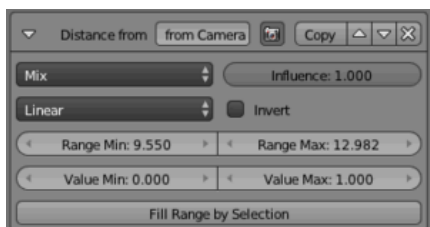
最小幅(Min Thickness)と最大幅(Max Thickness)

生成される最小と最大の幅(上述どおり、最小はストロークの方向が前述の方向に垂直な時で、最大は平行になった時です)。



カリグラフィ(Calligraphy)モディファイアーデモ。T.K.氏作[File:ToyCar Calligraphy.zip](http://file.toycar-calligraphy.zip)

カメラからの距離(Distance from Camera)



ラインスタイル厚さのカメラからの距離(Distance From Camera)モディファイアー

カメラからの距離(Distance from Camera)モディファイアーはアクティブカメラからの距離をパラメーターとして使用し、ベース幅をリニア(Linear)のグラデーション、またはカスタムカーブ(Curve)による新しい幅に変更します。

マッピング(Mapping)

リニア(Linear)グラデーション(**0.0**から**1.0**。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。

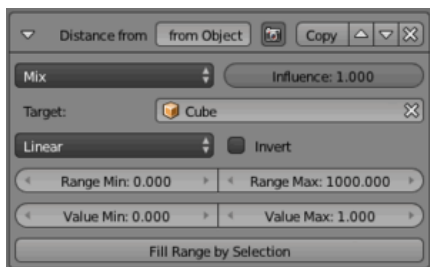
最小範囲(Range Min)と最大範囲(Range Max)

「マッピング内の幅」にマッピングする「カメラからの距離」の範囲。もしそのストロークの現在位置がアクティブカメラから最小範囲(Range Min)以下の距離であれば、マッピングの最初の幅を取り、逆にカメラから最大範囲(Range Max)以上離れていれば、マッピングの最後の幅を取ります。これらの値は現在のシーンの単位であり、ピクセルではありません！

選択から範囲設定(Fill Range by Selection)

最小・最大範囲の値を現在選択中のオブジェクトとカメラ間の距離から設定します。

オブジェクトからの距離(Distance from Object)



ラインスタイル厚さのオブジェクトからの距離 (Distance From Object)モディファイアー

オブジェクトからの距離(Distance from Object)モディファイアーは、与えられたオブジェクトとの距離をパラメーターとして使用し、ベース幅をリニア(Linear)のグラデーション、またはカスタムカーブ(Curve)の新しい幅に変更します。

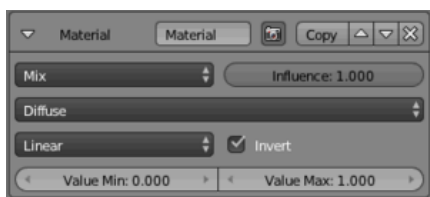
ターゲット(Target)
距離を測るオブジェクト。

マッピング(Mapping)
リニア(Linear)グラデーション(0.0から1.0。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。

最小範囲(Range Min)と最大範囲(Range Max)
「マッピング内の幅」にマッピングする「ターゲットからの距離」の範囲。もしそのストロークの現在位置がターゲットから最小範囲(Range Min)以下の距離であれば、マッピングの最初の幅を取り、逆にターゲットから最大範囲(Range Max)以上離れていれば、マッピングの最後の幅を取ります。これらの値は現在のシーンの単位であり、ピクセルではありません！

選択から範囲設定(Fill Range by Selection)
最小・最大範囲の値を現在選択中のオブジェクトとターゲット間の距離から設定します。

マテリアル(Material)



ラインスタイル厚さのマテリアル(Material)モディファイアー

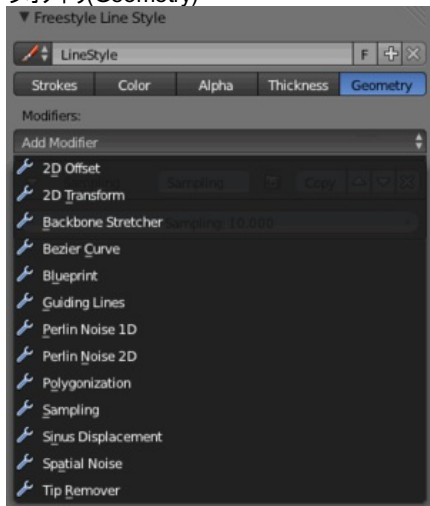
マテリアル(Material)モディファイアーはベース幅をストロークの下にある現在のマテリアルから持ってきた物で変更します。

マテリアルの様々なプロパティを使用可能で、いくつかはマルチコンポーネント(RGBの結果など)です。この場合、中間値が使用されます。

マッピング(Mapping)
リニア(Linear)グラデーション(0.0から1.0。反転(Invert)オプションで反転)、またはカスタムマッピングカーブ(Curve)のいずれか。反転オプションなしのリニアグラデーションは「何もしない」と同じであることに注意してください。これはマテリアルの元の値がすでに[0.0, 1.0]の範囲であるためです...

ストローク(Stroke)タブのマテリアル境界(Split by Material)オプションとともに使用していれば、ストロークに沿ったマテリアル間でその結果がぼやけなくなります。

ジオメトリ(Geometry)



ラインスタイル形状(Geometry)の UI 全体

このタブではストロークの形状をコントロールできます。

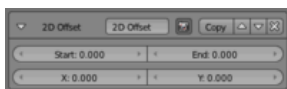
モディファイアー

13つの形状モディファイアーが利用可能です。これらのモディファイアーにはミックスや影響設定はなく、常にストローク形状に完全に適用されます(オブジェクトのモディファイアーと同じ方法です)。これらは Freestyle ラインセットの二次元のストローク結果を取り、様々な方法で変形します。

Blender の他のモディファイアースタック同様、上から下に適用されていきます。

2Dオフセット(2D Offset)

The 2Dオフセット(2D Offset)モディファイアーは、二次元のオフセットをストロークのバックボーン形状に加算します。独立したオプション/エフェクトのセットを二つ持ちます。



ラインスタイル形状の2Dオフセット(2D Offset)モディファイアー

開始(Start)と終了(End)

これら二つのオプションは与えられたオフセット量をストローク開始(終了)点で、その(2D)ノーマルに従い加算します。

エフェクトはストローク全体にブレンドされるため、例えば開始(Start)のみを50にセットした場合、ストローク開始がその法線方向に50ピクセル移動し、ストロークの真ん中ではその時の法線方向に25ピクセル移動、終端では移動しません。

XとY

これらの二つのオプションは単に一定の水平と垂直オフセットをストローク全体に加算します。

2Dトランスフォーム(2D Transform)



ラインスタイルジオメトリ2Dトランスフォームモディファイアー

2Dトランスフォーム(2D Transform)モディファイアーは二次元のスケールリングや回転をストロークのバックボーン形状に適用します。スケールは回転前に適用されます。

2Dトランスフォームの中心(ピボットポイント)は以下が利用できます。

ストローク中心(Stroke Center)

ストロークの中間点。

ストローク開始(Stroke Start)

ストロークの開始点。

ストローク終了(Stroke End)

ストロークの終点。

ストロークポイントパラメーター(Stroke Point Parameter)

ストロークポイントパラメーター(Stroke Point Parameter)係数は、ストローク上のピボットポイントの位置を操作します(0.0で開始点、1.0で終点)。

絶対2Dポイント(Absolute 2D Point)

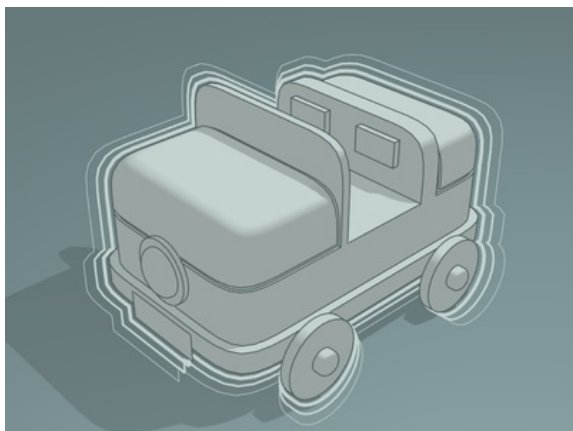
ピボットX(Pivot X)とピボットY(Pivot Y)で最終レンダーでのピボット位置を設定できます(左下が原点)。注意:現在、「実際の」レンダリングサイズ、つまり解像度と解像度のパーセンテージを考慮する必要があります!

拡大X(Scale X)と拡大Y(Scale Y)

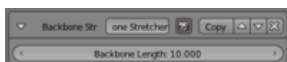
各軸に対するスケーリング係数。

回転角度(Rotation Angle)

回転角度!



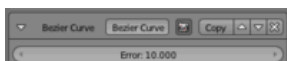
2Dトランスフォームモディファイアー [File:Toycar Three Contours.zip](#)

バックボーン拡大(Backbone Stretcher)

ラインスタイルジオメトリのバックボーン拡大モディファイアー

バックボーン伸縮(Backbone Stretcher)モディファイアーは、ストロークの開始点と終点を(長さを加えて)伸縮します。

バックボーン長さ(Backbone Length)
ストローク終点に加算する長さ。

ベジェカーブ(Bezier Curve)

ラインスタイルジオメトリのベジェカーブモディファイアー

ベジェカーブ(Bezier Curve)モディファイアーは、ストロークを近似したベジェに置き換えます。

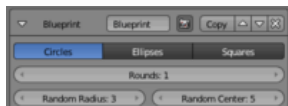
エラー(Error)

ベジェカーブと元のストロークの間に許容される最大距離。



ベジェカーブモディファイアーデモ。T.K.氏作 [File:Toycar bezier.zip](#)

設計図(Blueprint)



ラインスタイルジオメトリの設計図モディファイアー

設計図(Blueprint)モディファイアーは設計図のように、円や楕円、四角形の輪郭を使用したストロークを生成します。設計図はフリーハンド描画の最初に描かれた線を参照し、オブジェクトのシルエットを円や楕円、四角形のようなシンプルな形状で捉えます。

形状(Shape)

どのベース形状をこの設計図に使用するか。円(Circles)、楕円(Ellipses)、四角形(Squares)のいずれか。

回数(Rounds)

ペンが同じストロークを複数回描くように、生成する回数を指定します(要はこのプロセスを何回繰り返すか)。

範囲のランダム性(Random Radius)と中心のランダム性(Random Center)

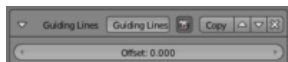
円(Circles)と楕円(Ellipses)形状用。関連する状況において毎回ランダム性を加えます。2回以上では前回は正確に繰り返すだけになるため、通常ランダム性がないと意味がありません。

バックボーン長さ(Backbone Length)とランダムバックボーン(Random Backbone)

四角形(Squares)形状用。前者は生成された四角形の各辺に長さを追加します(後者のパラメーターからも影響を受けます)。後者は四角形にランダムさを与えます。

ちなみに、小さいストロークによってできたノイズの回避にストローク(Strokes)設定の最小2D長さ(Min 2D Length)機能が非常に便利です。

ガイドライン(Guiding Lines)



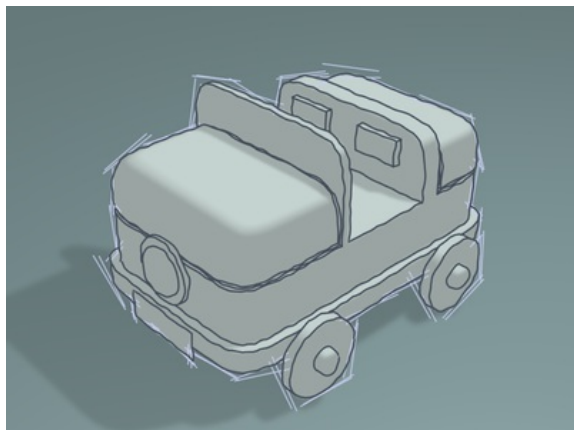
ラインスタイルジオメトリのガイドラインモディファイアー

ガイドライン(Guiding Lines)モディファイアーはストロークを両端を結んだ直線で置き換えます。

オフセット(Offset)

新しくまっすぐなストロークが作成される前の、オリジナルのストロークの始点と終点へのオフセット。

短いストロークは直線で近似しやすいため、このモディファイアーは十分に短いストロークの方が適切な結果が得られます。そのため、このモディファイアーはストローク(Strokes)パネルの分割オプション(2D角度(2D angle)や2D長さ(2D length)など)と一緒に使用することをおすすめします。



ガイドラインモディファイアーデモ。T.K.氏作 [File:Toycar Guiding Line.zip](#)

パーリンノイズ1D(Perlin Noise 1D)



ラインスタイルジオメトリのパーリンノイズ1Dモディファイアー

パーリンノイズ1D(Perlin Noise 1D)モディファイアーは1次元のパーリンノイズをストロークに加えます。

頻度(Frequency)

ノイズの密度(ストロークに対するスケールのようなもの)。

振幅(Amplitude)

ノイズがストロークを角度(Angle)の方向に歪ませる量。

シード(Seed)

乱数生成のシード(同じストロークに同じシードを与えると常に同じ結果になります)。

オクターブ(Octaves)

ノイズの「ディテールレベル」。

角度(Angle)

ノイズが適用される方向 (**0.0°** で完全に水平になります)。

パーリンノイズ2D(Perlin Noise 2D)



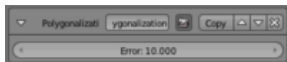
ラインスタイルジオメトリのパーリンノイズ2Dモディファイアー

パーリンノイズ2D(Perlin Noise 2D)モディファイアーは1次元のパーリンノイズをストロークに加えます。

設定はパーリンノイズ1D(Perlin Noise 1D)モディファイアーと全く同じです。

TODO: これらの二つのモディファイアーの違いは？

ポリゴン化(Polygonization)



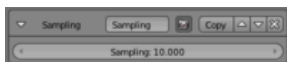
ラインスタイルジオメトリのポリゴン化モディファイアー

ポリゴン化(Polygonization)モディファイアーはストロークをできる限りシンプルにします(別の言い方をすれば、スムーズなストロークをカクカクした線分に変換します)。

エラー(Error)

新しくシンプル化されたストロークとオリジナルとの間にとれる最大距離(この値が大きい場合、よりカクカク/大まかな結果になります)。

サンプリング(Sampling)



ラインスタイルジオメトリのサンプリングモディファイアー

サンプリング(Sampling)モディファイアーは以下のモディファイアーのための設定、ストロークの精度を変更します。

サンプリング(Sampling)

この値を小さくするとストロークの精度が向上します。小さすぎるとレンダリング中の時間とメモリ消費量が増大することに注意してください！

Sinディスプレイスメント(Sinus Displacement)



ラインスタイルジオメトリのSinディスプレイスメントモディファイアー

The Sinディスプレイスメント(Sinus Displacement)モディファイアーは、正弦曲線の変位をストロークに加えます。

波長(Wavelength)

ストロークに伴う波の長さ。

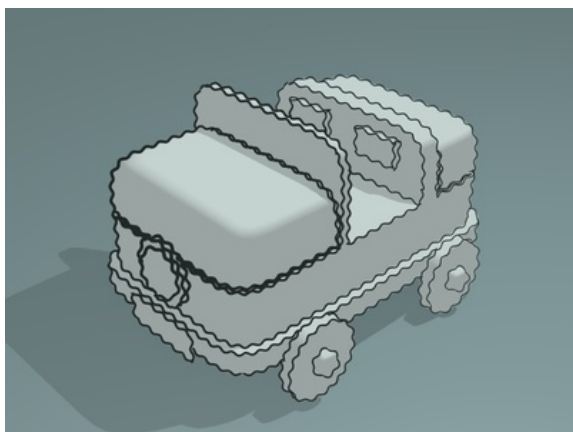
振幅(Amplitude)

ストローク全域での波の高さ。

初期ロール※ (Phase)

ストロークに伴う波を「移動」する「オフセット」。

(※ 後に「位相」あたりになる可能性があります -[Yamyam](#))



Sinディスプレイスメントモディファイアーデモ。T.K.氏作
[File:ToyCar Sinus.zip](#)

空間ノイズ(Spatial Noise)



ラインスタイルジオメトリの空間ノイズモディファイアー

空間ノイズ(Spatial Noise)モディファイアーはストロークに空間ノイズを加えます。

TODO: "空間ノイズ(spatial noise)"の定義!

振幅(Amplitude)

ノイズがストロークを歪める量。

拡大縮小(Scale)

ストロークに伴うノイズの幅。

オクターブ(Octaves)

ノイズのディテールレベル。

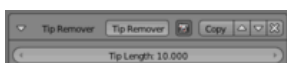
スムーズ(Smooth)

有効時、生成されたノイズをスムージングします。

純乱数(Pure Random)

無効時、次に生成された乱数値は前回の物に依存し、そうでない場合は完全に別個になります。この設定を無効にすることで、ストロークに伴うノイズがより「一貫した」物になります。

先端削除(Tip Remover)



ラインスタイルジオメトリの先端削除モディファイアー

先端削除 (Tip Remover) モディファイアーはストロークの最初と最後の部分を削除します。

先端の長さ(Tip Length)

両端から削除するストロークの長さ

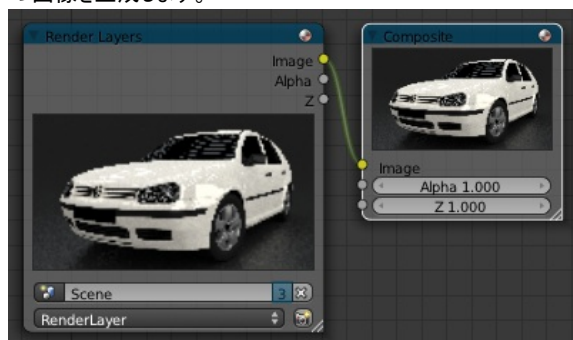
Composite Nodes

(コンポジットノード/ノード合成)

ノード合成を使うと画像(や動画)の合成と改良を同時に行えます。ノード合成を使えば撮影したフィルム同士をくっつけて全フレームを同時に着色することができます。一枚の画像でも完全な一本の映画でも、静的な方法か(動画の進行に応じて)時間経過で変化する方法で色調補正を行います。このように、ノード合成は映像クリップの合成にも、改良にも使えます。

用語: 画像(Image)

画像とは、単独の画像、連番画像の一枚の画像、動画の各フレームを指します。入力の種類に関わらず、ひとつのノードレイアウトは一度に一枚ずつ画像を生成します。



画像を加工するために、Blender に画像を取り込むノード、変更するノード、他の画像と混ぜ合わせるノード、保存するノードを使うことになるでしょう。

右の図はもっとも簡単なヌードルの例です。入力ノードがカメラビューを出力ノードに渡して、保存できるようにしています。

Nodes Concepts

Nodes

(ノード)

「ノード」とは、決まった処理を実行する個々のブロックを指します。ノードによってはひとつ以上の出力を持ちます。

概念上はノードには三つの基本タイプがあります。

- **Input Nodes**(入力ノード)

情報を作り出しますが、ノード自身の入力を持ちません。

例: *Render Layers*、*Value*、*RGB* の各ノード

- **Processing Nodes**: (処理ノード)

入力を *フィルター (filter)* または *変換 (transform)* し、ひとつ以上の出力を作ります。

例: *RGB Curves*、*Defocus*、*Vector Blur* の各ノード

- **Output Nodes**: (出力ノード)

入力を *消費* して何らかの有意義な結果を作ります。

例: *Composite* ノード (Blender が使う最終的な出力を決めます)、*Viewer* ノード (ソケットの出力を表示します)、*File Output* ノード

Noodles

(ヌードル)

ノードの基本思想は、ひとつ以上のノードの *出力* をひとつ以上のノードの *入力* につなげ、いくらでも入り組んだノードのネットワークを作れることです。そして各ノードのパラメータは目的に応じた最適値に設定できます。

このネットワークは「noodle/ヌードル」と呼ばれ、目的の結果を生むための文字通り *情報の流れ* を描きます。

Node Groups

(ノードグループ)

ノードグループを定義し、単一のノードのように利用することができます。

他ファイルからノードグループをリンクしたり、追加したりできます。

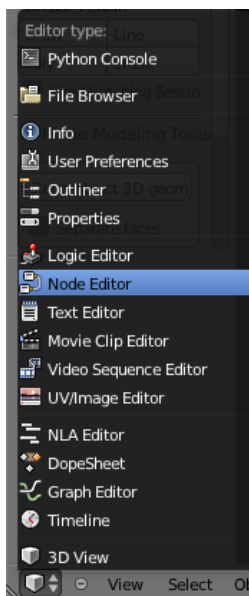
Accessing and Activating Nodes

(ノードの呼び出しと実行)

[ノードエディタ](#) を開いて *Image* アイコンをクリックし、*Composite Nodes* を有効にします。

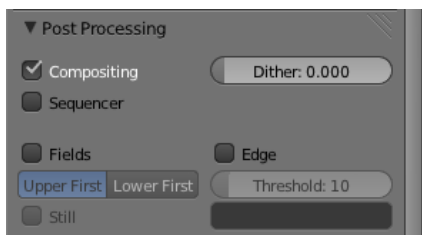


Composite Nodes が有効化されている Node エディタのヘッダー



Node Editor ウィンドウの
選択

合成用のノードを有効化するには、Use Nodes(ノードを使用) チェックボックスをクリックします。Blender はつながった二つのノードからなるデフォルトの開始マードルを作ります。



Composition Node の利用

この小さな図を使うためには、Blender がノード合成を使って画像合成するよう設定しなければなりません。具体的には、Render ボタン領域に切り替えて Post Processing タブ内にある **Compositing** ボタンを有効化します。これで Blender は合成ノード図を通過させることで最終的な画像の合成を行います。

さてノード図には最初のマードル、Composite 出力ノードにつなげられた RenderLayer 入力ノードがあります。ここから思う存分(物理メモリが許すまで) [さまざまな種類の合成ノード](#) を追加し、つないでいくことができます。

Examples

(例)

ノードを使うと、画像に対してほぼ何でも行えます。

ブルースクリーンの前で撮影した俳優のビデオや、オブジェクトに何かをさせてレンダリングしたものを、背景映像の上に重ね合わせるすることができます。両方を合成すると、合成された映像を作ることができます。

画像の雰囲気を変えることができます:

- 冷たい感じにするには、絵に薄い青色を足します
- フラッシュバックや記憶を表すとき、絵は柔らかく加工されるでしょう
- 嫌悪感やフラストレーションを表すのに、絵に薄い赤色を足すか、赤みを強くします。私が観た最高の例は映画「Sin City」です。
- 衝撃的な事件は鮮明にし、濃淡が強調されるかもしれません。
- 楽しい気分を表すにはお察しのとおり、鮮やかで明るい黄色を使います(すなわち赤と緑にわけられ、青はありません)。
- 画像に現実味を少し足すのに、cloud テクスチャを追加してちりや空気中のほこりを表現することがよくあります

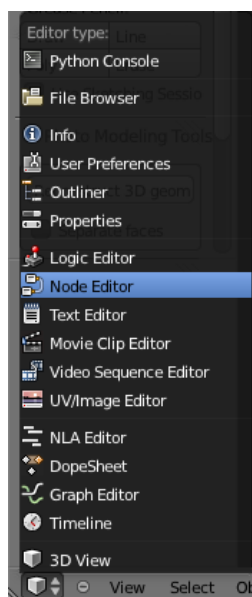
The Node Editor

(ノードエディタ)

このページでは通常のウィンドウと、ヘッダーメニューにあるオプション機能について説明します。Blender 内でノードを有効化する方法も説明します。

Accessing The Node Editor

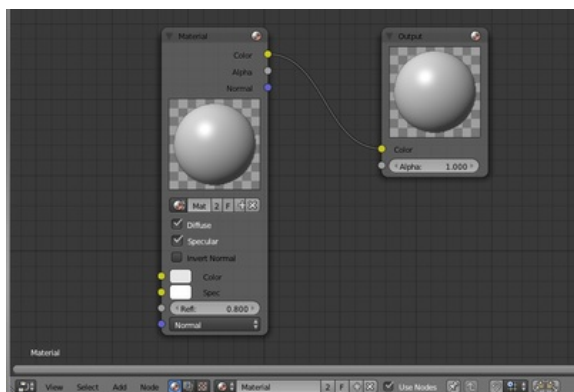
(ノードエディタを開く)



Node Editor ウィンドウの
選択

スクリーンレイアウトから「Compositing」を選ぶと、ノードエディタを含むウィンドウ配置になります。もしくは、任意のウィンドウのタイプを Node Editor に変えて、ノードエディタに入りましょう。図「Node Editor ウィンドウの選択」にあるように、ウィンドウタイプアイコンをクリックし、ポップアップリストから Node Editor を選びます。ノード図は非常に大きくなることもあるので、広いウィンドウを使うか、大きめのウィンドウを作ってください。ウィンドウには方眼紙様式の背景と、ヘッダーがあります。

blend ファイルにある各シーンは、複数のマテリアルノード図と「一つの」コンポジティングノード図を持つことができます。ノードエディタウィンドウにはノード種類セレクターで選択されている、いずれかのノード図が表示されます。



ノードエディタ

デフォルトではヘッダーの最初の表示は次のようになります:



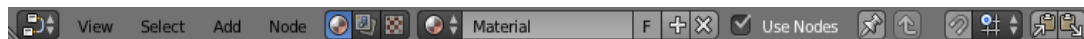
デフォルトのノードエディタのヘッダー

Activating Nodes

(ノードの有効化)

- どのノードを使うべきか？
 - マテリアル(シーダー)ノード図で作業したいときは、ノード種類セレクターの球体のアイコンを選択します(図「マテリアルノードが有効なノードエディタのヘッダー」)
 - テクスチャノード図で作業したいときは、ノード種類セレクターのチェッカー模様のアイコンを選択します(図「テクスチャノードが有効なノードエディタのヘッダー」)
 - コンポジティングノード図で作業したいときは、ノード種類セレクターの写真の重なったアイコンを選択します(図「コンポジティングノードが有効なノードエディタのヘッダー」)
- ノードを実際に有効化するには Use Nodes ボタンを押します

- 最初にマテリアル(シェーダー)、コンポジティングノード、テクスチャノード図のいずれかを選んだ時は、ノードエディタウィンドウにはすでに結合された入力ノードと出力ノードが置かれます。



マテリアルノードが有効なノードエディタのヘッダー



コンポジティングノードが有効なノードエディタのヘッダー



テクスチャノードが有効なノードエディタのヘッダー

Node Editor Window Actions

(ノードエディタウィンドウでの操作)

カーソルがウィンドウ内であれば、次のような Blender の標準的なホットキーやマウス操作を使えます:

ポップアップメニュー

- ⇧ ShiftA - 追加するノードを選ぶポップアップメニューを表示します。

削除

- X か Del - 選択ノードを削除します。

ボックス選択

- B - 範囲選択を開始します。カーソルを移動して LMB クリックし、ドラッグして複数ノードを選択します

接続の切断

- Ctrl LMB クリックしてドラッグ - アイコンが小さなナイフに変わり、ノード間を接続するスレッドをまたぐようにドラッグして切断します

取り消し (Undo)

- CtrlZ を押し忘れてボックス選択をしようとしたとき、便利です。

やり直し (Redo)

- CtrlY か Ctrl⇧ ShiftZ - ちょっと「Undo」しすぎたときに使えます :)

複数選択

- ⇧ Shift LMB か ⇧ Shift RMB - 複数ノードを選択します。

つかむ/移動

- G - 選択中のものを動かします

通常のウィンドウ操作

ノード図は非常にひげもじゃく(広くて入り組んでいるという意味です)になる可能性があります。ウィンドウの中身(ノード図)は他の Blender のウィンドウと同じように、MMB クリックしてドラッグすると平行移動することができます。Wheel を上下させるかテンキーの + NumPad/- NumPad を使うと、ズームイン/ズームアウトができます。通常のウィンドウと同じようにウィンドウのサイズを変えたり、結合させたりできます。

Node Editor Header

(ノードエディタのヘッダー)

At a glance

(項目一覧)

ウィンドウのヘッダーには次のオプションがあります:

- View (ビュー) - はっきり見たいものを選びます
- Select (選択) - 条件をつけてノードを選択します
- Add (追加) - ノードを追加します。種類別にまとめられています
- Node (ノード) - 選択したノードに対する操作です。頂点のメニューに似ています
- マテリアル/テクスチャ/コンポジティングノードセレクタ
- Use Nodes (ノードを使用) ボタン
- ピン止め ボタン
- 親ノードに移動 ボタン
- スナップ ボタン
- スナップノードセレクタ
- コピーノード ボタン
- ペーストノード ボタン
- Free Unused (未使用ノードを解放) ボタン
- Backdrop (背景) ボタン
- Auto Render (自動レンダリング) ボタン

Menus

(メニュー)

View, Select and Add

(ビュー、選択、追加)

基本機能を持つメニューです:

View(ビュー)

ウィンドウの視点を変えるメニューです。+ NumPad (ズームイン)、- NumPad (ズームアウト)、 \wedge Home (すべてを見渡せるまでズーム)といった標準的なキーボードショートカット、または同等のマウス操作の代わりになります。

Select(選択)

ノードやノードグループを選ぶメニューです。全選択 A やボックス選択の開始 B といったホットキーの入力と同じ操作ができます。

Add(追加)

ノードを追加するためのメニューです。追加できるノードの種類やその役割については、次の節をご覧ください。このメニューにある項目は、ウィンドウ内にカーソルがあるときに \diamond ShiftA を使うのと同じものです。

Node

(ノード)

Hide(隠す)

H - 選択ノードを隠します。メッシュ操作と同じ要領です。

Grouping(グループ作成)

最重要項目で、ユーザー定義のノードグループを作ることができます。作成したグループは編集したり、ノード図に追加したりできます。グループを作るには、対象ノードを選び、Node(ノード) » Make Group(グループ作成) を選ぶか、キーボードショートカット CtrlG を使います。名前はグループ内にある小さな入力欄で変更できます。緑色のヘッダーや、あなたのつけたクールな名前によってグループが簡単に見分けられます。

Delete(削除)

X - 選択ノードを削除します。

Duplicate(複製)

\diamond ShiftD - 元のものと同じ設定を持ち、元のものとはつながっていない複製を作ります。

Grab(つかむ/移動)

G - ノードをマウスで動かします。メッシュ操作と同じです

複製されたノード

ノードを複製すると、新しいノードは **ぴったり元のノードの上に重なります**。下にあるノードにスレッドがつながっているように見えたとしても、複製はつながっていないので、コントロールをいじっても画像には何も起きません。複製を動かして下にあるつながったノードを見えるようにする必要があります。

Buttons

(ボタン)

Material/Composite/Texture Selector

(Material/Composite/Texture セレクター)

ノードは操作対象によってグループ分けされています:

- [Material Nodes \(マテリアルノード\)](#) / [Blender 2.67 では Shader Nodes \(シェーダーノード\)](#) で作業するには球体のアイコンをクリックします。
- [Compositing nodes \(コンポジットングノード\)](#) で作業するには写真の重なったアイコンをクリックします。
- [Texture nodes \(テクスチャノード\)](#) で作業するには、チェッカー模様のアイコンをクリックします。

Use Nodes Button

(ノードを使用)

有効にするとマテリアル色の計算や最終的な画像のレンダリングにノード図を使うよう、レンダリングエンジンに指示します。チェックを外すとノード図は無視され、マテリアルタブやシーンの基本のレンダリングが行われます。

Use Pinned Button

(ピン止めを使用)

レンダーエンジンにピン止めされたノードツリーを使うよう指示します。

Go to Parent Button

(親に移動)

親ノードツリーに移動できます。

Snap Button

(スナップ)

ノードエディタウィンドウ内でのノードのスナップモードを切り替えます。

スナップノード要素 セレクタ

スナップ対象要素を次から選べます:

Grid (グリッド、デフォルト)

ノードエディタウィンドウのグリッドにスナップ

Node X (ノードX)

ノードの左右の縁にスナップ

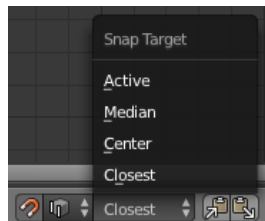
Node Y (ノードY)

ノードの上下の縁にスナップ

Node X/Y (ノードX/Y)

ノードの縁にスナップ

ノード縁へのスナップは次のスナップターゲットを考慮します:



Snap Target

Snap Target

どの部分をスナップさせるか選べます。

Closest (近接): 一番近い位置をスナップ

Center (中心): 中心をスナップ

Median (中点): 中点をスナップ

Active (アクティブ): アクティブ要素をスナップ

Copy Nodes Button

(コピーノードボタン)

選択ノードをクリップボードにコピーします。

Paste Nodes Button

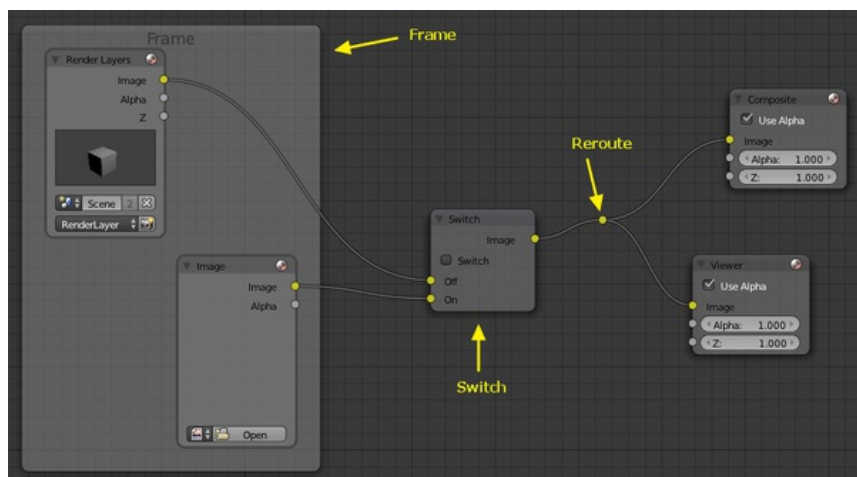
(ペーストノードボタン)

アクティブなノードツリーに、クリップボードからノードをペーストします。

Layout Nodes

(レイアウトノード)

Layout ノードはノードエディタウィンドウのノード配置の改善用に設計されています。Add (追加) » Layout (レイアウト) から利用できます。



レイアウトノードの例

Blender は次のレイアウトノードを備えます:

Frame (フレーム)

このノードは単に1箇所に複数ノードをまとめるために使います。

Reroute(リルート)

複数スレッドを枝状にしてノードエディタウィンドウのスペースを節約するためのものです。

Switch(スイッチ)

カラー(色)値の切り替えに使われるノードです。コンポジットノードでのみ使えます。

Free Unused Button

(未使用ノードを解放)

(コンポジットノード専用)非常に複雑なノード図で、未使用ノードのメモリ空間を解放します。お勧めです。

Backdrop

(背景)

(コンポジットノード専用)アクティブな Viewer ノードの出力を背景に使用します。有効にすると、ヘッダーとプロパティパネルにさらに設定が現れます:


Backdrop Channels(背景に使うチャンネル)

画像の Color、Color and Alpha または Alpha だけを表示します。

Zoom

背景画像の大きさを指定します

Offset

背景の画面に対する位置を変更します。Move ボタンを押すかショートカット Alt MMB  を使って実際に動かしながら変えることもできます。

Auto Render

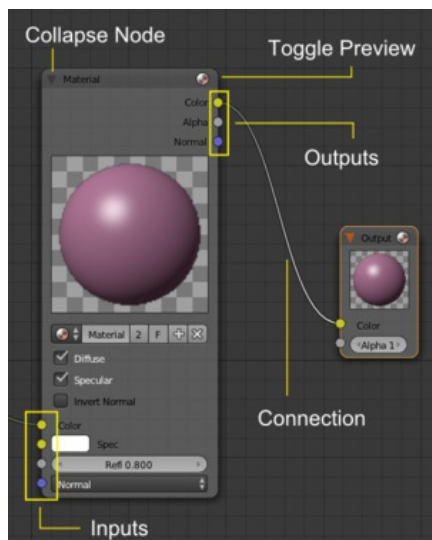
(自動レンダリング)

(コンポジットノード専用)3Dシーンで編集が行われると、変更のあったレイヤーの再レンダリングと合成を行います。

Node Controls

(ノードのコントロール)

このページではノードを操作するウィジェットを説明します。



ノードの主なコントロール

上から Collapse Node: 折り畳み Toggle

Preview: プレビュー切り替え Outputs: 出力

Connection: 接続 Inputs: 入力

タイトルバー

ノード名が表示され、折り畳み用のボタンがあります。

入力 (Input) ソケット

ノードの左側には入力ソケットがあります:

- 青色ソケットはベクトルを受け入れます
- 黄色ソケットは色を受け入れます
- 灰色ソケットは単独の値を受け入れます (例: アルファ)

出力 (Output) ソケット

ノードの右側には出力ソケットがあります:

- 青色ソケットはベクトルを作ります
- 黄色ソケットは色を作ります
- 灰色ソケットは単独の値を作ります (例: アルファ)

画像プレビュー

ノード内にはこのノードが出力する画像のプレビューやノードの動作を制御する曲線 (例えば RGB ノードにあるもの) を表示する領域があります。

ボタンとメニュー

画像プレビューの下には、ノードの動作を制御するボタンとメニューがあります。

スレッド

曲がった線は出力ソケットから入力ソケットへの接続を示しています。ソケットの種類は一致しなければなりません。

アクティブノードに関連した接続は見やすくなるようにハイライトされます。

Collapsing toggles

(折りたたみ表示の切り替え)

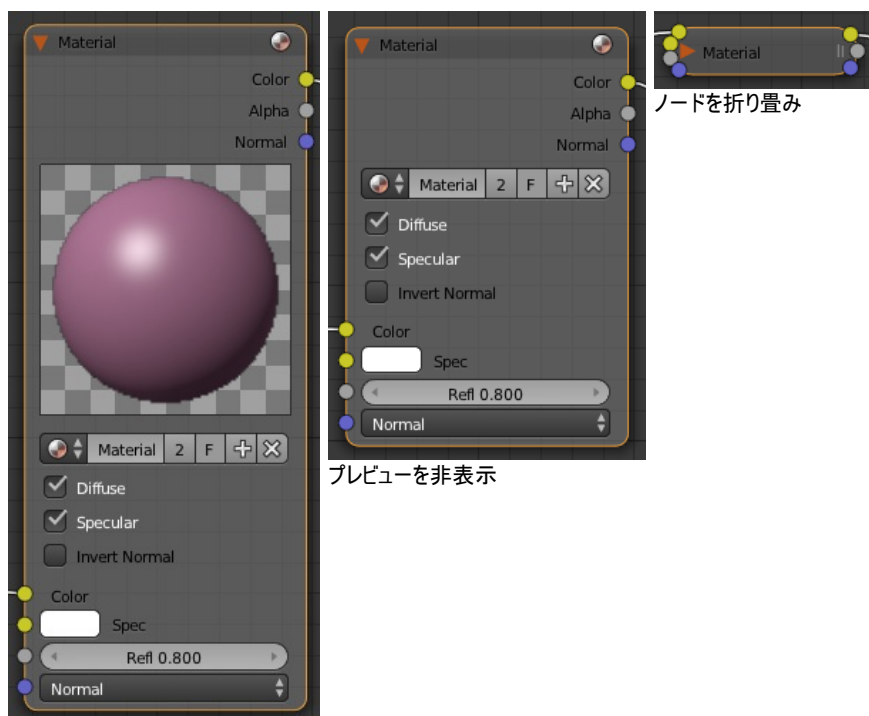
ノードの最上部には最大4つの外見に関わるコントロールがあります。クリックするとノードが表示する情報量に作用します。

ノード切り替え (▼▶)

左にある三角形はノードを折り畳み/展開します。

プレビュー画像の切り替え

タイトルバーの一番右にある球のボタンはプレビュー画像の表示/非表示を切り替えます。




すべて表示

プレビューを非表示

ノードを折り畳み

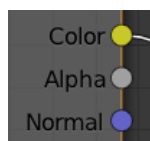
Sizing the node

(ノードのサイズ調整)

ノードの左右の端を LMB  クリックしてドラッグするとノード個別にサイズの調整を行えます。

Sockets

(ソケット)






ノードソケット

ノードウィンドウにある各ノードは色のついた小円形の「ソケット」(「コネクタ」ともよく呼ばれます)を持ち、入力データや出力データをつなげられます(図「ノードソケット」)。

ノード左側にあるソケットは *入力* を、右側にあるソケットは *出力* を表します。

便利のようにノードは送信/受信することを期待される情報の種類によって *色分け* されています。三つの色があります:

-  **黄色ソケット**
色 情報が入力に必要、またはノードから出力されることを示します。
-  **灰色ソケット**
値(数) 情報を示します。単独の数値かいわゆる「値マップ」を使えます(値マップは各点の値が明暗に反映されるようなグレースケールのマップだと考えることができます)。「値マップ」ソケットの入力として単独の値が使われると、マップのすべての点がこの値と同じ値になります。
一般的な用途: アルファマップやノード用の値オプション。
-  **青/紫色ソケット**
ベクトル/座標/法線 情報を示します。

ノード間は、コンバータ(*converter*)を使わない限りは、黄は黄に、灰は灰に、青は青に繋げなければなりません。コンバータについては後ほど説明します。

ノードの色名の隣にはソケット名が表示されています。常にはいえませんが、ソケットが何の情報を表すものか名前から想像できます。しかし必ずしも「従わねばならない」ものではありません。例えば Alpha と書かれた灰色ソケットから Material ノードの灰色の Reflection ソケットにつなぎ、結果を得ることができます。重要なのはこれが「灰色から灰色」への接続であるということです。

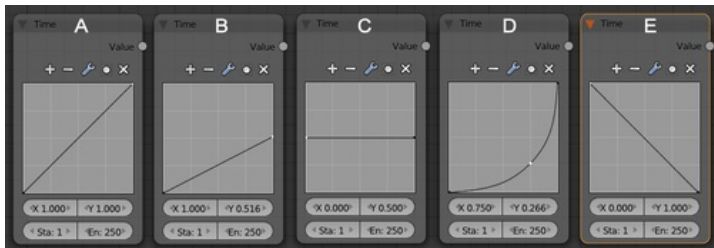
黄色(例: カラー画像)と灰色(例: グレースケール)をコンバータなしで組み合わせられる、例外があります。必要なら普通は Blender がコンバータを置くので、気軽に試してみてください。この後で説明する "Viewer" 出力ノードを使うと、動作するかどうかや、動作の様子を見ることができます。

Curves

(曲線)

ノードによっては入力値を出力値に変換するための曲線領域を持つものがあります。曲線の形状は制御点をクリックして動かすか、制御点を追加し

て編集できます。以下は例です:



Curve ノードの変形

どの曲線も傾き 1 の直線が始まります。曲線は始め両端に小さな黒い制御点を持っています。制御点を LMB クリックすれば選ぶことができ、制御点は白くなります。

曲線の変更は出力の生成のされ方に影響します。入力 X は普通 底の軸に交差して線形に(一定の間隔を持って)進みます。曲線にぶつかるまで上に進み、ぶつかれば右に進んで、X に対応する Y 出力を決めます。二つ目の例では、X が底に交差して 0 から 1.0 に進むに連れ、Y は 0.0 から 0.5 まで変化しています。3つ目の例では X は 0.0 から 1.0 に進みますが、Y は 0.5 に留まり続けています。したがって、上の図の曲線は次のような時間的効果を持ちます: **A** は影響なし、**B** は遅延させ、**C** は停止、**D** は加速、そして **E** は時間を逆向きにします。

「曲線」ウィジェットはどこでも使える Blender UI の組み込み機能で、与えた曲線データはこのウィジェットに渡されています。現在のところノードエディタと UV ウィンドウ内で使われています。

ウィジェットは水平方向に入力値をマップし、曲線の高さで示される新たな値を返します。

註釈: スクリーンショットでは曲線上の点が一「白く」なっていますが、深い意味はありません。単に、著者がチュートリアルを準備しているときに、たまたま最後に選択した点だったというだけです。ここで重要なのは曲線の形状であって、それを定義するための制御点の位置(または色)ではありません。

RGB Curves

(RGB 曲線)

ひとつのウィジェット内で複数の曲線を編集できます。その代表的な使用法が RGB Curves ノードで、このノードは「すべてを組み合わせた」結果である「色 (C)」を最初の曲線として持ち、R、G、B 要素それぞれの曲線も持ちます。4つの曲線すべてが有効で、「C」の曲線が最初に評価されます。

Selecting curve points

(曲線の制御点の選択)

- LMB は常にひとつの点を選び、残りの選択を解除します。
- Shift を押しながらかlickすると、選択範囲を増やすか少なくすることができます。

Editing curves

(曲線の編集)

- 点を LMB クリックしてドラッグすると点を動かせます
- 曲線を LMB クリックすると新たな点を追加します
- 他の点にちょうど重なるように動かすと一つにできます
- Shift を押しながらかlickするとグリッド単位にスナップします
- Ctrl LMB は点を追加します
- 選択した点を削除するには X アイコンをクリックします

Editing the view

(ビューの編集)

デフォルトではビューは 0.0 から 1.0 の範囲に固定されています。デフォルトの設定である clipping が有効なら、ビューの拡大やドラッグは行えません。丸いアイコンを押して clipping を無効にできます。

- 曲線の外側を LMB クリックしてドラッグするとビューを動かせます
- + と - のアイコンを使ってズームイン/アウトを行います

Special tools

(特別なツール)

レンチアイコンはビューのリセット、各点の補間の定義、曲線のリセットを行うメニューを表示します。

Using nodes

(ノードの利用)

Adding Nodes

(ノードの追加)

ノードは次のいずれかの方法でノードエディタウィンドウに追加します。

- ノードエディタのツールバーにある Add メニューをクリックし、目的のノードタイプを選びます
- カーソルをノードエディタウィンドウに置いた状態で、⇧ ShiftA を押し、ポップアップの Add メニューからノードを選びます


Arranging Nodes


(ノードの並べ替え)

ふつうはウィンドウ内でノードを並べ替えて、画像が左から右、上から下に流れるようにしてください。コントロールなどに影響しない部分をドラッグしてノードを動かします。ノードはほとんどどこでもクリックし、ドラッグすることができます。すべての接続は可能な限りベジエ曲線として再整形されます。

Connecting nodes

(ノードの接続)



ソケットを LMB -クリックしてドラッグすると、枝が伸び出てきます。これは「スレッド」と呼ばれます。

ドラッグしたまま他のノードの入カソケットにスレッドをつなげ、それから LMB  を離します。

この場合各出力のコピーがスレッドに沿って送信されます。一方、ひとつの入カソケットにはひとつのスレッドだけをつなげることができます。

Disconnecting nodes

(ノード接続の解除)

ソケット間のつながりを切るには、切りたいスレッドに近い何も無い空間を Ctrl LMB -クリックしてドラッグします。マウスポインタは小さなカッターナイフのアイコンになります。スレッドを横切るようにマウスを動かして、LMB  を離します。

Duplicating a node

(ノードの複製)

目的のノードをクリックして ⇧ ShiftD を押し、元のノードが見えるようにマウスを動かします。

わかった！

ノードを複製すると、新しいノードは *正確*にもとのノードの上に重なります。そのままにしておく(そうするのは本当に簡単です)、ふたつノードがあることが簡単にはわからなくなります！ 疑わしいときは、ノードをつかんで少し移動し、何か隠れているものがないか確かめてみてください。

Node Groups

(ノードグループ)

マテリアルとコンポジットのどちらのノードもグループ化できます。ノードをグループ化すれば、ノードエディタ上でノード回路網の配置を単純化でき、ヌードル(ノードの回路網)が扱いやすくなります。またノードグループ(.blend ファイル内)またはノードツリー(アペンド時)と呼ばれるものが作られます。

概念的には「グループ化」と「単体のノード」であるかのように扱えるノードの *セット* を定義することができます。ノードグループは同じ .blend ファイル内でも別の .blend ファイルからも、何度でも再利用できます。

例えばノードを使って作ったマテリアルを、別の .blend ファイルに単に追加することができます。しかし新たに作ったマテリアルから、既存のマテリアルノードの回路網から一部の分岐を使いたいときはどうでしょう？ 分岐を再作成することもできるでしょう。もしくはそのマテリアルを新たな .blend ファイルに追加して、使いたい分岐をカット&ペーストして新たなマテリアルに足すこともできるでしょう。どちらの方法もうまくいきますが、.blend ファイル間で作業するときはそれほど効率が良くありません。「被写界深度」のコンポジットノードの回路網を作り、別の .blend ファイルで使いたいときにはどうでしょう？ 一連の操作を正確に何十回も繰り返したいときは？ この場合も回路網の再作成はできますがそれほど効率が良くありません。マテリアルノードの分岐とコンポジットノード回路網のどちらにおいても、ノードグループを作って再利用するのが上手なやり方と言えるでしょう。

一旦定義したグループは、再利用可能なソフトウェア部品になり、ブラックボックス化します。定義を厳密に知らなくても、働きを知れば(何度でも好きなだけ)利用できます。グループは [Blender のライブラリと標準的な追加\(アペンド\)方法](#) を通じて利用可能な状態にできます。

Grouping Nodes

(ノードのグループ化)

パネル: [Node Editor](#)

メニュー: ◊ ShiftA » Group (グループ) » Make Group (グループ作成)

ノードグループを作るには、ノードエディタでグループに含めたいノードを選択し、CtrlG か ◊ ShiftA » Group » NodeGroup を使います。ノードグループは緑色のタイトルバーを持ちます。選択したノードはすべて最小化され、ノードグループ内に入ります。ノードグループのデフォルトの名前は *NodeGroup*、*NodeGroup.001* といったものです。ノードグループには名前を入力欄があり、クリックして変更できます。名前を意味のあるものに変えてください。別の .blend ファイルにノードグループをアペンドしたときに、Blender はマテリアルノードグループとコンポジットノードグループの区別をしません。したがってこの種類を見分けやすくする命名規則を作るのがよいでしょう。例えばマテリアルノードの分岐には *Mat_XXX*、コンポジットノードの回路網には *Cmp_XXX* と命名します。



グループに含むべきではないもの(すべてのタイプのノードエディタ)

覚えておきたいのは基本思想として、グループは簡単に再利用できる、自己完結したソフトウェア部品であるべきだということです。マテリアルノードグループは以下のノードを **含むべきではありません**：

Source nodes (ソースノード)

グループ内にソースノードを含めると、最終的にソースノードがグループ内とグループ外の新たなマテリアルノード回路内の二箇所にてきます。

ソースノードの例としてマテリアルノードの *Material* ノードや、コンポジットノードの *Render Layers* ノードがあります。

Output node (出力ノード)

グループ内に出力ノードを含めると、このグループからの出力ソケットが使えなくなります！

出力ノードの例としてマテリアルノードの *Output* ノードや、コンポジットノードの *Viewer* ノードがあります。

Editing Node Groups

(ノードグループの編集)

ノードグループを選択して ⇄ Tab を押すか、ノードグループ右肩の「ノード」アイコンをクリックすると、ノードがウィンドウ枠いっぱいになり、グループ内にある各ノードが表示されます。各ノードに対して移動や、コントロールの調整、スレッドのつなぎなおしなど、通常のエディタウィンドウの一部であるかのような操作ができます。外部にあるノードと直接スレッドをつなぐことはできず、グループノードの端にある外部ソケットを使う必要があります。ノードをグループに追加したりグループから削除するには、一旦グループを解除する必要があります。

Ungrouping Nodes

(ノードグループの解除)

AltG コマンドはグループを壊して、各ノードをエディタの作業領域に置きます。内部の接続は保持され、内部にあったノードを作業領域内にある外側のノードと接続できるようになります。

Appending Node Groups

(ノードグループの追加)

ノードツリーをグループとして .blend ファイルに一旦追加すれば、ノードエディタ内で ◊ ShiftA » Add (追加) » Group (グループ) を選び、追加したグループを選ぶとそのグループを利用できます。グループの「コントロールパネル」はノードグループ個別にあります。他のノードと同じように操作できます。

ノードの種類

ノードの種類を、機能の似たもの別にグループ分けします:

- [Input/入力 \(eng\)](#) - 画像や値をノード図に追加します
- [Output/出力 \(eng\)](#) - 処理中の結果を小さな画像として表示します
- [Color/色 \(eng\)](#) - 画像の色を操作します
- [Vector/ベクトル \(eng\)](#) - 画像の強度と反射を操作します
- [Filters/フィルター \(eng\)](#) - 隣接ピクセルに働きかけて、画像の質を高める処理を行います
- [Convertors/変換 \(eng\)](#) - 画像を映像要素ごとにわけるか、形式を変換します
- [Mattes/マット \(eng\)](#) - 画像の一部をマスクするためのマットを作ります
- [Distortion/歪み \(eng\)](#) - 画像の形を変えます
- [Groups/グループ](#) - ユーザーが定義した、ノードのグループです

Sequence Editing

(シーケンス/動画の編集)

モデリングやアニメーションの機能に加えて、Blender は完全に機能するビデオシーケンスエディタ (VSE) と、高度なノードベースのエディタ (これもビデオストリームを操作します) を備えています。[ノード合成 \(Compositing Nodes\)](#) は画像とビデオストリームのどちらもうまく扱い、きめの細かな画像処理をストリームに施すことができます。

高い概念レベルで処理を行い、動画制作工程の後半で使われる VSE は Blender に古くからあるもので、映像の断片であるストリップのセットを一度に操作します。典型的な作業工程では、Blender の機能の多くが協力しあいます:

- モデリングによりオブジェクトを構築
- マテリアルの割り当てと照明の導入によりオブジェクトに色をつける
- アニメーションによりオブジェクトに動きをつける
- カメラを使って映像レイヤーをレンダリングする
- コンポジットノードを使って:
 - 色調補正やシーン内の特殊効果の追加により画像を改良する
 - 画像を重ね合わせて合成画像シーケンス (ストリップ) にする
- VSE を使い、映像ストリップをまとめて一本の映画にする

Blender 内蔵の VSE は複数の映像チャンネルを組み合わせてたり、エフェクトを付け足したりできる、完全な動画編集システムです。この機能は初期の Blender からありました。VSE の演算処理は多くありませんが、強力な動画編集に使えます (Blender のアニメーションの力と組み合わせた場合は特に!)。さらにプラグインシステムを通じて拡張でき、無制限の数の画像操作を行えます。

VSE を使って複数の映像クリップを読み込んで隣り合わせに並べ (ときには重ね合わせて)、クリップ間の連結部にはフェードやトランジションを挿入できます。最後に音声トラックを追加して、動画シーケンスとタイミングを合わせることができます。VSE を使用すれば、完成した映像ができます。

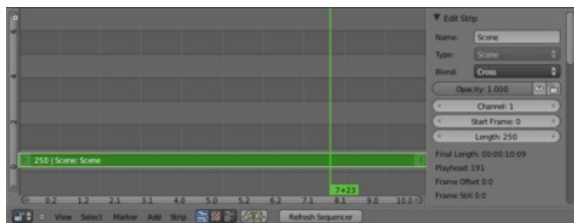
FFMPEG のサポート

FFMPEG を使った AVI/QuickTime Movie のエクスポートは現時点で (2.44以降) Linux と Windows 版でのみサポートしています。FFMPEG のサポートがあれば、動画といっしょに音声トラックを保存できます。

Overview of the Sequence Editor

(シーケンスエディタの概要)

Blender のビデオシーケンスエディタ(VSE)はビデオ映像編集用の柔軟性のあるツールです。映像の見直しやたくさんの動画シーケンスを貼り合わせるのに使います。シーケンスからシーケンスへのトランジション用の多数の組み込みみかプラグインのエフェクトを備え、プロ並みのビデオ向けの高度なハリウッド式のエフェクトを提供します。



Sequence 表示モードの VSE

「Video Editing」スクリーンレイアウトは、ビデオシーケンスエディタでの作業用です。ビデオシーケンスエディタはヘッダ(メニューとビューモードが表示されているところ)とワークスペースを持ち、ビューモードの一つとして動作します。マーカーメニューでは VSE 内にマーカーを追加できます。マーカーはアニメーションエディタ間で共有されます。[マーカー](#) をご覧ください。

シーケンサーのワークスペースにはチャンネルごとに横縞があり、ビデオストリップはいずれかの横縞(チャンネル)に入ります。各チャンネルの左端には番号が振られ、番号は 0 から始まります(この特別なチャンネルには何も入れられません!)。縦縞はもっと重要で、この話はこの後すぐに行います。X 方向には時間の単位にアニメーションの秒数かフレーム番号(CtrlT で選べます)が使われます(図では0.2秒から10.0秒が表示されています)。ズームキーかマウス操作で時間の拡大縮小ができます(詳しくはリファレンスをご覧ください)。



シーケンス出力が有効な状態

Render か Anim をクリックして画像か動画を生成すると、現在のフレーム/スクラブ範囲用に合成する画像をどれにするか選択できます:

- 現在のシーンレイヤーの結果
- シーケンスエディタのチャンネル0の結果
- コンポジションノードエディタのレンダーレイヤーの結果

ビデオシーケンサーはデフォルトで有効です。

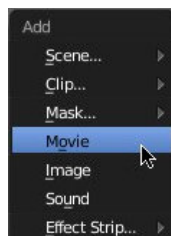
VSE を開いたまま普段レンダリングが行われるレンダーパネルを開き、アニメーションか画像をクリックすれば、シーンではなく VSE 用のクリップがレンダリングされます。

Using the Sequence Editor

(シーケンスエディタの利用)

Adding Strips

(ストリップの追加)



追加メニュー

Add(追加)メニューは VSE にコンテンツを追加するために使われるメインメニューです。操作は概ね、ストリップを追加し、特殊なトランジション効果のストリップを作り、Render(レンダー)コンテキストのアニメーションボタンを押してシーケンスを動かす、といったものになります。ヘッダーにある Add(追加)メニューを使うことも、シーケンスのワークスペース上にマウスカーソルを動かして ⇧ ShiftA を押すこともできます。

クリップが巨大になる可能性

3分の quicktime .mov ファイルは 140MB にまでなることがあります。高速LAN回線であっても読み込みに多少時間がかかります。少しの間何も起きなくても、コンピュータや Blender が動かなくなつたとは思えないほうがよいでしょう。

最初にクリップを追加しましょう:

- AVI 形式の動画クリップ (*.avi file)
- Apple QuickTime 形式の動画クリップ (*.mov)
- フレームの数だけ繰り返される単独の静止画像 (*.jpg、*.png など)
- 連番の振られた一連の画像 (*-0001.jpg、*-0002.jpg、*-0003.jpg など。画像形式は任意)
- あるディレクトリ内の1つ以上の画像
- .blend ファイル内のシーン

どれを使っても構いません。この中から自由に混ぜたり組み合わせたりできます。VSE では、ストリップは種類(動画、静止画、音声、シーン、エフェクトなど)によって色分けされます。

Add (追加) メニューではまず対象の種類を選びます。動画や画像、音声を選ぶと VSE ウィンドウがファイルブラウザに切り替わり、追加するものを選びます:

Add Movies or Images

(動画や画像の追加)

Add (追加) メニューで Movie (動画) を選んだ場合:

ファイルブラウザでフォルダをたどり、目的のファイルをクリックします。

複数ファイルを選択してまとめて追加できます。ファイル名の昇順にファイルの数だけストリップが追加されます。



Add Movie Strip (動画ストリップを追加) ボタンをクリックするとウィンドウは VSE に戻り、現在のカーソル位置からストリップが追加されます。

Image (画像) を選んだ場合:

ファイルブラウザでフォルダをたどり、目的のファイルをクリックします。

複数ファイルを選択すると、追加されるストリップにはファイル名の昇順に、1フレームごとに1画像が割り当てられます。画像種類は混在可能です (jpg、png、exr など)。

次のいずれかの方法で複数選択ができます:

- 範囲選択 (左クリックのあとドラッグ、または B)
- 全選択 (A)
- 追加選択 (⇧ Shift LMB  または ⇧ Shift RMB )

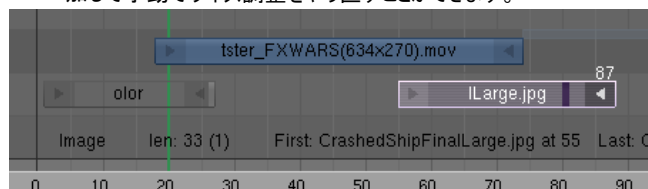
Add Image Strip (画像ストリップを追加) ボタンをクリックするとウィンドウは VSE に戻り、現在のカーソル位置からストリップが追加されます。

ストリップの拡大や縮小をしたいときは、ストリップの始めか終わりの矢印を RMB  クリックして、左右にドラッグします。マウスを動かすたびに矢印の位置を示すフレーム番号が更新されます。LMB  で変更を確定するか RMB  でキャンセルします。



サイズ混在時の対応

サイズの違う画像とサイズの違う出力をうまく処理するのは大変です。ピクセル視点で考えてみてください。入力画像とレンダリング出力の大きさが合わない場合も、VSE は画像が出力におさまるようにサイズを調整するようなことはありません。はみ出した画像の端は切り取られます。これを避けたい場合は Strip Input (ストリップ入力) パネルにある Crop (クロップ/トリミング) と Offset (オフセット) を使って、出力におさまる画像の範囲を移動したり選択したりしてください。クロップやオフセットを使うと自動スケールは無効になり、Transform (トランスフォーム) エフェクトを追加して手動でサイズ調整をやり直すことができます。



ワークスペースをスクロールすると情報チャンネルがあり(縦に見てチャンネル0の位置)、アクティブなストリップに関する役立つヒントが表示されます。上の例ではフレーム 1 から 25 までの Color ストリップに続いて動画ファイル、それから画像ストリップがあります。情報チャンネルは画像ストリップに関する役立つ情報を表示しており、画像ストリップの名前はストリップの表示では略されていますが、情報ストリップでは詳しく書かれています。

"9999 frames go by"

はい、これは99個の風船を歌った歌のわかりにくいダジャレでした。わたしたちは Blender がこんなに早くビデオ編集の主流に進出するとは予想してませんでした。当初画像シーケンスのファイル名(のフレーム番号部)に確保された桁数はあいにく4桁でした。これで400秒までのビデオ(US なら約5分)用の連番を準備できますが、Blender が動画を扱い始めたことにより、画像ストリップは重複しない4つの数字とそれ以外による5桁、つまり(10000-19999)や(20000-29999)などに分割する必要があります。いまのところ影響を受けるのは画像ストリップだけです。他のストリップはすべて、最大 300,000 フレームまでうまく動きます(約3時間です。ベン・ハーでも大丈夫!)。

コーデック

お使いのコンピュータに AVI ファイルをデコードできるコーデックが必要です。Blender はこの操作をしません。たとえば XviD コーデックが www.xvid.org から入手できます。

FFMPEG のサポート

FFMPEG のサポートされた Blender のビルドを使っている場合は、音声と動画ストリップをいっしょに読み込みます。音声トラックを含む動画を選んで追加したストリップは、音声とビデオチャンネルにわかれます。

Add Scene

(シーンの追加)

いま開いている .blend ファイルにあるシーンの、仮想的な画像出力を追加することもできます。ポップアップリストからシーンを選ぶとストリップが追加されます。ストリップの長さはそのシーン(VSE で同じシーンを操作していない限りは現在のシーンとは別のシーン)のアニメーション設定に基づいて決まります。

シーンストリップを追加するときに気をつけたいのは、VSE 画像プレビューモードでストリップを表示するためには、Blender がそのシーンをレンダリングす

必要があることです。複雑なシーンならばばく時間がかかるため、シーンを選んでストリップが表示されるまでに遅延があるかもしれません。遅延を抑えるにはレンダリングするレイヤーを少なくして、シーンのレンダリングを単純化してください。

(複数回のテストレンダリングとプレビューのどちらについても)シーンのレンダリングに余分な処理負荷がかかるのが問題になる場合は、ディスク容量が十分あれば、シーンを PNG のシーケンスとしてレンダリングし、シーンストリップの代わりに画像シーケンスストリップを使うことを検討してください。この手法は、透明度のアニメーションが使われた静止画でしかない、タイトルカードのようなオーバーレイでよく使われます。

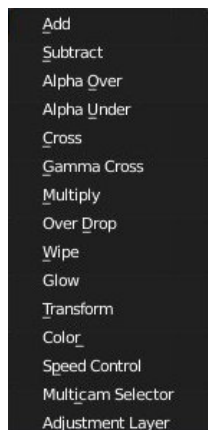
Add Audio

(音声の追加)

VSE ではスクラブして音を聴ける音声チャンネルを組み込めます。ビデオアニメーションを音声トラックに同期させたい場合(またはその逆)は音声トラックを追加してください。詳しくは [音声シーケンスの節](#) をご覧ください。(訳注:スクラブしたときデフォルトでは音声は再生されません。この機能を使うにはタイムラインのヘッダーメニューで Playback(再生) » Audio Scrubbing(オーディオスクラビング) にチェックを入れます)。

Adding Effects

(エフェクトの追加)



利用可能な組み込みエフェクト

右の図は Blender の備える組み込みエフェクトの一覧です。

次のページでは組み込みのエフェクトをひとつずつ説明しますが、どれも同じ方法で追加や操作ができます。エフェクトストリップを追加するには RMB クリックでベースストリップ(画像、動画、シーン)をひとつ選択します。Cross(クロス)トランジションエフェクトのような一部エフェクトでは、重ねあわせる二つ目のストリップを Shift RMB クリックする必要があります(エフェクトによります)。次に Add(追加) -> Effect(エフェクト)を開いて、ポップアップメニューから利用するエフェクトを選びます。この結果、元のストリップの上にエフェクトストリップが現れます。Color(カラー、後述)のような独立したエフェクトでは、ソースストリップの指定は不要です。

エフェクトストリップのほとんどは1つか2つのソースストリップに依存していて、フレーム位置や長さがソースストリップに左右されます。エフェクトストリップを直接動かすことはできず、動かすにはソースストリップを動かす必要があります。

二つのストリップの統合エフェクトやトランジションを間に入れる(あるいはコンポジット)エフェクトを使うためには、範囲選択(B)か Shift RMB クリックを使って二つを選択する必要があります。追加したエフェクトストリップで、二つのストリップの間は完全に埋まります。

AlphaOver のような一部エフェクトでは、ストリップを選ぶ順番が重要です。エフェクトストリップを他のストリップの入力、ソースストリップとして使って、一つずつエフェクトを重ねていくこともできます。

備考: 唯一の例外が Color(カラー)エフェクトです。ベースストリップに依存せず、どのストリップからも独立して追加や配置ができます。他のストリップと同じ操作で長さを変えられます。

エフェクトの変更

モード: シーケンス、エフェクトストリップ選択時

ホットキー: C

メニュー: Strip(ストリップ) » Change Effect(変更)

メニューから間違ったエフェクトを選んでしまったら、ストリップを選択して(RMB Strip(ストリップ) » Change Effect(変更) を選べばいつでもエフェクトを変更できます。もしくはエフェクトストリップを選んで C ("C")hange)を押せば、エフェクトの切り替えができます。

Adding Plugin Effects

(プラグイン・エフェクトの追加)



Strip Properties

(ストリップのプロパティ)

ストリップのプロパティは、プロパティシェルフ(N)の各パネルで確認や設定ができます。

- Edit Strip(ストリップ編集) - ストリップのプロパティの変更
- Strip Input(ストリップ入力) - 画像の取得元
- Effect(エフェクト) - エフェクトストリップの設定
- Filter(フィルター) - 画像のプリプロセス
- Proxy(プロキシ) - 実際の画像の代理を使用。低スペックPC用
- Scene(シーン) - シーンストリップ用の設定
- Sound(サウンド) - 音声クリップ用の設定

Edit Strip Panel

(ストリップパネルの編集)

Name(名前)

ストリップに名前をつけたり、名前を変えたりできます。

Type(タイプ)

選択したストリップのタイプを表示します。

Blend Mode(ブレンドモード)

デフォルトでは、ストリップは下にあるすべてのストリップの出力画像に置き換わります(Replace/置き換え)。ストリップのタイプに応じて他の多数のブレンドモードが利用できます。たとえば Alpha-Over(アルファオーバー)は下にあるストリップに、自動的に画像をオーバーレイします。自動ブレンドモードにより、エフェクトストリップを別に分ける必要がなくなります。Opacity(不透明度)でストリップのエフェクト影響度や経時的な変化を調整します。

Opacity(不透明度)

ストリップの不透明度を設定します。

Mute(ミュート、目のアイコン)

最終的な画像の計算に関わらせないように、ストリップを隠します。

Lock(ロック、鍵のアイコン)

ストリップが動かせないようにします。

Channel(チャンネル)

ストリップのチャンネル番号(行)を変えます。

Start Frame(開始フレーム)

ストリップの開始フレーム番号を変えます。ストリップをマウスで動かすのと同じです。

Length(長さ)

ストリップに使うフレームの枚数を指定します。

Effect Strip

(エフェクトストリップ)

ビルトイン、プラグインの種類によらずエフェクトストリップはどれも、通常は他のストリップか別のチャンネルにある二つのストリップに対して、特別な画像操作を行います。エフェクトストリップのエフェクトは、それぞれコントロールは違いますが、どれもプロパティパネルで調節できます。遷移速度を変えるにはストリップの長さを調節します。エフェクトストリップはいずれかのチャンネルに表示されますが、その結果として得られるエフェクトはチャンネル0に表示されます。

Strip Input

(ストリップ入力)

ストリップの入力を調整します。ファイルパス、ファイル名、画像オフセット、クロップの設定があります。

ここでストリップが使うファイルのパスを編集/更新できます。何らかの方法でファイルが動かされたときに役立ちます。ストリップを削除して再作成する手間を省けます！

Filter

(フィルター)

プリプロセスの共通オプションを簡単にセットできます。

Strobe(ストロボ)

Flip(反転)

Xは画像を左右に反転します。Yは上下です。

Backwards(逆方向)

ストリップ画像シーケンスを逆方向に動かします

De-Interlace(インターレース解除)

ビデオファイルにあるフィールドを取り除きます

Saturation(彩度)

画像の彩度を増減させます。

Multiply(乗算)

この値を色に乗算します。

Premultiply (プリマルチプライ)

アルファチャンネルを事前に乗算します (バージョン 2.66 以降は Strip Input (ストリップ入力) パネルにあります)。

Convert Float (Float に変換)

入力を浮動小数点データに変換します。

Use Color Balance (カラーバランス)

配色調整用の3つのフィルター Lift、Gamma、Gain を使えます。各パスは効果の正負をボタンで切り替えられます。カラスウォッチから効果の総量を設定してください。白 (RGB 1,1,1) は一切効果を持ちません (バージョン 2.64 以降は Modifiers (モディファイア) パネルにあります)。

Proxy Strip Properties Panel

(プロキシストリップのプロパティパネル)

プロキシは元の画像の代わりになる、小さな (読み込みが速い) 画像です。Strip (ストリップ) » Rebuild Proxy and Timecode Indices (プロキシとタイムコードインデックスの再構築) を行うと、Blender は多少時間をかけて大きな画像の代わりにサムネイルのような小さな画像を作ります。これでスクラブやスクロール、プロキシを使った Cross のような合成のような編集機能はずっと速くなりますが、結果の解像度は落ちます。最終的なレンダリングの前には、プロキシを無効にしてください。

プロキシを実際に 使うためには、プレビューモードのシーケンサービュー (編集したものが再生される場所) のプロパティシェルフ、View Settings (ビュー設定) で適切な「Proxy Render Size (プロキシレンダリングサイズ)」ドロップダウンの値を選んでおく必要があります。

Sound

(サウンド)

このパネルはサウンドファイルが選択されたとき現れます。

サウンドファイルのパスや名前を指定できます。

Pack (パック)

サウンドファイルを現在開いている .blend ファイルにパックします。

Caching (キャッシュ)

サウンドファイルをデコードして RAM に読み込みます。

Draw Waveform (波形を描画)

ストリップ上に音声ファイルの波形を描画します。

Volume (ボリューム)

音声ファイルの音量を設定します。

Pitch (ピッチ)

音声ファイルのピッチを設定します。

Pan (パン)

音声ファイルのパンを設定します。

Trim Duration: Start/End (期間トリミング)

音声ストリップの開始終了のオフセット。

Scene

(シーン)

操作対象としているシーンストリップにリンクするシーンを指定します。

Sequencer (シーケンサー)

シーケンサーストリップが存在すれば、ビデオシーケンスエディタのパイプラインを通してレンダリング (およびコンポジション) 処理を行います。これはレンダー設定にあるものと同じ機能です。(バージョン 2.65 以降はこのパネルでは設定できなくなりました)

Camera Override (カメラオーバーライド)

使用するカメラを変更します。


Adjusting the View


(ビューの調整)

VSE のシーケンスビューの調節にショートカットを使えます:

パン MMB 

ズーム Wheel 

縦スクロールには ◊ Shift Wheel  を使うか、左のスクロールバーのハンドル (明るい灰色の部分) をドラッグしてください。

横スクロールには Ctrl Wheel  を使うか、下のスクロールバーのハンドル (明るいハンドルの部分) をドラッグしてください。


縦方向にビューを拡大縮小するには、縦スクロールバーのハンドルの端にある円をドラッグしてください。

横方向にビューを拡大縮小するには、横スクロールバーのハンドルの端にある円をドラッグしてください。

いつものように View (ビュー) メニューはワークスペースの表示の仕方を調整します。

Properties (プロパティ) N


ビューの表示方法に関するオプションのある、プロパティシェルフを表示します。

View all Sequences (全てのシーケンスを表示) 

すべてのストリップが表示されるようにズームインアウトします。

View Selected (選択部分を表示) . NumPad

選択したストリップに合わせて表示をズームします。

Fit preview in Window 

(プレビューモードのみ)ウィンドウに合わせてプレビューのサイズを変えます。

Show Preview 1:1 1 NumPad

(プレビューモードのみ)プレビューサイズを 1:1 の倍率に変えます(実際の大きさ)。

多数のストリップの整理に画面全体を使いたいとき使ってください。

モード: シーケンスモード

ホットキー: T

メニュー: View(ビュー) » Show Frames(フレーム数を表示)、Show Seconds(秒数を表示)

Draw Frames(秒数を表示)

フレーム番号インジケータに、時間の代わりにフレーム番号を表示します。

Show Frame Number Indicator(フレーム数を表示)


ワークスペース最下部にある目盛りの、単位を秒数またはフレームに切り替えます。

Sync Markers

ストリップだけでなくマーカーもトランスフォームします。

Scrubbing

(スクラブ)

動画内を行き来するにはタイムラインウィンドウを使います。タイムラインウィンドウ内を LMB  クリックして左右にドラッグすると、カレントフレームを示す縦棒が移動します。縦棒にあわせて、VSE ウィンドウにはそのフレームの画像が表示されます。

画像シーケンスか動画ファイルを見ているときは、普通のコンピュータならリアルタイムスクラビングと画像表示が可能です。シーン画像は個別にレンダリングされる必要があり、多少時間がかかるかもしれません。

View Modes

(ビューモード)

ヘッダーにあるアイコンは VSE のビューを変更します。デフォルトではシーケンサーだけが表示されています。2番目のボタンはプレビューウィンドウだけを表示し、3番目のボタンはシーケンサーとプレビューの両方を表示します。

プレビューが有効なら、表示するプレビューを数種類から選べます。詳しくは [シーケンサーの表示モード](#) をご覧ください。

Scene Preview

(シーンのプレビュー)

シーケンサー内でシーンストリップを使うと、プレビューウィンドウの表示の仕方を(プレビューウィンドウの)プロパティシェルフの設定で決められます。

Open GL Preview(Open GL プレビュー)

Open GL を利用できるなら、この設定を有効にしてシーンのプレビューレンダリングに Open GL を使ってください。

ドロップダウンメニューではシーンの表示方法を選べます(バウンディングボックス、ワイヤーフレーム、ソリッド、テクスチャなど)。

View Settings

(ビューの設定)

プロパティパネルの View Settings(ビュー設定) には、さらなる表示オプションがあります。

Show Overexposed(露出過多を表示)

1以上にすると、プレビュー画像に露出過多な部分を表す縞模様のオーバーレイを重ねます。値が高いほど、露出過多のしきい値が高くなります。

Safe Margin

プレビューにタイトルセーフ領域を重ねて示します。

Proxy Render Size(プロキシレンダサイズ)

元の解像度またはプロキシの別の解像度を使ってプレビューを描画します。レンダリングの解像度はレンダ設定パネルで決まります。プレビューサイズは小さくするほど高速です。

Separate Colors(色の分離)

(Luma Waveform ビューのみ)R、G、B をそれぞれ別のグラフにわけます。

Refresh View

(ビューの最新化)

3D ビューでオブジェクトを動かすような特定の操作では、シーケンサー内のレンダリング画像が最新化されないことがあります(移動はレンダリング画像に影響しない可能性があるため)。ストリップとして使われている画像や動画が Blender 外部のアプリケーションから変更されても、Blender には OS からの通知を実際に知る方法がありません。強制的に Blender にファイルを再度読み込ませ、3D ビューの再描画をさせるには、ヘッダーにある Refresh Sequencer(シーケンサーを更新) ボタンをクリックします。Blender がキャッシュされたすべての画像を更新して同期し、現在のフレーム内容を計算します。

Selecting Strips

(ストリップの選択)

Select (選択) メニューではさまざまな方法でストリップを選択できます。

Strips to the Left (左側のストリップ)

選択中のストリップより左にあるストリップをすべて選択します。

Strips to the Right (右側のストリップ)

選択中のストリップより右にあるストリップをすべて選択します。

Select Surrounding Handles (両ハンドル)

選択中のストリップの両ハンドルを選択します。

Left Handle (左ハンドル)

選択中のストリップの左ハンドルを選択します。

Right Handle (右ハンドル)

選択中のストリップの右ハンドルを選択します。

Linked (リンクしたものを選択)

選択中のストリップにつながったストリップをすべて選択します。

Select All (全選択) A

読み込まれているストリップをすべて選択します。

Select Inverse (反転)

現在の選択を反転します。

Border Select (範囲選択) B

範囲 選択モードに入ります。シーケンスワークスペースにあるストリップの領域をクリックし、ドラッグして矩形の囲みを作ります。マウスボタンを離すと、矩形内のストリップが追加選択されます。


Moving and Modifying Strips

(ストリップの移動と編集)

G で選択ストリップを時間単位またはチャンネル単位で移動します。マウスを左右に動かすとストリップの位置が時間単位で変わります。上下に動かすとチャンネルを変更できます。

- スナップさせるにはドラッグ中に Ctrl を押し続けてください。
- 「ripple edit」(ドラッグ中のストリップ用に場所を空ける) をするには、ストリップを置くときに Alt を押したままにしてください。

間違って追加したストリップやもう要らないストリップがあれば、X を押すか Strip (ストリップ) » Erase Strips (ストリップを消去) で削除できます。

つながっていない複製を作るにはストリップを Duplicate (複製) します。ドラッグして開始時間やチャンネルを決めたら、LMB  クリックして確定します。

Strip (ストリップ) メニューには、ストリップを扱うためのさらなるツールがあります:

Grab/Move

Grab/Extend from Frame

Cut (hard) at frame

Cut (soft) at frame

Separate Images

Deinterlace Movies

Duplicate Strips

Erase Strips

Set Render Size

Make Meta Strip

UnMeta Strip

Reload Strips

Reassign Inputs

Swap Inputs

Lock Strips

UnLock Strips

Mute Strips

Un-Mute Strips

Mute Deselected Strips

Snap Strips

Swap Strips

Snap to Frame

(フレームにスナップ)

ストリップを選択し、カーソル(緑の縦線)を希望の場所に動かして ◊ ShiftS を押せば、ストリップの先頭をカーソル位置に正確に合わせることが出来ます。時間表示の単位が秒なら、ズームすると秒の小数部が表示されます。単位がフレームなら、個々のフレームに至るまでズームできます。




Separate Images to Strips

(ストリップの分割)


画像シーケンスを入力に使う画像ストリップは、Y で一定のフレームごとにストリップを分割できます。

Editing Strips

(ストリップの編集)

- ストリップの中ほどを RMB  するとストリップ **全体** をつかめます。マウスを押したままにして(または G を押して)ストリップを自由に動かせます。
- ストリップの左の矢印を RMB  すると、ストリップの **開始** フレームのオフセットを選べます。マウスを押したままにして(または G を押して)左右に動かすと、動かしたフレーム数だけストリップの開始フレームを変更します:
 - 20枚の画像シーケンスのストリップがあるとして、左矢印を右に10フレームドラッグすれば、ストリップは画像11から開始します(画像 1 から 10 まででは飛ばされます)。ロールアップや不要な導入部を切り取る時に使ってください。
 - 左矢印を左にドラッグすれば、最初のフレームをドラッグしたフレーム数分コピーします。このクリップへのトランジション用にフレームが何枚か必要なときに使ってください。
- ストリップの右矢印を RMB  するとストリップの **最終** フレームを選択します。マウスを押したまま(またはGを押して)動かすと、ストリップの最終フレームを変更できます:
 - 右矢印を左にドラッグするとクリップを短くできます。末尾にあった元の画像はすべて無視されます。ロールダウンを手早く切り取る時に使ってください。
 - 右矢印を右にドラッグするとクリップを延長します。動画と画像シーケンスでは、延長するほど使われるアニメーションは(底を尽かない限り)長くなります。アニメーションの終端を越えてクリップを延長すると、Blender は末尾の画像を複製します。クリップ外側のトランジション用に使ってください。

複数選択

◦ Shift RMB  クリックすれば複数のストリップ(のハンドル)を選択できます。G を押すといま選択されているものすべてをマウスで動かせます。このため最初のストリップを移動して別の2つの長さを短くし、4つ目のストリップは延長する、といったことが同時に行えます。


- **ストリップ延長**。複数のストリップを選択して E を押すと延長モードに入ります。選択したストリップの、カレントフレームインジケータより「マウス側」にあるハンドルはすべてトランスフォームします。これで基本的に、他のすべてとの釣り合いを取りつつ、ストリップをカレントフレームマーカーに向かって正確に延長することができます。

2つのストリップの端と端を並べて置くだけで繋ぎを作れますが(訳注: ストリップそのものは繋がりません)、切断するには K を押します。選択ストリップは、現在のフレームで2つに切断されます。ロールアップやリロードイン、ロールダウン、あるいは余分な撮影をカットするのに使ってください("C" は Change 用に使われているので "K" を使います)。

切断の注意点

ストリップを Cut/切断しても、本物のフィルムの「古い編集」のように本当に切ってしまうことはありません。実際にはストリップのコピーが作られます。元のストリップの終わりとし新しいストリップの始めの「フィルム」はそれぞれ、切断箇所です巻き取られます。

たとえば **50** フレームの長さのストリップがあって、先頭10フレームを削除する場合を考えてみましょう。必要な操作は **11** フレームに移動して K を押すことです。切断によりストリップは二つの部分に「分割」されます。これで最初の小さな部分(フレーム 1 から 10)を選んで X で削除できます。

フレーム **1** から **10** までが本当に消えたように見えますがそうではなく、フィルムのように「巻き取られて」まだフレーム **11** の下にあります。切断操作でコピーされたストリップの、片方を削除したというだけです。「なくなった」フレームはいつでも取り戻すことができます。ストリップの左矢印を RMB  クリックして、G で希望のフレームが表示されるまで左に動かすだけです(もしくは右に動かしてもっとフレームを「隠す」ことができます。これはストリップの始めや終わりのフレームを削除するもう一つのやり方です)。

これはほぼすべての編集操作で重要になり、非常に便利です!

動きの停止

開始フレームをそれ以前に伸ばしたり、終了フレームをそれ以降に伸ばしたりするとき、一番端にある画像だけがコピーされることは忘れないでください。再生してみると、動きはそのフレームで止まります。動きが止まったことが目立たないように、動きがまだある間に少し早めにトランジション(fade、cross)を開始してください(もちろん80年代の連続ホームコメディを作りたい場合は除きます)。

エフェクトストリップの長さを変えるには、元のストリップの開始/終了フレームを変更します。

Copy and Paste

(コピーとペースト)

ヘッダーにある二つのボタンを使って、クリップのコピーやペーストが行えます。

Meta Strips

(メタストリップ)

メタストリップはストリップのグループです。グループ化したいストリップをすべて選んで CtrlG でグループ化し、ひとつのメタにします。メタは最初のストリップの始めから最後のストリップの終わりまでの長さを持ち、ちょうど音声ソフトウェアで言うミックスダウンのように、全チャンネルを単一のストリップに凝縮します。メタをバラバラにする(グループ解除する)と、それぞれが元の位置やチャンネルに復元されます。

メタのデフォルトのブレンドモードは Replace (置き換え)です。ブレンドモードがアニメーションの結果を変えることはよくあるので、アニメーションの結果をよく見て、必要ならブレンドモードを調整してください。

メタストリップを使うと便利なのは、複数のストリップに同じエフェクトを用いる場合です。たとえばループのスケールリングです。Blender がループエフェクトを手に入れるまでは、クリップをループさせるにはクリップを複数回コピーして並べるしかありません。クリップをトランスフォームさせる必要があれば(たとえばアニメーションさせた透かしや別のアスペクト比を持ったソースマテリアルの拡大縮小と移動)トランスフォームの組み合わせをループ内のストリップそれぞれに使うよりも、先にストリップをメタストリップにして、メタに対して一度だけ使うほうがはるかに便利です。

メタストリップを選択して Tab を押すと、中身を編集できます。ストリップの編集を終えるには Tab をもう一度押します。メタストリップは入れ子にできるので、メタストリップの階層をひとつ抜け出すときは、Tab を押す前にアクティブなメタストリップがないことを確認してください。

Sequence Display Modes

(シーケンサーの表示モード)

ヘッダーバーにあるオプションを使って、ビデオシーケンスエディタ(VSE)はシーケンスをさまざまな方法でリアルタイムに表示できます。

VSE のタイプをヘッダーバーの3つのボタンから選べます。1つ目のボタンはシーケンサーだけをエディタに表示し(デフォルト)、2つ目はプレビューだけ、3つ目はシーケンサーとプレビューの両方を表示します。デフォルトの Video Editing スクリーンレイアウトでは左下にシーケンサータイプ、右上にシーケンサー+プレビュータイプの VSE が表示されています。

プレビュータイプの VSE はヘッダーバーにさらに表示モードのボタングループが現れ、現フレームの合成結果からワークスペースに表示する特性を選べます:

- 画像/シーケンス: 色(目に見えるもの)
- 彩度(Chroma)ベクトルスコープ: 色調と彩度
- 輝度(Luma)波形: 明るさ/コントラスト
- ヒストグラム: 赤、緑、青の度合い

彩度、輝度、画像モードではチャンネルセレクトが現れます。重なったストリップやエフェクトストリップをすべて合成した結果がチャンネル0です。チャンネル1は現在のフレームのチャンネル1にあるストリップの画像がどのようになるか示しています(チャンネル1は重なり最下層にあたります)。このモードの表示は合成結果(チャンネル0)かストリップによるフレーム(チャンネル1からnまで)のいずれかになります。

ワークスペースのビューのズームにはマウスホイールをスクロールさせます。

Image Preview

(画像のプレビュー)

Video Editing スクリーンレイアウトの右上にある VSE ウィンドウは、画像プレビューモードに設定されています。ここにはビデオが保存された時にどのように見えるかを表示します。ストリップの追加、移動、分割、グループ化(メタの作成)、特殊エフェクトを使った結合といった作業の中心となるモードです。

Luma Waveform

(輝度波形)

この表示では選択チャンネルの明るさ、輝度の分布図が描かれます。

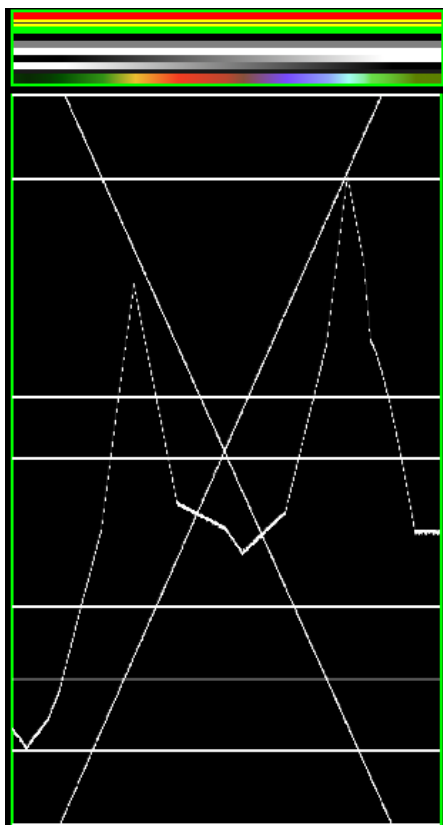
輝度波形を使うとビデオ信号の輝度分布の特色を評価でき、各制御モニター上の普段の出力表示の代わりに、輝度の波形を見ることができます。

この表示は各スキャンラインの輝度を描きます。ラインは他に重ねて描かれます。ラインの交差点は明るさが増します(数百のスキャンラインがあればたいてい交差します)。テレビのビデオ出力の Luma/輝度 にオシロスコープをつなぐと、この図を最もよく理解できるでしょう。基本的には同じ見方です。

このモードは縦軸が輝度を示します。最下部が0、最上部が1です。横軸はフレームの横軸と対応します。フレームにあるスキャンラインの本数と同じ数の曲線があります。曲線はそれぞれ、ひとつのライン上のピクセルの輝度を表します。またピクセルの色は、同じ輝度を持つフレームの対応する列にあるピクセル数を表します。つまりこの点を通過する曲線の数です(ピクセルがなければ黒/透明、少なくとも3ピクセルあれば白/不透明です)。

このモードの用途:

- 波形が図の全体に広がっていなければ gamma-plugin の「setup」と「gain」のマスタースライダーを調整し、全体に広げるとよいでしょう(コントラストの自動伸張)(訳注:このプラグインを使えない場合(バージョン2.6系)は、ストリップモディファイアで色調補正を行えます。モディファイアはストリップのプロパティパネルから追加できます。ストリップを個別に調整する代わりに、「Adjustment Layer(レイヤー調整)」エフェクトストリップを一番上(あるいは希望のチャンネル)に差し込んで、このストリップにモディファイアを入れることもできます)。
- もっと高度な gamma-plugin を使うと彩度を減らす箇所(特に暗い部分)を自分で決めることができます。
- 不自然な省略がある一番上や一番下のデータをすべて捨てることもできます。



単純な絵。

輝度波形にある数本の横線は元の画像の単色の線と一致します。

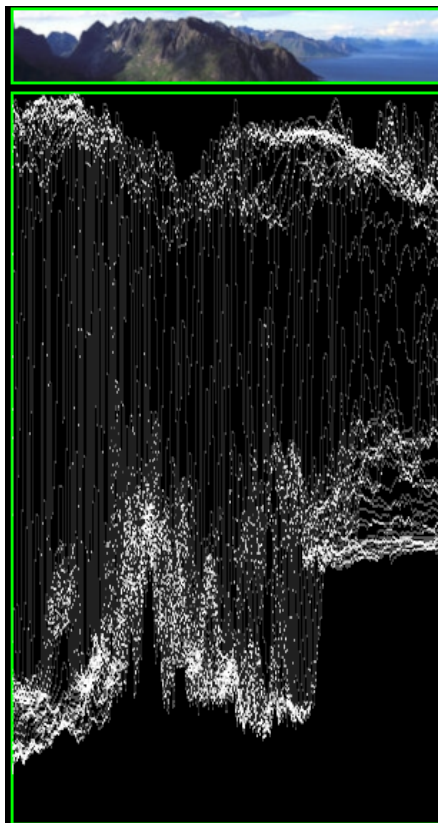
「灰色 20%」の1ピクセル幅の線(黄色のストリップ内)が輝度波形では灰色の線で表されていることに注目してください。

「X」を描く2本の線は、2つの線形グラデーション(白→黒と黒→白)から来ています。

最後に、折れ線は元の画像の一番下にある複雑なグラデーションに対応しています。

VSEの輝度プレビューの例。

なお輝度波形に表示する曲線数を限定するため、画像(一番上の最初の緑の枠内)は50pxの高さしかありません。



現実の写真。

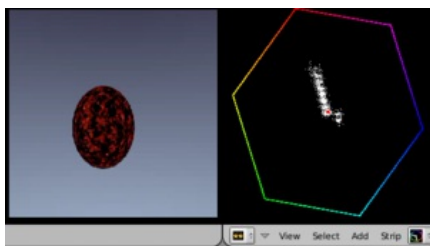
非常にわかりやすい曲線が描かれています。

輝度は空で80-100%、海で約40%になり、山では10-20%ですが日の照っている箇所では約40%に増えていることがわかります。

この表示を使って、チャンネルの全フレームに渡って妥当なコントラストと輝度であることを確認してください。一様に照明の当たるはずの箇所がそうならないければ、照明が消えたように見えたり、余分な照明が付けられたように見えたりします。途切れずつながるはずの二つのストリップが別の照明条件でレンダリング/撮影されている場合に、こういったことが起きます。

Chroma Vectorscope

(彩度ベクトルスコープ)



VSEのChromaプレビューの例

このモードは色の分布や彩度の評価に使ってください。UV散布図を見ることもできます。

画像はYUV形式に変換されます。UとVの値は色の角度を表します。画像の各ピクセルに対して、点がひとつUとV値の位置に描かれます。複数ピクセルが同じUV値を持っていれば、図のピクセルの明るさが増します。

何色であるのかわかりやすくするために、最も遠い点(赤、マゼンタ、青、シアン、緑、黄)を示す六角形と、原点を示す赤の四角が描かれます。

言い換えれば、選択チャンネルの色空間を六角形の内部に表示します。六角形の各頂点は赤、マゼンタ、青、シアン、緑、黄といった原色です。中央は黒で、外辺に近づくほど全体的に彩度が増えます。右の例では画像に赤(50%の彩度)が多く、少し青があり、緑はないことがわかります。

常に最終結果のモニターを開いたままにしてください。色調補正は好みの問題で、あなたの思惑次第です。

このモードは彩度が多すぎないか調べるのに使ってください。彩色過多の画像はオプアートやコンピューターディスプレイでは見栄えがしますが、テレビの大型画面ではひどいものになります。AltAを使ってビデオをスクラブしてください。この表示は各フレームごとに新しい/修正された図で更新されます。画像プレビューで見た目を確認する要領で、彩度ベクトルスコープで色の使い方を観察してください。

このモードの用途:

- 画像がとても陰気で彩度が少なすぎるようなら、UVプロットを見るとよいでしょう。おそらく原点に全ピクセルが集まっていることがわかります。gamma-plugin で(訳注: プラグインを使わない場合 2.6 系ではフィルターパネルやモディファイアで)彩度を足す場合は、色を歪ませていないか UVプロットでわかります。
- 手動でカラーマッチングをする場合はさまざまなチャンネルのモニターを見て角度を合わせます。

Histogram

(ヒストグラム)

このモードは現在表示中の画像の、ピクセルの色分布グラフを表示します。X 軸は0から1 (または0から255) でピクセルの値を表し、Y 軸はその階調範囲にあるピクセルの数を表します。全体的に暗い画像はほとんどの情報がグラフ左側に偏るでしょう。

このモードは画像の階調範囲のバランスを取るのに使ってください。うまくバランスのとれた画像は色値が滑らかに分布します。

Sequencer Screen Layout

(シーケンサーのスクリーンレイアウト)



デフォルトの Video Editing スクリーンレイアウト

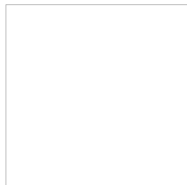
Sequencer Effects

(シーケンサーエフェクト)

Blender は16種類のエフェクトを内蔵しており、誰もが使えます。2つのストリップを操作するもの、1つのストリップを操作するもの、新たにストリップを作るものがあります。エフェクトは作品の質を何らかの方法で向上させたり、プロ品質のトランジションを行ったりします。

[edit] Add

(加算)



雷鳴が聴こえますか？

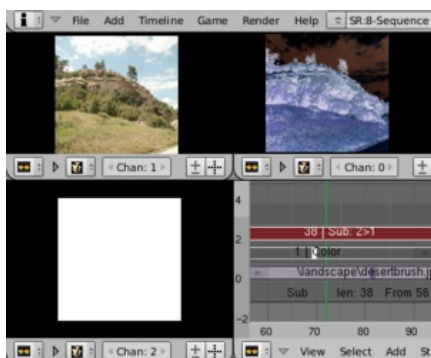
Add(加算)エフェクトは二つの色を足します。赤とシアン(緑と青)は白になります。赤と青はマゼンタ(紫！)になります。赤と緑は黄色になります。

Add(加算)エフェクトは2つのストリップの色を足します。ベースになる画像ストリップと修正用のストリップを用意します。修正用ストリップはベタ塗りか白黒のマスク、または完全に別の画像です。右の図は画像に灰色を Add した様子で、エフェクトを経時変化させています。灰色(R:.5, G:.5, B:.5)を足すので画像は明るくなります。つまり青色(R:.1, G:.1, B:.5)は元の色相(hue、色の間の関係)を保ったまま明るくなって(高い値になって)(R:.6, G:.6, B:1.0)に変わります。このように画像全体に用いると画像全体がパッと光るように見えます。

このエフェクトは画像の明るさを上げるのに使うことができ、白黒のマスクを使えば画像の特定の部分だけ明るさを上げられます。Mix ノードを Add モードで使うと、Add ストリップを使うのとまったく同じことがノードで行えます。Factor(係数)を入力すれば調整も同じやり方で行えます。

[edit] Subtract Effect

(減算)

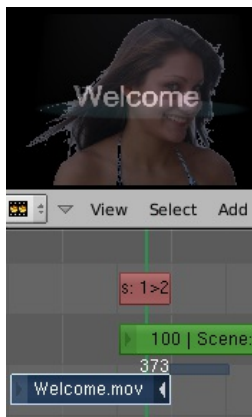


Subtract エフェクト

このエフェクトは2つ目のストリップの色から1つ目のストリップの色を減じます。画像のネガを作ったり、ストリップの順序を変えてただストリップを暗くするために使えます。白い画像から青の色相を減じれば、残った赤と緑によって画像は黄色くなります。

[edit] Cross and Gamma Cross

(クロスとガンマクロス)

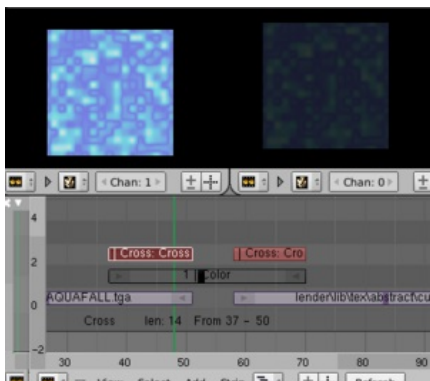


2つのストリップの重なりあった部分で、画像全体を一方から一方に溶け込ませるようにして徐々に遷移します。遷移の速さは重なった部分の長さ(フレーム数)に応じて変化します。非常に役立つストリップです。

Gamma Cross(ガンマクロス)はフェード中に色調補正を行い、目にやさしい滑らかな遷移を生みます。

[edit] Fade to Black

(フェードアウト)



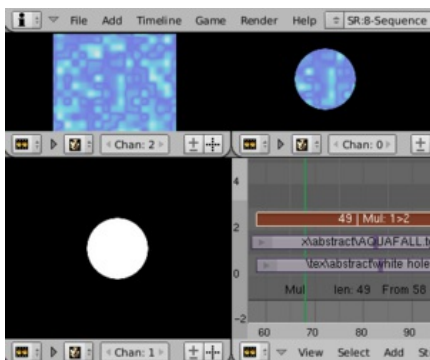
黒を介した Cross-Fade

場面の切り替えを直接行わず、徐々に暗転させてから行い、また徐々に明るくすることがよくあります。

このためのストリップ設定が右の図です。2つのストリップはチャンネル1にあり、この2つをまたぐようにチャンネル2に Color(カラー)ストリップを追加します。色を黒に変え、Cross エフェクトを2つ追加します。1つ目はチャンネル1からチャンネル2(黒)に、片方はチャンネル2からチャンネル1に設定します。最初のストリップは黒にフェードアウトし、続いて2番目のストリップが黒からフェードインします。もちろんトランジションの色には好きな色を使えます。黒を介すると落ち着きます。赤は不安にさせます。2番目のストリップを導くために、その主調色を使うことがあります。

[edit] Multiply

(乗算)



Multiply エフェクト

Multiply(乗算) エフェクトは2つの色を乗算します。Blender は **0.0** から **1.0** までの値を色に使います。**0.0** から **1.0** までの値を2つ乗算しても常に結果は **0.0** から **1.0** までの値になるので、この処理を正規化する必要はありません(3バイトを使う「伝統的な」表現なら—たとえば RGB(**124, 255, 56**)—、乗算結果ははるかに高い値になり— RGB(**7316, 46410, 1848**)のように—、この場合「回復させる(正規化する)」必要があります。**0** から **255** の範囲に「戻す」には単に **256** で割るだけです！)。

このエフェクトの主な用途は二つあります:

マスクと一緒に使う

マスクとは白黒の画像で、「通常」の画像と乗算を行えばこの画像の白い部分だけが表示されます(他はすべて黒くなります)。ジェームズ・ボンドの映画のオープニングタイトルシーンではカメラが銃身からジェームズを覗いています。このエフェクトの良い例です。

ベタ塗りの色と一緒に使う

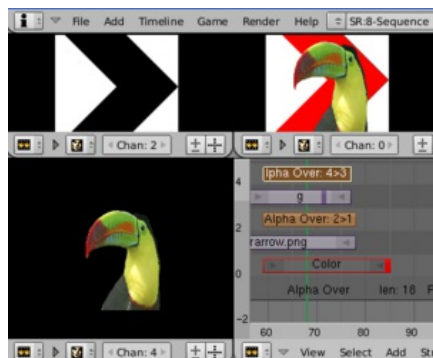
「通常」の画像と単色を乗算すれば、色相を部分的にやわらかくできます(相対的に他色の強調にもなります)。たとえば茶色のピクセル RGB(**0.50, 0.29, 0.05**)に対してシアンのフィルター(RGB(**0.0, 1.0, 1.0**)のベタ塗り)を乗算すれば、RGB(**0.0, 0.29, 0.5**)ができます。見た目には赤が弱まって青と緑が(実際の値は変わりませんが、相対的に)持ち上がります。物理的にはチョコレートバーをシアン光で照らすのと同じ効果です。印象としては植物はより青々とし、水は魅力を増してカリブ海っぽくなり、空は心地よさが増します。

備考

このエフェクトは画像の大域照明の明度を下げます。片方の画像が真っ白なら結果はもう片方の画像と同じです。片方の画像が真っ黒なら結果も真っ黒になります。

[edit] Alpha Over, Under, and Over Drop

(アルファオーバー、アルファアンダー、オーバードロップ)



AlphaOver エフェクト

このエフェクトはアルファ(透明度チャンネル)を使って、ドミナント画像の透明な部分に基づいて結果を合成します。Scene(シーン)ストリップを使うと、画像の色のない部分は透明になり0のアルファ値を持ちます。Movie(動画)ストリップを使うと動画は1のアルファ値(完全不透明)を持ちます。(訳注: シーンの色のない部分を透過させるにはレンダー設定の Shading(シェーディング) パネルで Alpha(アルファ) を Transparent(透過) にする必要があります(Blenderレンダーの場合))

CGのシーンを動画の上に合成するのに Alpha Over/Alpha Under エフェクトを使うことができます。この結果あなたが作ったキャラクターの動きを何であれ動画の一部であるように見せることができます。係数の曲線は、背景に重ね合わせた前景を溶け込ませる割合を操作します。前景画像の透明な部分は無視され、背景の色は変化しません。

2つストリップを選択してください(⇧ Shift RMB):

- Alpha Over を使うとストリップは選択順に重ねられます。最初に選択したストリップが背景になり、2番目のストリップが *その上* になります。Factor(係数)は *前景の透明度* を調節します。Factor(係数)を **0.0** にすれば背景だけを表示し、**1.0** にすれば前景が背景の完全上になります(もちろん透明な部分は除きます!)。
- Alpha Under を使うと正反対になります。最初に選択したものが前景に、2番目が背景になります。さらに Factor(係数)は *背景の透明度* を調節します。**0.0** にすれば前景だけを表示し(背景は完全に透明になります)、**1.0** にすれば Alpha Over を使うのと全く同じ結果になります。
- Alpha Over Drop は両者の中間です。Alpha Under と同様に最初に選んだストリップが前景になりますが、Factor は Alpha Over と同様に前景の透明度を調整します。

右の例は AlphaOver エフェクトによる重ね合わせです。最下部のチャンネルは赤色で、矢印はその上のチャンネルにあります。この二つはチャンネル3に AlphaOver されています。私の好きなオオハシはチャンネル4にいて、チャンネル5は合成された赤い矢印の上にオオハシを AlphaOver します。最後に追加したエフェクトは、レンダリングに使われるチャンネル0に結びつけられます。

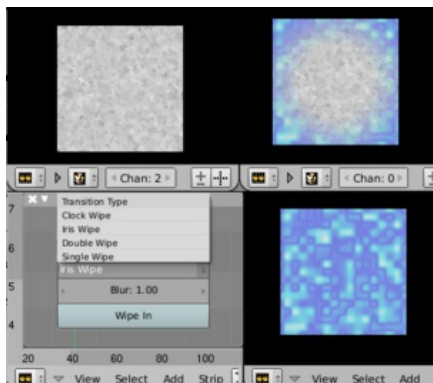
前景のストリップのプロパティパネルで PreMult Alpha ボタンを有効にすると、二つのストリップのアルファ値は乗算や加算がされません。色々なアルファチャンネル(不透明の部分、透明部分、その間の部分)を持つ前景ストリップを、均一な不透明の(アルファが 1.0 以上の)チャンネルを持つストリップの上に追加するときは、このエフェクトを使ってください。AlphaOver 利用時に前景オブジェクトのあちこちが光っていたり、変な透明部分があたりしたときは、PreMultiply を有効にしてください。

AlphaOver Drop エフェクトは Cross によく似ていますが、一番上か二番目の画像に優先度をつけ、Cross が行うブレンドよりもゆるやかなオーバーレイ効果を与えます。もちろん Alpha エフェクトはどれもアルファチャンネルを考慮しますが、Cross はしません。

アルファの適用度、つまり混色度は Fカーブで調節できます。Sine 波にすると前景のフェードインやフェードアウトの効果が得られます。

[edit] Wipe

(ワイプ/拭き取り)



Wipe 組み込みエフェクト

Wipe(ワイプ)はストリップからストリップへ徐々に移行します。柔軟性のあるエフェクトで、次の4種類のトランジションを備えます:

- Clock(時計): アナログ時計の針のように 9:00 の位置から時計回りか反時計回りに円を描きます。円が描かれるにつれ、次第に次のストリップが見えます。
- Iris(絞り): カメラの絞りや人の目の虹彩のように、徐々に広がる(または縮む)円の内側に、次のストリップを表示します。blur(ぼかし/ブラー)を使うと、インクが紙にじむような見た目になります。
- Double Wipe(ダブル): 中央から外に向けてワイプし、次のストリップを徐々に表示します。外から内に向けたワイプインもできます。ワイプの向

かう角度や blur(ぼかし/ブラー)の設定もできます。

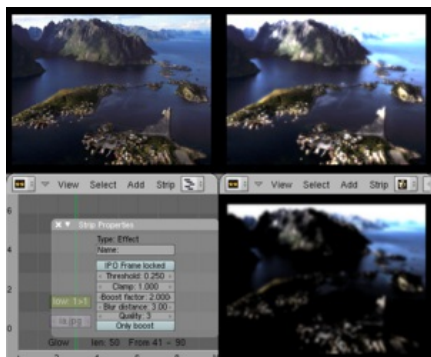
- Single Wipe(シングル): カバーをめくるように次のストリップを表示します。角度を変えて隅からめくるようにしたり、表示をぼかすことができます。

ストリップ1つから作ったワイプは無効です。ワイプの持続時間は2つのストリップの交差部分の長さで決まり、ワイプそのものでは調整できません。いずれかの入力ストリップの端を交差範囲が変わる方向に動かして調節します。

なお古いプラグインには同様の機能を持つものがあります。

[edit] Glow

(グロー/発光)



Glow エフェクトを画像に使った例。

左上: ベース画像 (Lofoten Islands, Norway –

出典: wikipedia.fr);

右上: エフェクトの結果;

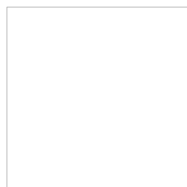
左下: エフェクトの設定;

右下: Only boost ボタンの有効な状態での結果

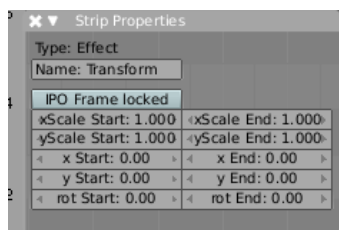
このエフェクトは画像の輝度チャンネルに働きかけて、画像の一部の明るさを増します。Glow はベースとなる画像と、Threshold(しきい値): より明るい部分がぼかされた編集用の画像との重ね合わせで、Threshold(しきい値): より明るい部分にぼかしがかかります。Glow ストリップのプロパティではこの Threshold(しきい値): や、加算できる最大輝度 (Clamp(範囲制限):)、その Boost factor(増幅係数):、ぼかしの大きさ (Blur distance(ブラー距離):)、ぼかしの Quality(品質): を設定できます。Only boost(増幅のみ) はベース画像を使わずに「編集された」バージョンだけを表示/使用する設定です。発光エフェクトをアニメーションさせるには、ベース画像から発光した画像へのクロス効果を得るため、このエフェクトとベース画像を Gamma Cross エフェクトを使って混ぜ合わせます。

[edit] Transform

(トランスフォーム)



トランスフォームは画像操作における十徳ナイフです。ストリップ内で画像の移動、拡大縮小、回転を行います。右の例は単独の画像を使ってどういったことを行えるか示しています。最終的な効果への滑らかな遷移を作るには、遷移の始点と終点でプロパティをアニメーション化して(項目上でI)、グラフエディタで補間曲線を編集してください。



Transform ストリップを選択した状態でプロパティパネルを開くと、このエフェクトの設定を調整できます:

(x,y) Scale(拡大縮小):

拡大率(大きさ)を調整します。xは幅、yは高さの定義です。**1.0**より高い値は画像を拡大し、**1.0**より低い値は縮小します。

Uniform Scale(均一スケール):

XとY両方に同じ拡大率を使います。

(x,y) Position(位置):

位置を調整します(平行移動)。xは横方向の位置で、正の値なら画像は右に、負の値なら左に移動します。yは縦方向の位置で、正の値なら画像は上に、負の値なら下に移動します。

Rotation(回転):

回転は度数法(**360**で一周します)で指定し、元の値より大きくすれば反時計回りに、小さくすれば時計回りに回転します。

[edit] Color

(カラー)

このエフェクトは単体で動作し、カラーストリップを作ります。デフォルトの長さは25フレームですが、端の矢印をつかんで動かすと延長できます。別の色を選ぶには Effect Strip (エフェクトストリップ) パネルのカラースウォッチをクリックしてください。デフォルトは黒です。メインの動画と重ねてフェードインやフェードアウトを作るのに使えます。

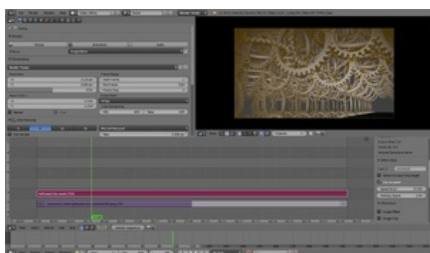
[edit] Speed Control

(スピードコントロール)

スピードコントロールはストリップの時間を歪め、通常より速いか遅い速度で再生します。Multiply Speed (速度を乗算) 欄を 1.0 を下回る値にするとストリップの再生を遅くします。1.0 を上回る値にすると速くします。速く再生するということはフレームの一部が飛ばされ、終端フレームが来るまでにフレームを使い果たすということです。表示するフレームがなくなればストリップの最後のフレームが単に繰り返され、物体の動きは止まったように見えます。これを避けるには元のストリップの、動きを止まらせたくない箇所の下に、次のストリップを配置してください。

[edit] Creating a Slow-Motion Effect

(スローモーションエフェクトの作成)



スピードコントロールを使った 50% のスローモーション

ストリップの再生を遅おくしてみましょう。全体のフレームレートに影響させずにビデオクリップの速度を変える必要があります。クリップを選んで Add (追加) » Effect Strip (エフェクトストリップ) » Speed Control (スピードコントロール) を選びます。Nを押してプロパティシェルフを開きます。Effect Strip (エフェクトストリップ) パネルの *Stretch to input strip length (入力ストリップの長さストレッチ)* のチェックを外してください。Multiply Speed (速度を乗算) に速度調整に使いたい係数を設定します。表示速度を 50% だけ抑えるのなら 0.50 を入力してください。これで275フレームを持つクリップは半分の速度で再生され、したがって最初の137フレームだけが表示されます。

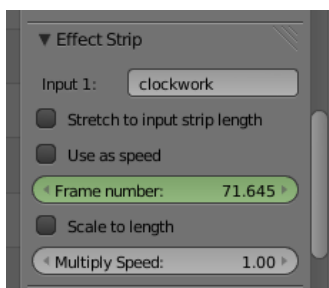
この最初のフレームを表示したあと、残りのフレームもスローモーションで表示したいときは、入力ストリップの長さを倍にしてください(エフェクトストリップの範囲は入力ストリップで制御します)。0.5 以外の速度係数を使うのなら、次の計算式が使えます:

$$\text{新しい長さ} = \text{実際の長さ} / \text{速度係数}$$

以上！ この例の場合、レンダー設定で 550 フレームすべてをアニメーションする設定にしてください。

[edit] Keyframing the Speed Control

(スピードコントロールのキーフレーム化)



クリップのタイミングをもっと細かく制御したいときは、曲線を使いましょう！ 速度係数をキーフレーム化することもできますが、通常はフレーム番号を直接キーフレーム化するのがよいでしょう。

Stretch to input strip length (入力ストリップの長さストレッチ) のチェックを外し、さらに *Use as speed (速度として使用)* のチェックを外します。これでキーフレーム化するフレーム番号の入力欄が現れます。何にせよストリップをアニメーションさせたいならキーフレームをいくつか差し込む必要があります、そうしないと写真のように見えます。曲線の補間はデフォルトのページでは望む結果にならないことがほとんどなので、普通はグラフエディタを使って Linear/線形 にするのがよいでしょう。

代わりに速度係数をキーフレーム化することを選んだ場合は、VSE のストリップビューのヘッダーにある Refresh Sequencer (シーケンサーを更新) ボタンを忘れずにクリックしてください。忘れると変更が反映されません。

[edit] Changing Video Frame Rates

(動画のフレームレートの変更)

スピードコントロールは動画の毎秒の再生フレーム数 (FPS) の調整に使えます。画像シーケンスセットに動画をレンダリングする場合、速度係数を

1.0 以外の値にすれば、作られる画像ファイルの枚数を事実上増減させることになります。たとえば5分のビデオを 30 fps で撮影して 24fps で上映されるフィルムに転送する場合は、速度係数に 30/24 または 1.25 を入力します(また Frame Blending を有効にしてフィルムにぼけた感じを与えます)。 $5 * 60 * 30 = 9000$ 枚のフレームを作る代わりに、Blender は $9000/1.25 = 7200 = 5 * 60 * 24$ 枚のフレームを作ります。この場合開始フレームを 1、終了フレームを 7200、出力形式を Jpeg の 30fps にすると 0001.jpg から 7200.jpg までの画像ファイルがレンダリングされますが、これらの画像は 9000 枚のフレーム全体を含んでいます。画像ファイル 7200.jpg はフレーム 9000 と同じです。できた画像を .blend ファイルに 24 fps で読み込み直せば、ストリップはぴったり5分の長さになります。

[edit] Multicam Selector

(マルチカムセレクタ)

Blender を使ってマルチカム編集をしたいと思ったことはありませんか？ びっくり仰天するほど簡単です(訳注:これは「実際の」カメラ複数台で同時撮影した映像の扱い方です):

- 入力ストリップをチャンネル1から4あたりに追加します(好きなだけ追加可能ですが、10 以上あると操作の反応が少しごちかくなります。下記参照)。
- ストリップ間の同期を取ります。自動同期の機能は Blender にありませんが、2つのビューウィンドウを開いてカメラ映像のひとつをマスターチャンネルに選び、足の動きや照明のフラッシュを見て(編集する番組によります)他のカメラ映像の再生タイミングを合わせます。将来的には動画全体の明るさに基づいて自動同期する機能を追加するかもしれませんが(商用アプリケーションのほとんどが行う音声トラックに基づいた同期は、あまり賢いものではありません。撮影時にカメラ間が30メートル離れていることは珍しくありませんが、音の速度は毎秒約340メートルしかなく、これだけで3、4フレームのずれが発生します。これははっきりとわかる違いです)
- 入力動画ストリップすべてに低解像度(25%)のプロキシを作ります。
- 入力カメラをすべて、厳密にひとつのチャンネル内におさめるため、メタストリップを使います。
- 入力チャンネルごとにビューウィンドウを追加し、25%のプロキシ表示モードにします(ウィンドウを重ねて左側にまっすぐ並べることをおすすめしますが、何であれお気に入りのやり方を使ってください)
- 最終出力用の大きなビューウィンドウを追加し、フル解像度で表示させます。
- マルチカムセレクタのエフェクトストリップを、すべてのチャンネルトラックより「上」に追加します。
- 番組全体の再生時間に及ぶようにエフェクトストリップを伸ばします(単にプロパティパネルで Length(長さ)を変更するか右端のハンドルをドラッグします。非常に大きな数字を1つ入力するだけでいいので前者のほうがおそらく簡単です)。
- うまくいくよう祈ります :)(重要なのです :))
- マルチカムストリップを選んでストリップオプション(N)を見てみると、かなり簡単なものであることがわかるでしょう。このエフェクトは選択したチャンネルを入力として解釈します。それだけです。便利なキーボードショートカットが魔法を起こします。マルチカムを選ぶと(テンキーでない側の)1~0のキーは Python ハンドラに対応づけられ、このハンドラはマルチカムセレクタを分割して入力を変更します。
- つまりマルチカムストリップを選んで再生を始め、番組を見ながら1~4のキーを押します。
- 最終的にはカットの数だけ小さなマルチカムセレクターができるでしょう。

要するに: 画面を数秒再生して何が写るか確認したら、もう一度再生して数字キーを使ってストリップを大雑把に分割します。分割部分(=カメラ切り替え)の細かい調整は、隣り合ったマルチカムストリップのハンドルを動かして行います。

[edit] Adjustment Layer

(レイヤー調整)

Adjustment Layer(レイヤー調整)ストリップは、その下にあるすべてのストリップを入力とみなすことを除けば、通常の入力ファイルストリップのように動作します。

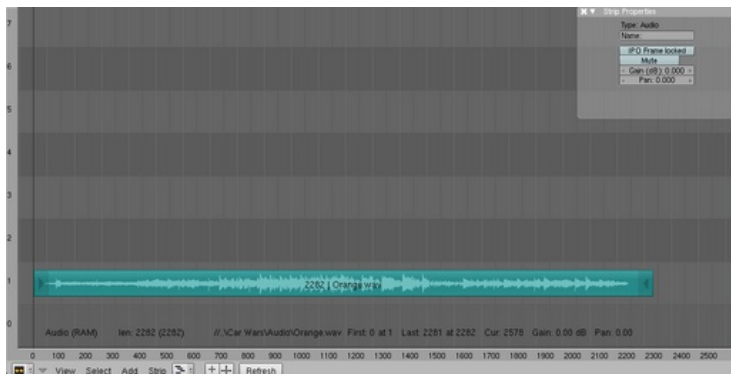
現実の使用事例: メタストリップを使わずに、シーケンサーのタイムラインの一部に仕上げの色調補正を足したいときは、単に一番上に Adjustment Layer を追加して調整するとよいでしょう。

もしくは: 主要な色調補正といくつかの補助的な色調補正を重ね合わせるができます(おそらく、新たに範囲選択用のマスク入力を使います)。

Sound Editing

(音声の編集)

Blender はマルチトラックの音声シーケンサーを備えています。ハードディスクにある WAV、MP3 ファイルを、ファイルとして、または動画内にエンコードしたものとして追加でき、Fカーブで音量を調整してミックスできます。



シーケンスエディタ内の音声ストリップ

[edit] Options

(オプション)

Sound (サウンド) パネルは音声ファイルが選択されたとき、プロパティシェルフ(N)に現れます。

音声ファイルのパスや名前を指定できます。

Pack (パック)

音声ファイルを現在開いている .blend ファイルにパックします。

Caching (キャッシュ)

音声ファイルをデコードして RAM に読み込みます。

Draw Waveform (波形を描画)

ストリップ上に音声ファイルの波形を描画します。

Volume (ボリューム)

音声ファイルの音量を設定します。

Pitch (ピッチ)

音声ファイルのピッチを設定します。

Pan (パン)

音声ファイルのパンを設定します。

Trim Duration: Start/End (期間トリミング)

音声ストリップの開始終了のオフセット。

シー、ポン、パチパチ

キャッシュ使用時にときどきシーという音が入るとい報告が複数のオーディオ好きユーザーからありました。キャッシュを使わないときには発生しないデコードやサンプリング処理が再生時ノイズを起こしています。ポンという音やパチパチという音が聞こえたら、通常はハードウェアがリアルタイム再生に追いつけなくなっている兆候です。最終的にレンダリングされたアニメーションにはこの音は現れないでしょう (Game モードでは現れるかもしれません)。またシーという音はタイムラインで二つ以上の音声ストリップが重なると起きています...

[edit] Audio Mixing in the VSE



(VSE 内での音声ミキシング)

好きな数だけ音声ストリップを使うことができ、結果はそれらすべてをミックスしたのようになります。プロパティシェルフ(N)では、各ストリップの名前やゲイン (dB単位) を設定できます。またストリップのミュートやパン (-1 で一番左、+1 で一番右、パーセンテージでその間を指定します) もできます。

重なりあったストリップはアニメーション処理中に自動的にミックスダウンされます。たとえばチャンネル5にアナウンサーの声、チャンネル6に BGM、チャンネル7に効果音を置くことができます。

[edit] Working with Audio Tracks

(音声トラックの編集)

音声トラック(ストリップ)は VSE の他のストリップと同じ要領で操作できます。中央付近を RMB  でつかめば移動、端にある矢印のハンドルを RMB  すればオフセットの調整、K を押せば分割を行えます。実用的な例として「えー、あー」という声や聴き取れない部分のカットができます。


[edit] Animating Audio Track Properties

(音声トラックプロパティのアニメーション化)

ストリップのプロパティ(N)を介して音量を調整できます。Volume (ボリューム) が 1.0 のとき音量は最大になり、0.0 なら完全に無音です。0.6 は 60% の音量を表します。

アニメーション化させる方法は、他のプロパティと同じです。たとえばフレーム 1 から 100 までの間に Volume (ボリューム) を 0.0 から 1.0 まで上げるのなら:

1. フレーム1にカレントフレームインジケータを合わせます
2. 音声ストリップのプロパティパネル(N)で Volume (ボリューム) を 0.0 にします
3. プロパティ入力欄にマウスカーソルを乗せてIを押します。キーフレームが作られ、プロパティに色がつきます
4. フレーム 100 に移動して、同様にボリューム値 1.0 をキーフレーム化します

デフォルトの Video Editing スクリーンレイアウトでは、左上にグラフィエーターが開いています。追加したキーフレームの位置や補間曲線の編集はここでできます。縦軸はプロパティの値、横軸は時間(フレーム番号)です。Ctrl LMB  で制御点 (=キーフレーム)を追加し、⇔ Tab で曲線を編集できます。

現在のところ FFMPEG の出力だけが音声と動画をひとつの出力ストリームにミックスできます。

音声ストリップの音量をアニメーションさせるとコンポジットの結果に影響します。BGM のフェードイン/アウトや音量の調整に使ってください。重なり合ったストリップはどちらも追加されます。低い番号のチャンネルが高い番号のチャンネルを上書きしたり取り除いたりすることはありません。これで Blender を音声ミキサーにできます。音声トラックを追加して各トラックの音量を曲線で調整すれば、自動化されたダイナミックなマルチトラック音声ミキサーを誰もが手にすることができます。

[edit] Output

(出力)

Render(レンダー) パネルにある Animation(アニメーション) ボタンを使うと出力は動画ファイルになります。音声ファイルを出力する場合は、同パネルの Audio(音声) ボタンを押します。音声シーケンス全体を含む WAV ファイルが作られ、Output(出力) パネルで指定したディレクトリに .WAV 拡張子がつけられて保存されます。動画と音声はあとで外部プログラムを使うか、前のページで説明されている画像シーケンスストリップに追加してミックスできます。

Blender のシーケンスエディタを使う利点は、同一アプリケーション内でフレームと音声を並べるので簡単に同期がとれることです。

タイムラインウィンドウのヘッダーにある Playback(再生) メニューには、音声に関する以下のオプションがあります。

Audio Scrubbing(音声スクラブ)

有効にするとフレームインジケータをマウスの左ボタンでドラッグしたときや任意のウィンドウでフレームを変更したとき、その部分の音声は再生されます。

アクションエディタでフレームの一定範囲をドラッグすれば、特定の音声がいいたいどのあたりで鳴るのかを聴き、そのフレームに対してポーズやシェイプを入力できます。

Audio Muted(消音)

チェックを入れると音声を再生しません。

AV-Sync(AV同期)

3D ビューウィンドウでアニメーションを再生するとき、音声の実際の時間に合わせて必要なら画像フレームを飛ばします。アニメーションのタイミンはだまかになります。

Blender の拡張

慣れ親しんだ他のプログラムと違うかもしれないのは、Blender が一枚岩的でなく発展的であることです。[Python スクリプト](#) を使って、ソースの編集や再コンパイルなしに機能を拡張できます。

アドオン/Addons

アドオンは Blender 内に追加の機能を得られるスクリプトで、ユーザー設定で有効化できます。Blender 実行可能ファイルの外には、たくさんの人が書いた文字通り何百もの addon があります:

- 公式にサポートされたアドオンが Blender に同梱されています。
- テスト中のアドオンには、開発中のビルドには含まれ、公式リリースに含まれないものがあります。多くは確実に動作してとても便利ですが、リリースできるほどの安定性の保証がないものです。

すべての addon の総覧がこの wiki の [Scripts Catalog](#) や [Extensions tracker](#) で利用可能です。

スクリプト/Scripts

addon のほかに、Blender の機能を拡張できるもっと簡単なスクリプトもあります:

- Modules: 他のスクリプトに読み込む汎用ライブラリ。
- Presets: Blender のツールやキーコンフィグの設定
- Startup: Blender の開始時に読み込まれるファイル。Blender のほとんどの UI 定義やさらなる core 操作に使われます。
- Custom scripts: アドオンに比べると主に、[テキストエディタ](#) を通じた一回きりの実行用です。

独自のスクリプトの保存

ファイルの場所

スクリプトはすべて [ローカルの、システムおよびユーザーパス](#) の `scripts` フォルダから読み込まれます。

スクリプトの追加の検索パスは [User preferences](#) (User Preferences (ユーザー設定) » File (ファイル) » File Paths (ファイルパス)) で設定できます。

インストール

アドオンの Blender へのインストールは、User Preferences (ユーザー設定) » Addons (アドオン) ウィンドウから手軽に行えます。Install from File... (ファイルからインストール...) ボタンをクリックして `.py` または `.zip` ファイルを選択します。

スクリプトや addon を手動でインストールするには、`addons`、`modules`、`presets`、あるいは `startup` のうち、種類に応じたディレクトリに置いてください。上記説明もご覧ください。

スクリプトは [テキストエディタ](#) ウィンドウに読み込んで実行することもできます。

Blender 2.6、Python マニュアル

前置き

Blender 2.6 Python マニュアルへようこそ。

Python www.Python.org はインタプリタ型で、対話的に利用できるオブジェクト指向プログラミング言語です。モジュール機構、例外処理、動的な型付け、高レベルの動的なデータ型やクラスが組み込まれています。Python は優れた能力とわかりやすい文法を合わせ持ちます。

Python スクリプトは Blender を機能的に拡張するための強力で万能な方法です。アニメーション、レンダリング、インポートとエクスポート、オブジェクト作成、繰り返し処理の記述など、Blender のほとんどはスクリプトにできます。

スクリプトは Blender とやりとりするために、Blender に密に統合された API (Application Programming Interface) を利用できます。

一般情報

スクリプトを書く上で便利なリンク

- [Blender Python API\(英文\)](#)
 - 公式 API ドキュメントです。スクリプトを書く際にはこちらを参照してください。
- [API introduction\(英文\)](#)
 - Blender の API を利用する際の簡単な説明です。利用例があります。
- [クックブック-コードスニペット](#)
 - 簡単なコード部品を集めたものです(まだ書かれていません)
- [FAQ\(英文\)](#)
 - よくある質問と回答。

スクリプトの配布に関するリンク

- [Sharing scripts\(英文\)](#)
 - スクリプトの共有方法と、Blender の公式ディストリビューションに組み込む方法についての情報。
- [Creating Add-Ons\(英文\)](#)
 - Blender 2.5 Alpha1 で提供される Blender のスクリプトを広める新しい方法です。
- [Extensions project\(英文\)](#)
 - Blender 拡張機能の主要レポジトリのメンテナンスを行うプロジェクト

はじめに - Wiki チュートリアル

以下のページはこの Wiki 上のページであり、Blender スクリプトの基礎から応用までの知識を提供してくれます(全て英文です)。

- [Hello World](#)
- [Console](#)
- [Text editor](#)
- [Geometry](#)
- [Properties, ID-Properties and their differences](#)

はじめに - 外部リンク

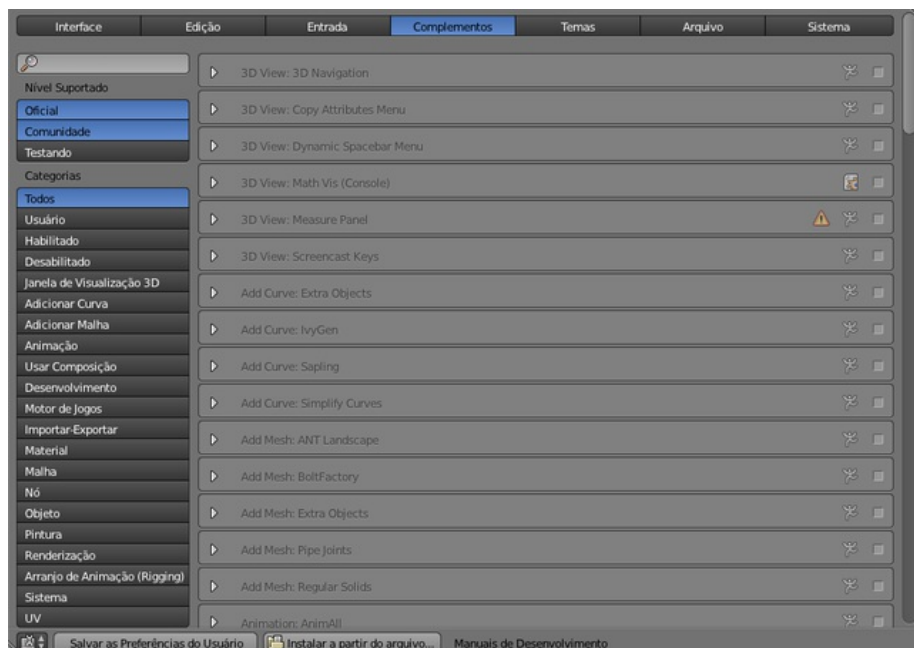
以下のページはこの Wiki 上のページではありませんが、Blender スクリプトを学び始めるに当たって多くの優れた情報を持っています。

- [Introductory tutorial by Satish Goda](#)
 - 基本的な API 操作を始めるにあたっての知識をもたらしてくれます。
- [Ira Krakow's video tutorials](#)
 - 最初の Blender スクリプトに関するビデオ・チュートリアルシリーズです。
- [Quickstart guide](#)
 - Python と Blender についてある程度の知識を持った人向けのクイックスタート・ガイドです。
- [Examples thread](#)
 - 開発中の短いスクリプトの例を多く含むスレッドがあるフォーラム。
- [Introduction to Python](#)
 - 1時間のビデオチュートリアルで、Python と Blender API を紹介します。

アドオン (Add-On)

アドオン (Add-On) は Blender の機能を拡張する補助的なスクリプトの一般的な呼称です。ユーザー・プリファレンス (User Preferences) ウィンドウのアドオン (Add-Ons) パネルで閲覧できます。ここではアドオンのインストールと有効化・無効化を切り替える事ができます。Blender は初期状態でいくつかの便利なアドオンとともにリリースされますが、自身で作成した物、ウェブ上で発見された興味深い物を追加する事もできます。アドオンに関する概要は [Catalog](#) を参照してください。

ファイル・メニューからユーザー・プリファレンスのウィンドウにアクセスするには、CtrlAltU というショートカットを利用するかウィンドウの表示設定を切り替えてください。



アドオンのインストール

アドオン・パネルで表示されるスクリプトは始めにインストールされるべき物です。そのためにアドオン・パネルが表示されたウィンドウのヘッダに有る Install Add-On ボタンを利用できます。一度インストールされると、そのスクリプトはパネル上に表示されます。別の方法として、手動でスクリプトをインストールできます。../blender/scripts/addons というフォルダ (..は各環境において Blender をインストールした場所) にスクリプトを移動させる事でインストールすることができます。(.py という拡張子を持った) スクリプトのファイルをそのフォルダに移動させるだけで十分です。ですが利用する OS によっては (一部のOS では .blender のようにドットで始まるフォルダを標準状態で隠してしまうため) そのフォルダが表示されていない場合があります。

2.5 におけるファイルの場所

- Windows 7 - C:\Users\%username%\AppData\Roaming\Blender Foundation\Blender\2.5x\scripts\addons
- Windows XP - C:\Documents and Settings\%username%\Application Data\Blender Foundation\Blender\2.5x\scripts\addons

Windows 7 環境では AppData フォルダは隠されています。(2.5 以前のバージョンの Blender ではバージョン番号を示す代わりに .Blender という名称のフォルダが利用されていました。)

この場所は OS および Blender の設定によって変わる場合があります。

有効化と無効化

アドオンをインストールしただけではその機能が Blender で利用できるようになるわけではありません。最初に有効化する必要があります。有効化したと考えるアドオンのアドオン有効化 (Enable Add-On) ボタンをクリックするだけです。そうするとアドオンの追加拡張機能が Blender で動作するようになります。再び無効化するには、先に変更したボタンをもう一度押してください。

各種のアドオンに関する更なる情報を得たい場合は、アドオン・パネル上の各要素の左側に表示されている下向き三角形の矢印を押す事で利用可能な情報が表示されます。

もし有効化してもアドオンが利用できない場合は、"コンソール・ウィンドウ (Console Window)" でスクリプトをロードする際に何らかのエラーが発生していないか確認してください。

開発ガイドライン

もしあなたがスクリプト開発者であるなら、[Addons development guidelines](#) を参照できます。

何でここでも Python チュートリアル？

2つ理由があります。

1. Python 3.xを使ったプログラミングを手早く効率的に紹介するためです
2. 一番重要なのは Blender のコンソールの中で教えることで、学習者が文脈を理解できることです。
3. 下記の [外部リンク](#) を見てください

プログラミングって何？

プログラミングは、簡単に言えば「データ操作」以外の何者でもありません。データの編集、もしくは新たなデータの作成操作です。

一番単純なデータは数値です。数値操作には足し算、引き算、掛け算などがあります。これは考えられる限りで最も単純なデータ型です。

しかし現実世界の問題解決には、簡単なデータ(例えば数値)からなる、複数のデータが必要です。Vector データ型は良い例で、3つの数値から構成されています。

このチュートリアルでは、Python 言語が用意しているデータ型にどんなものがあるか、それに、自分のプログラム用に独自の型を作る方法と、そのデータに働きかける操作の書き方を見ていきます。

Python って何？

[Python](#) はインタプリタ型の、インタラクティブな、オブジェクト指向プログラミング言語です。モジュール、例外処理、動的型付け、高レベルの動的なデータ型、クラスを含みます。Python は顕著なパワーと明快な構文を組み合わせます。

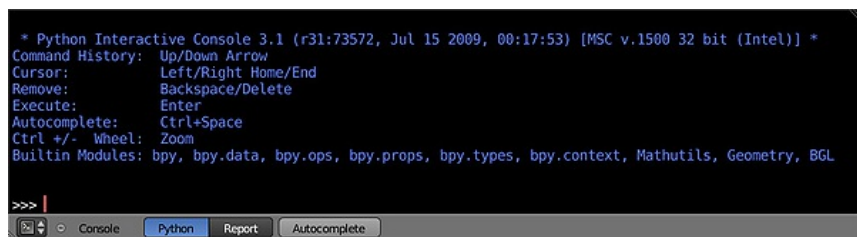
Python の学習はまた、プログラミングをしたことがない方にも非常に簡単です。

Python インタプリタ

このチュートリアルのエクササイズはすべて、Python 3.2 インタプリタを埋め込んである、Blender 2.6 内蔵のコンソールウィンドウを使う予定です。

次のビデオではインタプリタへの切り替え方法をお見せします。

[\[video link\]](#)



インタプリタのプロンプト `>>>` に続けて、Python コマンド、式や文の入力を開始できます。

Hello World

古典的な "Hello World" プログラムから始めてみましょう。

次の `print` 文をインタプリタのプロンプトに入力して `↵` Enter キーを押してください。



上の文を分析してみましょう。

1. "Hello World" は Python の文字列 [\[1\]](#) です。
 1. 文字列は文字(数字、アルファベット、記号)の連なりです。
2. `print()` は Python の組み込み関数で、出力を表示します。
3. `print("Hello World")` は `Hello World` をコンソールに出力します。

エクササイズ

次のコマンドを入力して結果を確認してください。

```
print("Hello World")
print("Hello \n World")
```

Python では、文字列リテラルに `数値` を「掛ける」ことができます。こうすると `数値` で指定した数だけ文字列を繰り返すことになります。

- 数値 * 文字列リテラル
- 文字列リテラル * 数値
- * は Python の掛け算の記号です

```
>>> print("Hello World "*10)
Hello World Hello World Hello World Hello World Hello World Hello W
orld Hello World Hello World Hello World Hello World
>>> |
```

Note

Check out [上のすべての例を1つにまとめて見る](#)

外部リンク

ウェブページ

- <http://www.sthurlow.com/python/>

動画チュートリアル

- Python プログラミングを広く学びたいければ、このチュートリアルを見てみてください。

[\[video link\]](#)

コンソール・エディター・タイプ

Blender 2.5 における対話型コンソールは進化を遂げました。オート・コンプリート機能 (Auto Complete), Python のリポート機能などの機能が追加されました。コンソールにおけるワンライナー (one-liners) のテストは Python API を学習する上で有効な方法です。

内蔵された Python コンソールにアクセスする

マウスを利用してコンソールを起動してください。

[\[video link\]](#)

Blender 上で (3D ビュー、タイムライン等の) エディター・タイプ (Editor Type) であるウィンドウ上で \diamond ShiftF4 というショートカットを利用すると、そのウィンドウ・タイプをコンソール・エディターに変更する事が出来ます。

```
* Python Interactive Console 3.1 (r31:73572, Jul 15 2009, 00:17:53) [MSC v.1500 32 bit (Intel)] *
Command History: Up/Down Arrow
Cursor: Left/Right Home/End
Remove: Backspace/Delete
Execute: Enter
Autocomplete: Ctrl+Space
Ctrl +/- Wheel: Zoom
Builtin Modules: bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, Mathutils, Geometry, BGL

>>> |
```

上図を参照すると、操作に利用していたホットキーが通常とは異なり、CtrlSpace というホットキーを利用するとオート・コンプリート (Auto-complete) が利用できる事に気づかれるかと思えます。

Blender 2.5 は Python 3.1 を利用しており、そのインタプリタがロードされて>>> というプロンプトにコマンドが入力されるのを待っている状態になります。

初めてのコンソール環境

インタプリタ環境に読み込まれているファイルを確認するために、プロンプトで dir() と入力して実行してください。

```
Command History: Up/Down Arrow
Cursor: Left/Right Home/End
Remove: Backspace/Delete
Execute: Enter
Autocomplete: Ctrl+Space
Ctrl +/- Wheel: Zoom
Builtin Modules: bpy, bpy.data, bpy.ops, bpy.props, bpy.types, bpy.context, Mathutils, Geometry, BGL

>>> dir()
['C:', '__builtins__', 'bpy']

>>> |
```

出力される結果の大きな形は以下の通りです。

'C'
Blender にハンドルされた動作に必要な各種情報

'__builtins__'
Python 3.1 のビルトイン (クラス、関数、変数)

'bpy'
トップレベルの Blender Python API モジュール

オート・コンプリート機能

続いて bpy. と入力してください。そして CtrlSpace を押すと実際にオート・コンプリート機能がコンソール上で動作するのを確認できます。

```
app context data ops props types utils
>>> bpy. |
```

bpy 内のサブ・モジュールの一覧が表示されます。これらのモジュールは私達が Blender Python API で出来ることをカプセル化した物で非常に有力なツールになっています。

bpy.app モジュールの内容を表示してみましょう。

```
>>>
binary_path build_date build_platform build_revision build_time build_type debug home n_field
ds n_sequence_fields n_unnamed_fields version version_string
>>> bpy.app.binary_path
'C:\DOCUMENTE-1\Furian\MYDOCU-1\DOWNLO-1\1225_R-1\Release\blender.exe'

version version_string
>>> bpy.app.version
(2, 50, 16)

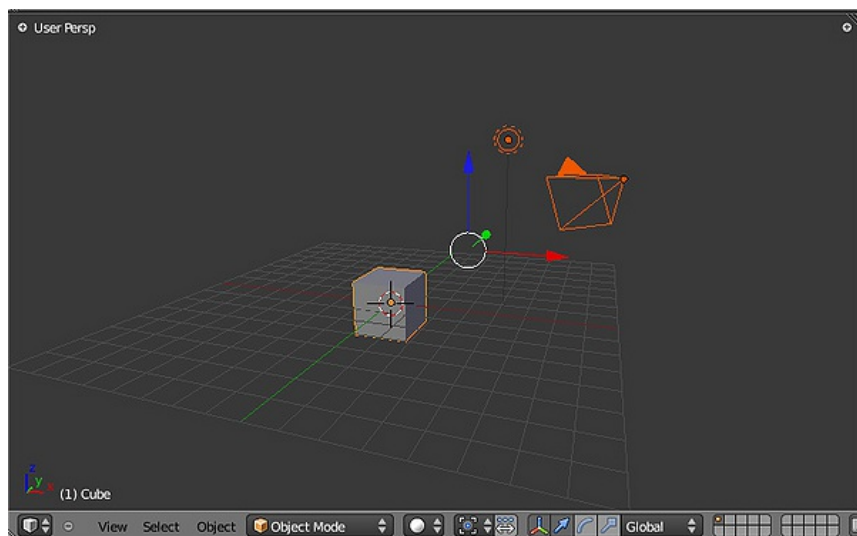
>>> bpy.app.version_string
'2.50 (sub 16)'
```

上図に表示された緑色の出力はオート・コンプリートを利用したときの物である事に注意してください。ここで表示されているのはオートコンプリートによる一覧表示の結果です。一覧上に表示されたのはモジュールの属性名ですが、括弧 '(' で終わっている名称は関数名です。

この機能を利用することが Blender の API をより早く習得する上で大きな助けになります。ここまでの説明でコツは掴めたはずですが、bpy 内のモジュールについてさらに調べてみましょう。

モジュールで試行錯誤する前に

Blender において初期状態の 3D ビューポートを見ると、3つのオブジェクトがある事に気づきます。立方体、ランプそしてカメラです。



- 全てのオブジェクトはコンテキスト上に存在し、操作されている状態に応じた多数のモードを持っています。
- いつもただ1つのオブジェクトのみがアクティブであり、1つ以上のオブジェクトが選択され得ます。
- 全てのオブジェクトは Blender ファイル内のデータです。
- bpy モジュールにはこれらのオブジェクトを作成あるいは調整するオペレーターや関数が含まれています。

上に示した状況 (示されていないものも存在します) では bpy モジュールがデータにアクセスして修正する機能を提供します。

例

bpy.context

Note

以下に示すコマンドによって適切な出力を行うために、3D ビューで選択されているオブジェクトを確認しておいてください。

```
>>>
active_base active_bone active_object active_pose_bone area driver_add( edit_object editable_
bones get( id data is_property hidden( is_property set( items( keyframe insert( keys( main
manager mode object particle edit_object path_resolve( path_to_id( recast_type( region rna_t
ype scene screen sculpt_object selected_bases selected_bones selected_editable_bases selected
_editable_bones selected_editable_objects selected_objects selected_pose_bones space_data textu
re_paint_object tool_settings user_preferences values( vertex_paint_object visible_bones visib
le_pose_bones weight_paint_object window
>>> bpy.context.mode
'OBJECT'

>>> bpy.context.object
[BPype_StructRNA "Object" -> "Cube"]

active_base active_bone active_object active_pose_bone
>>> bpy.context.active_object
[BPype_StructRNA "Object" -> "Cube"]

selected_bases selected_bones selected_editable_bases selected_editable_bones selected_editable_
objects selected_objects selected_pose_bones
>>> bpy.context.selected_objects
[[BPype_StructRNA "Object" -> "Cube"], [BPype_StructRNA "Object" -> "Lamp"], [BPype_StructRNA "Object" ->
"Camera"]]
```

やってみよう!

bpy.context.mode

現在の 3D View の (Object、Edit、Sculpt 等の)モードを出力します。

bpy.context.object および bpy.context.active_object

3D View でアクティブなオブジェクトを返します。

```
>>> bpy.context.object.location.x = 1
```

アクティブなオブジェクトの X 座標を 1 に変更します。

```
>>> bpy.context.object.location.x += 0.5
```

アクティブなオブジェクトの X 座標を 0.5 ユニットだけ移動させます。

```
>>> bpy.context.object.location = [1, 2, 3]
```


アクティブなオブジェクトの X、Y、Z 座標を変更します。

```
>>> bpy.context.object.location.xyz = [1, 2, 3]
```

上に同じです。

```
>>> type(bpy.context.object.location)
```

アクティブなオブジェクトのデータ型を出力します。

```
>>> dir(bpy.context.object.location)
```

表示されるのはアクセスできる大量のデータです。

bpy.context.selected_objects

(アクティブなオブジェクトを含んだ) 選択されている全てのオブジェクトの一覧を返します。

```
>>> bpy.context.selected_objects ( ここで {{Shortcut|Ctrl|Space}} を押す )
```

```
>>> bpy.context.selected_objects[0]
```

対象となる一覧の中で最初のオブジェクトの名前を出力します。

```
>>> [object for object in bpy.context.selected_objects if object != bpy.context.object]
```

分かりにくいのですが、選択されたオブジェクトの内アクティブでないもの全ての一覧を出力します。

bpy.data

bpy.data は Blender のファイル内の全てのデータにアクセスする多くの関数と変数からなります。

開かれている Blender ファイルの以下のデータにアクセスすることができます: オブジェクト (Object)、メッシュ (Mesh)、マテリアル (Material)、テクスチャ、シーン (Scene)、スクリーン (Screen)、音源 (Sound)、スクリプト、テキスト、カメラ、カーブ (Curve)、ランプ、ブラシ、アーマチュア (Armature)、画像 (image)、ラティス (Lattice)、ライブラリ、ワールド (World)、グループ、メタボール、パーティクル、ノード・グループ (Node_group) 等の多くのデータです。

やってみよう!

```
BoolProperty( BoolVectorProperty( CollectionProperty( EnumProperty( FloatProperty( FloatVectorP
property( IntProperty( IntVectorProperty( PointerProperty( StringProperty( actions add_image(
armatures bl_rna brushes cameras curves driver_add( filename get( gpencil groups id_data
images is_property_hidden( is_property_set( items( keyframe_insert( keys( lamps lattices lib
raries materials meshes metaballs node_groups objects particles path_resolve( path_to_id( r
ecast_type( rna_type scenes screens scripts sounds texts textures values( vfonts window_ma
nagers worlds
>>> bpy.data
[BPy_StructRNA "Main"]
>>> bpy.data.objects
[BPy_PropertyRNA "Main" -> "objects"]
>>>
>>> for object in bpy.data.objects:
...     print(object.name + " is at location " + str(object.location))
...
Camera is at location [7.481132, -6.507640, 5.343665](vector)
Cube is at location [0.000000, 0.000000, 0.000000](vector)
Lamp is at location [4.076245, 1.005454, 5.903862](vector)
>>> |
```

エクササイズ

```
>>> for object in bpy.data.scenes['Scene'].objects: print(object.name)
```

上記のように入力し、`↵` Enter を2回押します。'Scene' という名前を持った Blender のシーンに属する全てのオブジェクトの名前を出力します。

```
>>> bpy.data.scenes['Scene'].objects.unlink(bpy.context.active_object)
```

'Scene' という名前を持った Blender のシーンでアクティブなオブジェクトのリンクを解除します。

```
>>> bpy.data.materials['Material'].shadows
```

```
>>> bpy.data.materials['Material'].shadows = False
```

(要編集)

bpy.ops

Blender 2.5 におけるオペレータ (Operator) となるように開発されたツールあるいは機能のシステムです。これらのオペレータはコンソールから直接、もしくはボタンをクリックする、あるいは Python スクリプトのパッケージから呼び出すことが出来ます。これらは非常に強力です。

オペレータが扱うカテゴリの一覧を閲覧するには、[ここ](#)をクリックしてください。

3D ビューポート (Viewport) 上に5つの立方体の集まりを作成してみましょう。最初に、標準で存在する立方体オブジェクトを選択して X を押し、削除してください。

やってみよう!

以下に示すコマンドはレイヤー 1 に作成されたオブジェクトを指示する物です。まず後に参照するための配列を定義します。

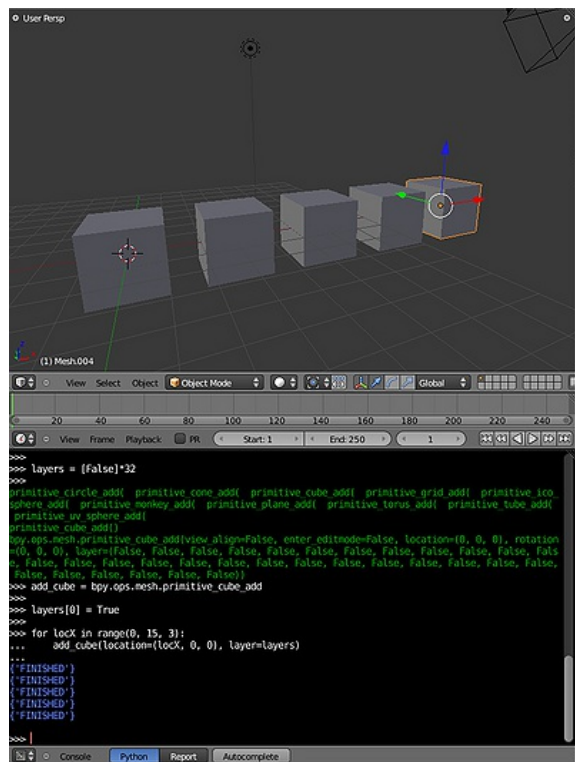
```
>>> mylayers = [False]*20
>>> mylayers[0] = True
```

オペレータが基本メッシュである立方体を作成する参照元を作成します。


```
>>> add_cube = bpy.ops.mesh.primitive_cube_add
```

ループ内で下のように記述し5つのオブジェクトを作成します。(スクリーンショットの上部では下に示すのとは異なるメソッドが用いられています。) コンソールのシェル・プロンプトにコマンドを入力 (2行目の頭の空白も入力) した後、2回エンターキーを押してください。

```
>>> for index in range(0, 5):
...     add_cube(location=(index*3, 0, 0), layers=mylayers)
```



Text Editor

Text Editor は Blender のウィンドウの種類の一つで、Window type メニューの Text Editor ボタン ( Text Editor)、もしくは ⇧ ShiftF11 でアクセスできます。

新たに開いた Text window は灰色で空っぽで、とてもシンプルなツールバー (*Text Toolbar*) が付いています。



Text Toolbar。


左から右に向かって、標準の Window type 選択ボタンとウィンドウメニューがあります。そして Text ID Block 一覧のボタン、その隣に新規テキストファイルを作るための New (新規) ボタンが続きます。一旦クリックするとツールバーが変わり、変わったままになります。



ファイルを開いた状態の Text Toolbar

これで、テキストファイルの名前変更のためのテキストボックスと、その隣に新規ファイルを作るための + ボタンができます。text block を取り除くには X ボタンをクリックしてください。

続く3つのボタンはそれぞれ、行番号、テキストの折り返し、構文ハイライトのオン・オフを切り替えます。

キーボードで文字を打つと、テキストバッファにテキストが現れます。いつものように LMB  を押ししたりドラッグしたり離したりしてテキストを選択できます。

次のキーボードコマンドを使えます:

- CtrlC - 選択テキストをコピーしてテキストのクリップボードに入れます。
- CtrlX - 選択テキストをカットしてテキストのクリップボードに入れます。
- CtrlV - テキストウィンドウのカーソル位置に、クリップボードからテキストを貼り付けます。
- ⇧ ShiftCtrlAltS - 未保存ファイルをテキストファイルとして保存します。File Browser ウィンドウが表示されます。
- AltS - すでに開いているファイルを保存します。
- AltO - テキストを開きます。File Browser ウィンドウが表示されます。
- AltP - テキストを Python スクリプトとして実行します。
- CtrlZ - 元に戻します。
- Ctrl⇧ ShiftZ - やり直します。
- AltR - 現在のバッファを開き直し(再読み込み)します (未保存の編集内容は失われます)。
- AltM - テキストウィンドウの内容を 3D text に変換します (最大 100文字)。

テキストバッファを削除するには、マテリアルなどの操作と同じで、単にバッファ名の隣にある X ボタンを押してください。

もっとも注目に値するキーは AltP で、バッファの内容を Blender 内蔵の Python インタプリタに実行させます。次のページで、Python スクリプティングの例をお見せします。先に進む前に知っておくに値するのは、Blender が組み込みの完全版 Python インタプリタや、[APIリファレンス](#) に記載のあるような多くの Blender 独自のモジュールを備えていることです。

Text Editor は Python スクリプト専用の機能も持っていて、クラス/関数/変数ブラウザ、補完...など、記述に役立つツールを追加します。これらの機能には Toolshelf(ツールシェルフ) (CtrlT) からアクセスできます。

テキストウィンドウの他の用途

テキストウィンドウは、.blend ファイルをコミュニティや友人と共有したいときにも役に立ちます。blender ファイルの内容を説明した README テキストの記述に使えます。別々のアプリケーションで作って保存するよりずっと便利です。テキストウィンドウが見えている状態で保存することを忘れずに！ ファイルのあるライセンスのもと、コミュニティで共有するときは、テキストウィンドウにそのライセンスを記述できます。

デモンストレーション

[\[video link\]](#)

エクササイズ

次のテキストをテキストウィンドウにコピーしてください。

```
import bpy
from math import radians, cos, sin

# オブジェクトは 20 までのレイヤーに存在できます。
# 次のコードはどのレイヤーに配置したいかを決定します

# カーソル位置を取得します
cursor = bpy.context.scene.cursor_location

# 円の半径
radius = 5

# 間隔を置いて立方体を円周上に配置します。デフォルトは36度間隔です。
# 角度をラジアンに変換したリストを取得します

anglesInRadians = [radians(degree) for degree in range(0, 360, 36)]

# 角度リストをループし、極座標を使って x,y を求め、
# オブジェクトを作ります
```

```
for theta in anglesInRadians:  
    x = cursor.x + radius * cos(theta)  
    y = cursor.y + radius * sin(theta)  
    z = cursor.z  
    bpy.ops.mesh.primitive_cube_add(location=(x, y, z))
```

スクリプトを AltP で実行してください。

上のスクリプトの実行結果を次の動画で見られます。

[\[video link\]](#)

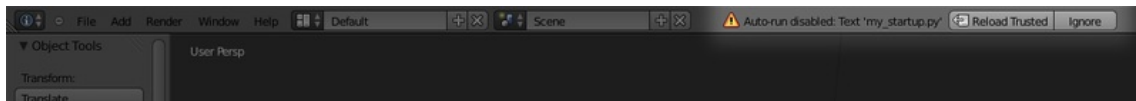
スクリプティングとセキュリティ

blend ファイル内に Python スクリプトを同梱する機能は、リギングや自動化、ゲームエンジンの使用といった上級の作業に便利です。しかし、Python がスクリプトで可能なことを制限していないため、セキュリティリスクを引き起こします。

そのため、皆さんの知っている、または信頼しているソースのスクリプトのみ実行するようにしてください。

自動実行はデフォルトでは無効になっていますが、いくつかの blend ファイルを正しく機能させるには必要になります。

blend ファイルがスクリプトを実行しようとするがそれが不可能な時、ヘッダーにメッセージと共に、**信頼してリロード (Reload Trusted)**と**無視 (Ignore)**オプションが表示されます。



Blend ファイル内のスクリプト

自動実行

blend ファイルが自動的にスクリプトを実行する方法はいくつかあります。

- 登録済みテキストブロック
各テキストブロックは**登録 (Register)** オプションを ON することで、読み込み時に実行できます。
- アニメーションドライバー
Python 式を値の操作に使用できます。これはさらに上級のリグやアニメーションでよく使用されます。
- ゲームエンジンの自動スタート
スクリプトはよくゲームロジックに使用されており、自動スタートを ON にした blend ファイルは読み込み時にゲームを実行できます。

手動実行

blend ファイル内のスクリプトを他の方法で実行するにはユーザーによる干渉が必要です(その結果、自動実行が OFF の時でも実行可能です)。ただし、これは必ずしも明白ではないことを承知しておいてください。

- テキストエディター内でのスクリプトの実行(OK、これは常識です！)。
- Freestyle によるレンダリング - FreeStyle はラインスタイルの操作にスクリプトを使用します
- ゲームエンジンの実行。

スクリプト実行のコントロール

Blender には blend ファイルのスクリプトを自動的に実行可能とすることをコントロールする方法がいくつかあります。

まず、ファイルセレクターに**信頼できるソース (Trusted Source)** オプションがあります。これは自動実行を状況に応じてコントロールするのに利用できます。

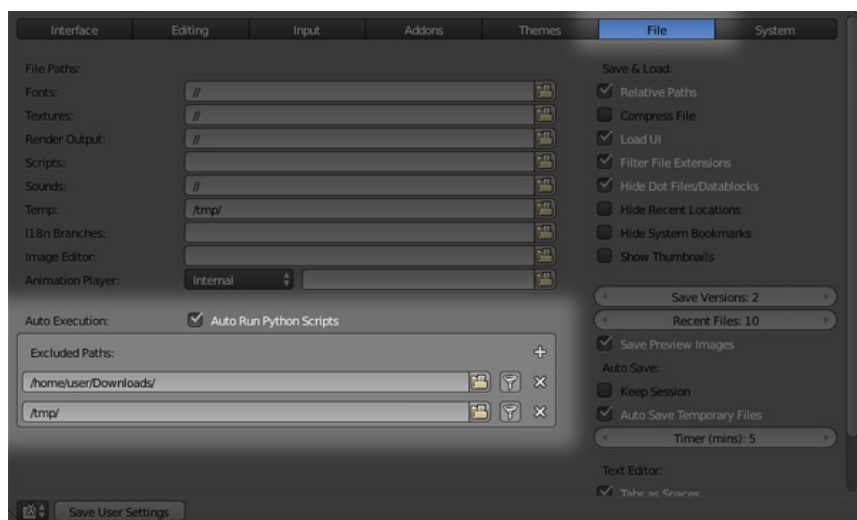
しかしこれをセットするのを忘れて、ファイルセレクタ以外でファイルを開くかもしれません — そのために、デフォルトを変更しましょう(次で解説)。

デフォルト設定

ユーザー設定(User Preference)の**ファイル(File)**タブに、**Pythonスクリプトの自動実行 (Auto-Run Python Scripts)** トグルボタンがあります。

これはファイルセレクタ内の**信頼できるソース (Trusted Source)** オプションをデフォルトで ON にし、blend ファイルをファイルセレクタを使用せず読み込んだ際にスクリプトを実行可能にすることを示しています。

有効にすると、一部のディレクトリを除外するオプションが表示されます。ダウンロードディレクトリを除いた、すべてのパスを信頼するのが典型的な設定です。



コマンドライン

バッチレンダリングやその他コマンドラインで行う(Blender をインターフェイスなしで実行させる)作業を行いたい場合があると思います。

この場合もユーザー設定(User Preference)を使用できますが、オーバーライドすることもできます。

- -y または `--enable-autoexec` で有効
- -Y または `--disable-autoexec` で無効

例: アニメーションをバックグラウンドモードでレンダリングし、ドライバーとその他スクリプトを実行可能にする

```
blender --background --enable-autoexec my_movie.blend --render-anim
```

注釈: これらのコマンドライン引数は通常の Blender インスタンスの起動時にも使用可能で、この場合もユーザー設定をオーバーライドします。

Blender の Python API

Blender の完全な Python API(アプリケーションプログラミングインターフェイス)は下記にドキュメント化されています。

[最新の API](#) - おそらく現在の安定版リリースよりは新しいです！

バージョン別:

[2.64 API](#)

[2.63 API](#)

[2.62 API](#)

[2.61 API](#)

[2.60a API](#)

[2.59 API](#)

[2.58 API](#)

[2.57 API](#)

[2.56 API](#)

スクリプト

100以上の Blender 用のスクリプトがネット上で利用可能です。

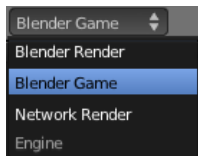
プラグイン同様、スクリプトは非常に流動的で、インターフェイスや機能、Web上の場所が本当に早く変化します。そのため、更新済みのリストや生きているリンクは、下記の主要な Blender サイトを参照してください。

- www.blender.org
- www.blenderartists.org
- [この wiki 上の Python extensions](#)

Blender Game Engine 2.5 概要

Blender はインタラクティブな 3D アプリケーションの開発を可能にするゲーム・エンジン (Game Engine) を内蔵しています。Blender ゲーム・エンジン (以下BGE) はパワフルでハイレベルな開発ツールです。ゲーム開発を主たる目的としていますが、インタラクティブな 3D 構造ツアーや教育における物理の研究のような、インタラクティブな 3D ソフトウェアの開発にも利用できます。

標準状態では Blender のレンダー・エンジンは "Blender Render" となっており、ゲームエンジンを利用するためにはレンダー・エンジンを "Blender Game" に切り替える必要が有ります。



もしレンダー・エンジンを "Blender Game" に切り替えていない場合、"Bullet Physics" の設定タブのような、幾つかの機能を利用できません。

Game Engine を利用する

BGE の根幹を成す構造は [ロジック・ブリック \(Logic Bricks\)](#) に示されています。ロジック・ブリック (Logic Bricks) の目的はいかなるプログラミング言語の知識も使う事無く用意にインタラクティブなアプリケーションの実装を容易に行えるようにする事です。ロジック・ブリック (Logic Bricks) には、センサー (Sensors) 、コントローラー (Controllers) 、アクチュエーター (Actuators) という3つの種類が存在します。それぞれに関する詳細は以下に示します:

- [センサー \(Sensors\)](#)
- [コントローラー \(Controllers\)](#)
- [アクチュエーター \(Actuators\)](#)

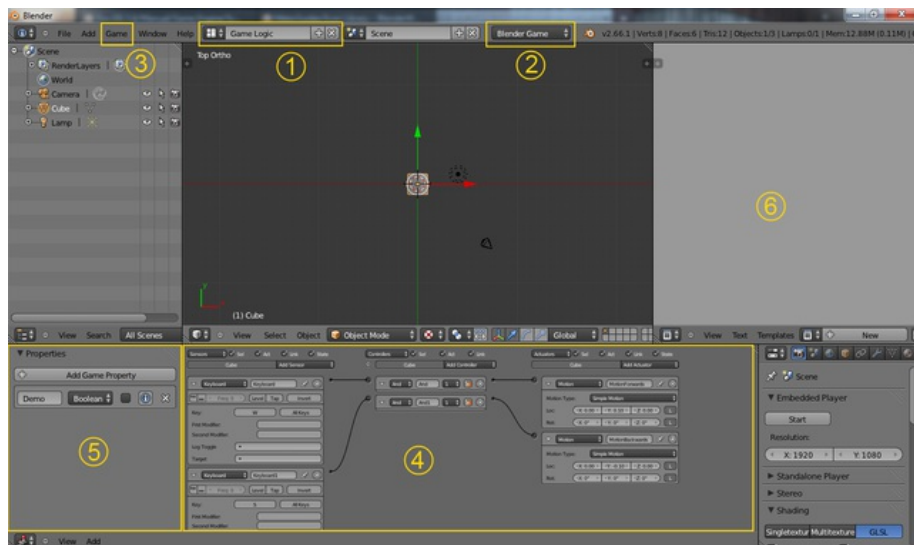
Python を用いたゲームの記述を好まれる場合でも、ゲーム・エンジンは Blender の他の実装から独立した [python api 2.56.0 Python API](#) を持っており、ゲーム操作を行うためのスクリプトを記述する際に利用できます。これは [Python Controller](#) の作成と、その Python Controller を Python スクリプトにリンクする事で可能になります。

Game Logic Screen Layout

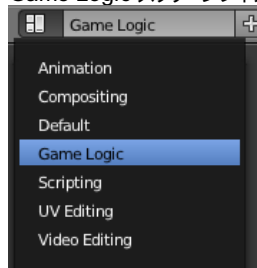
(「Game Logic」スクリーンレイアウト)

ゲームの設計、構築、デバッグ、実行には Blender の機能が幅広く使われます。この作業を助けるため、Blender は BGE ゲーム構築のためのスクリーンレイアウトを持っています。慣れ親しんでいるパネルの他に、BGE のみに関係する新たな [Logic Editor \(eng\)](#) パネル④があります。

下の図はデフォルトの Game Logic スクリーンレイアウト並びにゲームの配置/デバッグ/実行のためのオプションを示しています(ここに示した順序で表示されるはずです)。



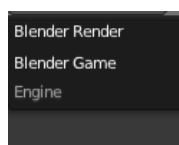
Game Logic スクリーンレイアウト



Game Logic メニュー

① Game Logic

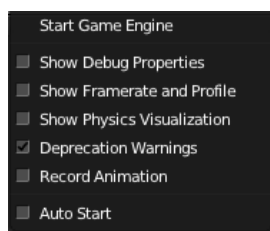
さまざまな用途のスクリーンレイアウトの一覧から選びます。Information、3D View、Properties といったすでに慣れ親しんでいるパネルが多くありますが、BGE 専用の新たな Logic Editor パネルもあります。



Render Engine メニュー

② Blender Game

レンダリングエンジンのメニューから選びます。すべての出力がリアルタイムに Blender Game Engine レンダーによって行われます。Game オプション(下記参照)や BGE レンダーのプロパティ領域(下記参照)などの種々のメニューオプションも開かれます。



Game オプション

③ Game

このメニューを使って Game Engine の実行状況を示す種々のオプションを選べます。

このメニューはレンダリングエンジンが Blender Game に設定されているときにだけ利用できることに注意してください。

Start Game

Game Engine でゲームを実行します (ショートカットはマウスカーソルが 3D View ウィンドウの上にあるときに p または ⇧ ShiftP です)。

Show Debug Properties

デバッグ用に印をつけたプロパティをゲーム実行中に表示します

Show framerate and profile

フレームレートおよび分析用の情報をゲーム実行中に表示します

Show Physics visualization

物理演算の接触範囲と接触を視覚化して表示します

Deprecation warnings

Python API の 非推奨の機能を使った時、警告を表示します

Record animation

アニメーションを F-カーブ に記録します

Auto Start

ロード時にゲームを自動開始します

④ Logic Editor panel

[Logic Editor \(eng\)](#) はゲーム内オブジェクトの挙動操作のため、[ロジック](#)、[プロパティ](#)、[ステート](#) を組み立てる場所です (Logic Editor パネルは Display Editor メニューにある Logic Editor を選ぶか、⇧ ShiftF2 や Ctrl→ を押しでも表示されます)。

⑤ Properties



同じ語の二つの意味

Blender の用語として、「プロパティ」という語は二つの違った意味で使われることに注意してください。広く使われるのはこの下で説明している プロパティ表示パネルを指すものです。もう一つは Game Engine のロジック変数であるプロパティを指しています。

プロパティパネルはいつものように Information メニューから選びます。しかしながら、レンダリングエンジン(②)が Blender Render から Blender Game に切り替わると、プロパティパネルの一部が変化することに注意してください。

詳しくは [Physics](#) の、プロパティパネルに関する記述をご覧ください。

Logic, Properties and States

(ロジック、プロパティ、ステート)

ゲームロジックは、ゲームエンジン内のデフォルトのスクリプト記述階層です。ゲーム内の `GameObject` はそれぞれ、シーン内での動作を制御するためのロジカル要素 (ロジックブリック) を保管できます。ロジックブリックは組み合わせて、シミュレーションの進行を決めるようなユーザー定義動作を行わせることができます。

Logic Bricks

(ロジックブリック)

ゲームロジックの主要部分はグラフィカルなインターフェースである [Logic Editor](#) を通して組み立てられ、プログラミングの詳細知識を必要としません。ブロック (または “ブリック”) は、あらかじめプログラムされた機能を表し、ゲーム/アプリケーションを作成するために微調整して組み合わせることができます。システムは3つの部分に分割されます: [Sensors](#)、[Controllers](#) および [Actuators](#) です。センサーは、衝突、キー押下、マウス操作などの物事の発生を感知します。コントローラーはセンサーの出力に対して論理的な操作を実行し、接続されているアクチュエータを、発動条件が整ったときに起動します。アクチュエータはシミュレーションと直接相互作用します。ゲーム内でこれができるのはアクチュエータだけです (Python コントローラー、および物理演算のような他のシミュレーション要素を除く)。

Properties

(プロパティ)

[プロパティ](#) (属性) は他のプログラミング言語における変数のようなものです。ゲーム全体を通じて、もしくは特定のオブジェクト/プレイヤー (名前など) 用に使われる、値の保存や取得に使われます。しかしながら、Blender Game Engine ではプロパティはオブジェクトと関連づけられています。プロパティにはさまざまな型があり、[Logic Editor](#) の専用の領域で設定できます。

States

(ステート)

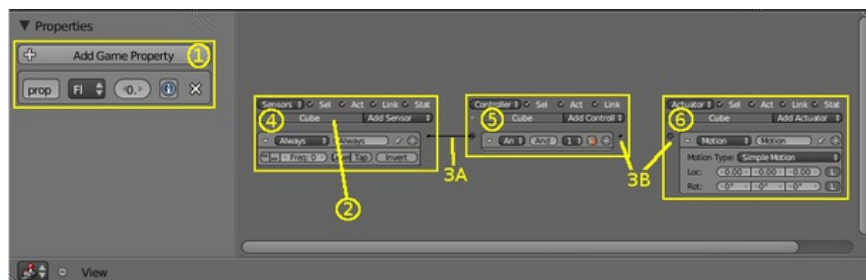
もう一つの便利な機能がオブジェクトの [ステート \(eng\)](#) (状態) です。シミュレーション実行中はいつでも、オブジェクトのロジックのうち、現在のステートに属するものがすべて実行されます。ステートは、動作のグループわけに使うことができます。例えばあるオブジェクトのステートが「寝ている」「起きている」「死んでいる」のいずれかであるとするれば、この三つの状態それぞれに、違った動作をするロジックを用意できます。オブジェクトのステートは、そのオブジェクトのコントローラーのロジックブリックで作成し、表示し、編集します。

Logic Editor

ゲームを構成するさまざまな俳優たち(=オブジェクト)用の、ゲームロジックの作成、編集作業の中心となるのが Logic Editor です。

関連する 3D パネルで選択中のオブジェクトのロジックが、logic brick として表示されます。センサー、コントローラー、アクチュエータの三つの列を持つ表として表示されます。ロジックブリックを繋いでいるリンクは、センサー-コントローラー間、コントローラー-アクチュエータ間に信号を伝えます。

ロジックパネルをより良く理解してもらえるように、メニューの一部を拡大して番号をつけたものが、下の画像です。下の画像に付けられた番号を追いながら、個別に各セクションを見ていきます。



Logic Panel の構成部品

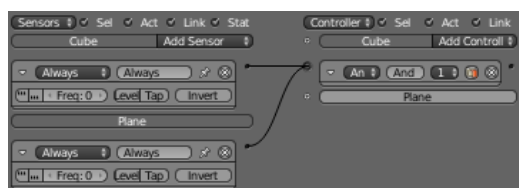
1 Properties

プロパティは、他のプログラミング言語における変数に似ています。これらは、オブジェクトに関連付けられたデータを保存したり、アクセスしたりするために使用されます。以下の種類があります：

- Timer - 定義された番号から始まり、カウントアップします。
- String - テキストを保持します。
- Float - -10000.000から10000.000の小数を保持します。
- Integer - -10000～10000の整数を保持します。
- Bool - TrueまたはFalseのいずれかです。

より詳細な説明については、[Properties](#)を参照してください。

2 Associated Object(s)



Logic for several objects.

Blender game engine のロジックは、オブジェクトに結び付けられます。オブジェクトは、その名前順に一覧表示され、それらが選択されている場合、ロジックウィンドウに表示されます。あなたは上の図のように単一のオブジェクトを選択するか、右の図のように複数のオブジェクトを選択することができます。

オブジェクトのロジックは、そのロジックに関連付けられているオブジェクト(s)が選択されているときのみ表示されます

3 Links

リンク(3A)は、オブジェクト間の論理的な流れの方向です。リンク線は LMB (マウス左ボタン) で一方のリンクのノード(3B)から他方へドラッグすることで描かれます。リンクは、センサからコントローラへ、もしくはコントローラからアクチュエータへにだけ描画することができます。送信ノード(センサとコントローラの右側にある黒い丸)は、複数の受信ノード(コントローラとアクチュエータの左側にある)に送信することができます。受信ノードは、同様に複数のリンクを受け取ることができます。

あなたが直接アクチュエータにセンサをリンクすることはできません。アクチュエータは、センサに戻るほうにリンクすることはできません。もしアクチュエータが完了した後にセンサーを起動させたい場合は、アクチュエータのセンサーを使用してください。

4 Sensors

センサーは、すべてのロジックのアクションを開始します。センサーは、オブジェクトの接近や、キーボードのキーの押下や、時限イベント等々を“感知”します。センサーがトリガされると、パルスがすべてリンクされているコントローラに送信されます。

より詳細な説明については、[Sensors](#)を参照してください。

5 Controllers

コントローラは、ロジック、センサーからのパルスの評価、およびそれに応じてアクチュエータへのパルスの送信を扱います。コントローラにはさまざまな種類があります：

- AND - すべての接続されたセンサーは正のパルスを送信するため、正の値でなければなりません。
- OR - 1つ以上の接続されたセンサーが正である必要があります。

- XOR - 排他的論理和 : どちらか片方だけ、接続されたセンサーが正である必要があります。
- NAND - 否定論理積、ANDのビット反転のコントローラ。
- NOR - 否定論理和、ORのビット反転のコントローラ。
- XNOR - 排他的NORコントローラ。
- Expression - あなた自身の表現をお書きください。
- Python - Pythonスクリプトやモジュールのセンサを制御します。

より詳細な説明については、[Controllers](#)を参照してください。

6 Actuators

アクチュエータは、オブジェクトまたはゲームに何らかの方法で影響を与えます。アクチュエータは動き、音、プロパティ、オブジェクト、などを変更します。これらの変更は、他のオブジェクト、物理学、プロパティなどにも及び、またそれらは他のロジックブリックへのトリガイベントを発生させることもできます。

より詳細な説明については、[Actuators](#)を参照してください。

センサ

センサはロジックが何かをする要因となるものです。これはオブジェクトに近づいたり、キーが押されたり、時間によってイベントが起動したり、などのトリガイベントです。センサが起動すると、リンクしているすべてのコントローラに正のパルスが送られます。

どのタイプのセンサー用ロジックブロックも [Logic Editor](#) を使って構築し、変更します。この作業の詳細は [センサーの編集](#) ページをご覧ください。

現在、以下のセンサーが利用できます:

Actuator	特定アクチュエータが実行信号を受け取ったことを検知
Always	一定間隔で信号を出し続けます
Collision	オブジェクト間またはマテリアル間の衝突を検知
Delay	指定した論理ティック値だけ出力を遅延させます
Joystick	ジョイスティックの特定操作を検知
Keyboard	キーボード入力を検知
Message	テキストメッセージかプロパティ値を検知
Mouse	マウスイベントを検知
Near	一定距離内に進入したオブジェクトを検知
Property	所有オブジェクトのプロパティ変化を検知
Radar	一定距離内、軸から一定角度内に進入したオブジェクトを検知
Random	ランダムにパルスを生成
Ray	軸角方向に光を放射して衝突を検知
Touch	オブジェクトと他オブジェクトとの接触を検知

Sensor Editing

(センサーの編集)

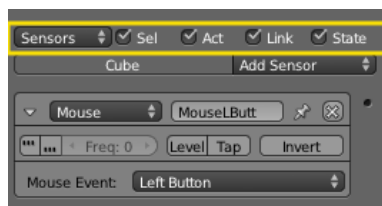


Sensor Column with Typical Sensor

センサはロジックパネルの左の列で編集します。

Column Heading

(列見出し)



Sensor Column Heading

Sensorsサブパネルの一番上にはSel、Act、Link、Stateの4つのボタンがあります。またSensorsサブパネルのタイトルSensorsをクリックするとメニューが現れます。これはセンサの表示・非表示を切り替えるもので、必要なセンサだけを見るのに便利です。

Sensorsメニューはラベルのようですがメニューです。4つのオプションがあります。

Show Objects	すべてのオブジェクトを展開する。
Hide Objects	すべてのオブジェクトを名前の表示されるバーのみに縮小する。
Show Sensors	すべてのセンサを展開する。
Hide Sensors	すべてのセンサを名前の表示されるバーのみに縮小する。

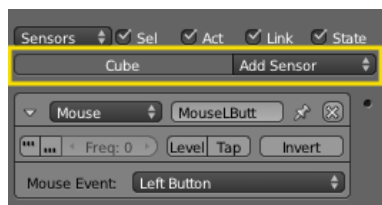
センサとオブジェクトの表示は別々にコントロールできます。

また表示するセンサの種類を指定できます。

Sel	選択されているすべてのオブジェクトのセンサ。
Act	アクティブなオブジェクトのセンサ。
Link	コントローラにリンクされているセンサ。
State	アクティブな状態を持つコントローラに接続されているセンサ。

Object Heading

(オブジェクト見出し)



Sensor Object Heading

センサは選択されたオブジェクトごとに表示されています。デフォルトでは、選択中のオブジェクトのすべてのセンサーが列内に並びますが、列見出しにある抽出機能を使って変更可能です。

センサー一覧の見出し部分には二つの項目があります:

Name

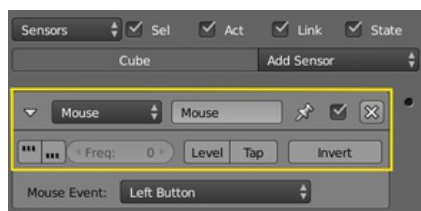
オブジェクト名

Add Sensor

クリックすると利用できるセンサーの種類を選ぶメニューが現れます。センサーを選ぶと新たなセンサーがオブジェクトに追加されます。利用できるセンサーについては [Sensors](#) をご覧ください。

Sensor Common Options

(センサー共通のオプション)



Common Sensor Options

すべてのセンサに共通のボタン、入力欄、メニューがあります:

三角形のボタン

センサの詳細を表示／非表示します。

センサーの種類 メニュー

[センサの種類](#)を参照。

センサー名

センサの名前。Python でセンサにアクセスする際に使います。そのオブジェクトのなかで唯一の名前である必要があります。

X ボタン

センサを削除します。

起動信号 (Trigger) について

コントローラーは、接続されているセンサー (状態は任意) によって起動されない限り作動することはありません。

センサーの状態が変化すると、接続されているコントローラーが起動します。センサの状態が負から正、または正から負へ変わると、センサーは接続されているコントローラーを起動します。センサーが非アクティブ状態からアクティブ状態になるときも、接続されているコントローラーを起動します。

以下のパラメータは、センサーに接続されたコントローラーの起動方法を指定します:

☐☐☐ (True level triggering)

これをセットすると、センサの状態が正である限り、真のパルスを出し続けます。パルスの間隔は周波数で設定します。

☐☐☐ (False level triggering)

これをセットすると、センサの状態が負である限り、偽のパルスを出し続けます。パルスの間隔は周波数で設定します。

Freq (周波数)

周波数という名前に反して、このパラメータはパルスを送る時間間隔を指定します。単位はフレームです (論理 tick と呼ぶこともあります)。デフォルトの値 0 の場合は、まったく間隔をあげません。Level 起動のパラメータが少なくとも一つ有効な場合にのみ使われます。パルスがそれほど頻繁でなくてもいい場合は、この数値を上げておくとパフォーマンスの節約になります。

例: (デフォルトのフレームレートを 60 Hz (60 フレーム/秒) と仮定します)。

freq	意味	起動信号を持つフレーム	起動信号を持たないフレーム	フレーム周期	周波数 (フレーム/秒)
0	センサーは次のフレームで起動信号を送ります。	1	0	1	60
1	センサーは第1フレームで起動信号を送り、再度起動信号を送るまでに1フレーム待ちます。結果として半分の速度になります。	1	1	2	30
29	センサーは第1フレームで起動信号を送り、再度起動信号を送るまでに29フレーム待ちます。	1	29	30	2
59	センサーは第1フレームで起動信号を送り、再度起動信号を送るまでに59フレーム待ちます。	1	59	30	1

Level ボタン

組み込みのステートマシンの状態 (State) が変化するときパルスを送ります。状態については [States](#) をご覧ください。

以下のパラメーターはセンサーの状態算出の方法を決めます:

Tap ボタン

センサーの状態を正に変え、(センサーの算出値が正であったとしても) 1 フレーム後に負に変えます。状態が変わるため、接続されたコントローラーは (Level と) 同じように起動します。Tap と Level はどちらか一方だけを選択できます。TRUE level triggering を設定している場合、センサーの算出値が偽になるまで真／偽のパルスが交互に送られ続けます。Tap が設定されていると FALSE level triggering は無視されます。

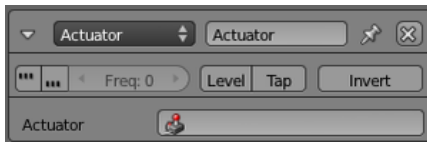
Invert ボタン

センサの出力を反転します。

真のパルスを送るべきときに偽のパルスを、偽のパルスを送るべきときに真のパルスを送ります。ただし、Tap が設定されている場合は、センサーは反転したセンサーの状態に基づいて、コントローラーに起動信号を送ります。

Actuator sensor

(アクチュエータセンサー)



Actuator sensor

Actuator欄に指定したアクチュエータが起動すると正のパルスを発信します。

指定したアクチュエータが非アクティブ化すると、センサーは偽のパルスを送ります。

[センサー共通のオプション](#) もご覧ください。

専用のオプション:

Actuator

アクチュエータ名。同じオブジェクトに属している必要があります。

Always sensor

(常時センサー)



Always sensor

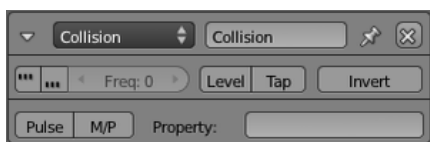
一定の論理 tick ごとに発信する必要がある場合、またはスタートアップ時に発信させたい場合 (Tapを有効にする) に使います。

[センサー共通のオプション](#) もご覧ください。

このセンサには共通オプション以外の設定項目はありません。

Collision sensor

(衝突センサー)



Collision sensor

Touch センサに似ていますが、プロパティやマテリアルで抽出できます。指定したプロパティ/マテリアルを持つオブジェクトが衝突した場合のみ正パルスが送られます。どのオブジェクトに対しても動作させるには欄を空のままにします。

[センサー共通のオプション](#) もご覧ください。

専用オプション:

Pulse ボタン

直前のパルスの要因となったオブジェクトとまだ触れているときにも、他のオブジェクトとの衝突を検出します。

M/P ボタン

フィルタリングする要素をマテリアルとプロパティで切り替えます。

Delay sensor

(遅延センサー)



Delay sensor

指定したticksだけ反応を遅らせます。これは他のアクションが先に起こるべきである場合や、時間によって起こるイベントの場合に便利です。

[センサー共通のオプション](#)もご覧ください。

専用オプション:

Delay

正のパルスを送る前に待つ論理 ticks 数

Duration

負のパルスを送る前に待つ論理 ticks 数

Repeat ボタン

遅延時間が終わった後、再始動します

Joystick sensor

(ジョイスティックセンサー)



Joystick sensor

ジョイスティックの動きやジョイスティック付随するイベント(ハット、ボタンなど)を検出します。ジョイスティックは複数使用できます(「Index」をご覧ください)。ジョイスティックの操作ボタン等の正確な配置は使用するジョイスティックの種類や形式に左右されます。

共通オプションは [センサー共通のオプション](#) をご覧ください。

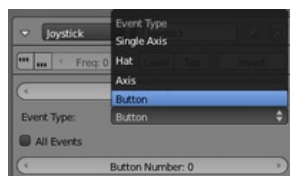
専用オプション:

Index(番号)

使用するジョイスティックを指定します。

All Events(全イベント)

指定したタイプのイベントはすべてトリガに使われます。



Joystick Events

Event Type(イベント種類)

イベントの種類を指定します。



Joystick Single Axis

Single Axis(単軸)

一つの軸を使います。

Axis Number(軸数)

- 1 = 横軸 (左右)
- 2 = 縦軸 (前後)
- 3 = パドル(上下)
- 4 = ジョイスティック(ねじり左右)

Axis Threshold(軸のしきい値)

精密さを指定(範囲は 0 - 32768)



Joystick Hat

Hat(ハット)

指定したハット番号を使います

Hat Direction: 使う方向を指定 (up, down, left, right, up/right, up/left, down/right, down/left)



Joystick Axis

Axis(軸)

Axis Number: 軸を指定

Axis Threshold: 精密さを指定 (範囲は 0 - 32768)

Axis Direction: 使う方向を指定 (Left, Right, Up, Down)



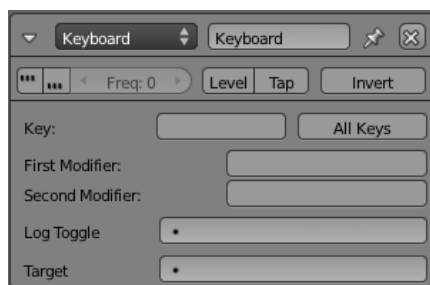
Joystick Button

Button(ボタン)

使うボタンの番号を指定します。

Keyboard sensor

(キーボードセンサー)



Keyboard sensor

キーボード入力を検出します。入力を[String プロパティ](#)に保存することもできます。

共通オプションについては [センサー共通のオプション](#) をご覧ください。

専用オプション:

Key

ひとつのキーの押下を検出します。何も書かれていない欄をクリックして、検出したいキーを押します。これがアクティブキーになり、正パルスを発信します。この欄をクリックして続けて別の場所をクリックすると、キーの登録は消去されます。キーが離された時、偽パルスを発信します。

All keys ボタン

押すと他のボタンは隠され、すべてのキー押下に対して正パルスが発信されます。これは[Pythonコントロール](#)でカスタムキーマップを使う場合に便利です。

Modifier ボタン

Keyと似ていますが、Keyで指定したボタンが押されたときにいっしょに押下されていた場合だけ検出されます。キーは2つ指定できます。ここにキーが登録されると、両方またはすべてのキーが押された場合だけ正パルスが発信されます。これはCtrlRやShiftAltEscなどの操作で特定のアクションをさせるときに便利です。

LogToggle

キーストロークがStringプロパティに保存されるかどうかをBool値で指定します。

Target

キーストロークを保存するStringプロパティを指定します。Propertyセンサーとともに使うと、たとえばパスワードの入力操作などに利用できます。

Message sensor

(メッセージセンサー)



Message sensor

テキストまたはプロパティ値を送信するのに使います。ゲームエンジンのどこかでメッセージが送られると、Messageセンサーは正パルスを発信します。Subjectで指定した件名の場合だけパルスを発信するようにもできます。

共通オプションについては See [センサー共通のオプション](#) をご覧ください。

専用オプション:

Subject

ここで指定したメッセージの受信時にセンサーを起動させます (空欄のままでも構いません)。

メッセージの送り方については [Message Actuator](#) をご覧ください。

Mouse sensor

(マウスセンサー)



Mouse sensor

マウス入力を検出します。使えるのは以下のイベントのみです。

- Left button.
- Middle button.
- Right button
- Wheel Up, スクロールアップ
- Wheel Down, スクロールダウン
- Movement, マウスによるあらゆる動き
- Mouse over, オブジェクトにマウスが重なったことを検出
- Mouse over any, 何らかのオブジェクトにマウスが重なったことを検出

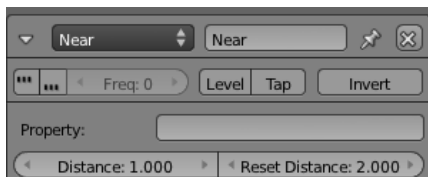
上記の状態が終わると、偽のパルスが送信されます。

具体的なマウスの動きや反応に対するロジックブリックはありません(たとえばプレイヤー視点でカメラを動かしたりなど)。それはPythonで記述する必要があります。

[センサー共通のオプション](#) もご覧ください。

Near sensor

(近接センサー)



Near sensor

指定した距離内に近づいているオブジェクトを検出します。Collisionと同様に、プロパティによってオブジェクトをフィルタリングすることもできます。

センサー共通のオプションについては [こちら](#) をご覧ください。

専用オプション

Property

指定したプロパティを持つオブジェクトのみに検出を制限できます。

Distance

オブジェクト同士の接近を検出する距離です (Blender単位)。

Reset

センサーがリセットされる (負パルスを発信する) 距離です。

註釈

- 1) 近接センサーの検出は他のオブジェクト (壁など) を「突き抜けて」行われます
- 2) オブジェクトの物理設定で「Actor」を有効にしないと検出されません

Property(プロパティ)センサー

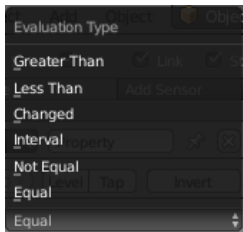


Property sensor

Property(プロパティ) センサーは、自身を所有するオブジェクトのプロパティの変更を探知するセンサーです。

共通するオプションは [共通する Sensor オプション](#) を参照してください。

特別なオプション



プロパティの評価

Evaluation Type(評価タイプ)

Property(プロパティ)がValue(値)に対してどう評価されるかを指定します。

Greater Than(大きい)

プロパティの値がセンサー内のValue(値)より大きい場合(値は含まない)、TRUE パルスを送ります。

Less Than(小さい)

プロパティの値がセンサー内のValue(値)未満の場合、TRUE パルスを送ります。

Changed(変更された)

プロパティの値が変更されると、すぐさま TRUE パルスを送ります。

Interval(区間)

プロパティの値がセンサーのMin(最小) and Max(最大)の間※ の場合、TRUE パルスを送ります(※ 最小と最大は含まない)。

Not Equal(違う)

プロパティの値がセンサー内のValue(値)と違う場合、TRUE パルスを送ります。

Equal(同じ)

プロパティの値がセンサー内のValue(値)と適合する場合、TRUE パルスを送ります。

他のプロパティ名を入力し、プロパティ同士を比較することもできます。

Radar sensor

(レーダーセンサー)



Radar sensor

Nearセンサーに似ていますが、軸から特定の角度の範囲(オブジェクトの中心を指す見えない円錐形)への進入を検出し、また接近は軸上での距離で測られます。

センサー共通のオプションについては [こちら](#) をご覧ください。

専用オプション

Property: 指定されたプロパティを持つオブジェクトのみに検出を限定します。

Axis: レーダーの方向を決めます。+ および - の記号は軸上での正負の向きです。

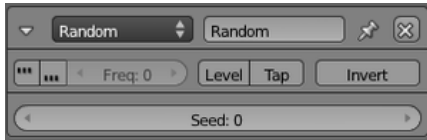
Angle: 円錐の広がりを指定します。

Distance : 円錐の長さを指定します。

このセンサーはたとえば自動的に行動するキャラクターが正面方向しか見えないようにするのに便利です。ただし、他のオブジェクトの向こう側にあるオブジェクトも検出します。

Random sensor

(ランダムセンサー)



Random sensor

Seed 値をもとにして、ランダムなパルスを発信します。Seed が 0 だとランダムでないパルスになり、これはテストなどに使います。値の範囲は 0 - 1000 です。

Seed 値を変えずに再実行すると、パルスの間隔は同じ並びになります。

センサー共通のオプションについては [こちら](#) をご覧ください。

Ray sensor

(レイセンサー)



Ray sensor

指定した軸方向にレイを発射して、何かに当たると正パルスを発信します。指定したプロパティまたはマテリアルを持つオブジェクトのみ検出することもできます。

センサー共通のオプションについては [こちら](#) をご覧ください。

専用オプション

ボタンなどはRadarと多くが共通しています。

Property

指定されたプロパティを持つオブジェクトのみに検出を限定します。

註釈

- 1) Property 欄が設定されていない限り、このセンサーは他のオブジェクト(壁など)を通過してオブジェクトを検知します
- 2) 「Actor」が有効でないオブジェクトは検知されません

Axis

光線の方向を決めます。+ および - の記号は軸上での正負の向きです。

Range

光線の長さを決めます (Blender 単位です)

X-Ray Mode ボタン

X線モードを有効にします。これは指定されたプロパティまたはマテリアルを持たないオブジェクトを通り抜けるようになります。

Touch sensor

(タッチセンサー)



Touch sensor

他のオブジェクトとぶつかったときに正パルスを発信します。Material 欄でマテリアルによるフィルタリングができます。この欄に指定したオブジェクトとぶつかった場合だけパルスが発信されるようになります。すべてのオブジェクトを検出する場合は空にします。衝突が起こったときに正パルスが発信され、離れたときに負パルスが発信されます。“True Pulse triggering”を有効にすると、オブジェクトに触れている間はパルスが発信され続けます。

センサー共通のオプションについては [こちら](#) をご覧ください。

コントローラ

コントローラはセンサからのデータを集めるブリックです。センサが起動すると正パルスが発信され、停止すると負パルスが発信されます。コントローラの役割は、それらの情報をつなぎ合わせて適切な反応を生み出すことです。

コントローラ用のロジックブロックは [Logic Editor](#) を使って構築し、編集できます。この操作の詳細は [コントローラの編集](#) ページをご覧ください。

コントローラの種類

入力を処理する方法は8通りあります。

- [AND](#)
- [OR](#)
- [XOR](#)
- [NAND](#)
- [NOR](#)
- [XNOR](#)
- [Expression](#)
- [Python](#)

以下の表はコントローラの種類と反応をまとめたものです。1列目は出力です。接続されているセンサから送られてくる正パルスの数を表しています。それに続く列は、それぞれの種類のコントローラの反応です。Trueは、コントローラにおいて条件が満たされ、それに接続されたアクチュエータが起動するということです。Falseは条件が満たされず、何も起こらないということです。それぞれのコントローラについてはあとのセクションでさらに解説します。

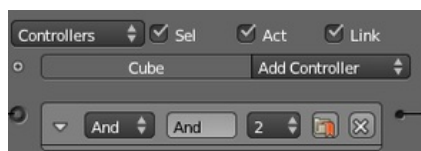
註釈

コントローラには2つ以上のセンサが接続されているとみなされます。1つしか接続されていない場合は表中の"全て"の段のふるまいになります。

正のセンサー	コントローラ					
	AND	OR	XOR	NAND	NOR	XNOR
なし	False	False	False	True	True	True
ひとつ	False	True	True	True	False	False
複数あるが全てではない	False	True	False	True	False	True
全て	True	True	False	False	False	True

Controller Editing

(コントローラーの編集)



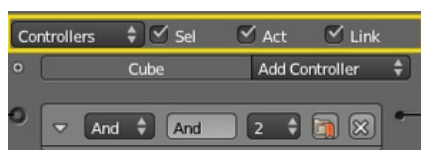
コントローラー列

コントローラーは Logic Panel の中央の列で作成/編集できます。このページではコントローラー全般で使われるコントロールやパラメータと、オブジェクトの状態 (State) の作り方、変更の仕方について説明します。

図はコントローラー列にひとつだけコントローラーがある様子を示しています。列の最上部にある列見出しに、現在の Game Logic にあるコントローラーのうちどれを表示するのか決めるメニューやボタンがあります。

Column Heading

(列見出し)



コントローラー列の見出し

列見出しには表示するコントローラーの選択や表示の詳細度を決めるボタンなどがあります。必要のないコントローラーを隠し、必要なものを目立たせるのに便利です。

Controllers

Show Objects	すべてのオブジェクトを展開する。
Hide Objects	すべてのオブジェクトを名前の表示されるバーのみに縮小する。
Show Sensors	すべてのコントローラーを展開する。
Hide Sensors	すべてのコントローラーを名前の表示されるバーのみに縮小する。

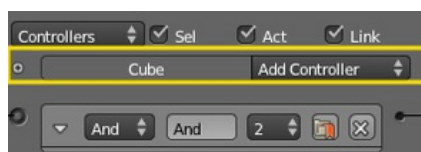
コントローラーとオブジェクトの表示は別々にコントロールできます。

また表示するコントローラーの種類を指定できます。

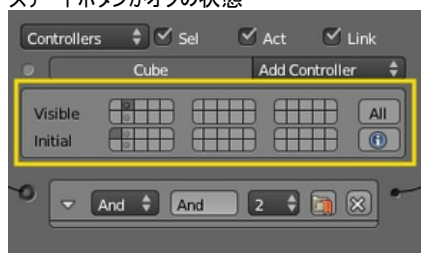
Sel	選択されているすべてのオブジェクトのコントローラー。
Act	アクティブなオブジェクトのコントローラー。
Link	コントローラーにリンクされているコントローラー。

Object Heading

(オブジェクト見出し)



コントローラー列のオブジェクト見出し、使用中ステータスボタンがオフの状態



コントローラー列のオブジェクト見出し、使用中ステータスボタンがオンの状態

列内には、コントローラーがオブジェクトごとに並びます。デフォルトでは選択中のオブジェクトのすべてのコントローラーが表示されますが、列見出しの抽出機能を使って変更できます。

オブジェクトのコントローラー一覧の見出しには三つの項目があります：

(使用中ステートボタン)

オブジェクトで使われているステートを表示します。パネルの詳しい使い方は [States](#) をご覧ください。

Name

オブジェクト名

Add Controller

クリックすると利用できるコントローラの種類を選ぶメニューが現れます。コントローラを選ぶと新たなコントローラがオブジェクトに追加されます。利用できるコントローラについては [Controllers](#) をご覧ください。

AND Controller

ANDコントローラは新規のコントローラを作成したときのデフォルトのタイプです。その理由は、これがもっともよく使われるということと、センサからのパルスを実際にアクチュエータに渡したい場合に上手く動作するからです(コントローラを介さずに渡すことはできません)。

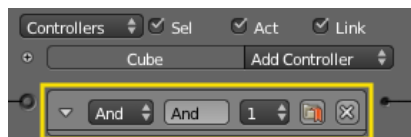
ANDコントローラは接続されているすべてのセンサの状態が正の場合のみアクチュエータを起動します。センサが一つだけ接続されている場合は、センサから発信がそのままアクチュエータの起動につながります。複数のセンサが接続されている場合は、すべてのセンサが同時に起動していなければいけません。これが名前の由来です。センサ1とセンサ2がコントローラに接続されていた場合、センサ1および(AND)センサ2が同時に起動している必要があります。

例

たとえばメニューボタンを作りたいとします。プレイヤーがそのボタンをクリックすると、次のシーンに移動します。

メニューボタンに Mouse Over と {{Literal|Left Button}} の二つの [Mouse センサ](#) を作成します。二つのセンサを同じANDコントローラにつなぎ、そのコントローラは Set Scene アクチュエータにつなぎます。このようにすると、マウスがメニュー上にあり、なおかつクリックされた場合だけ、次のシーンに進むということになります。

オプション



AND Controller

Controller Type メニュー

コントローラの種類を指定します

Controller Name

コントローラの名前で、ユーザーが指定できます。オブジェクト内で重複しない名前にする必要があります。Pythonでコントローラにアクセスする際に使われます。

State Index

このコントローラが実行される状態を設定します。

Preference ボタン

オンにすると、オフにしている他のすべてのコントローラより先に実行されます(開始時スクリプトに便利です)

ボタン

センサーを削除します

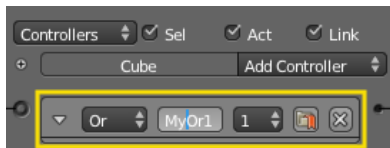
OR Controller

OR コントローラは、接続されたセンサのうちすくなくとも一つが起動していると、アクチュエータを起動します。センサが一つだけ接続されている場合は、センサから発信がそのままアクチュエータの起動につながります。複数のセンサが接続されている場合は、どれか一つのセンサは起動していなければいけません。OR は排他的ではありません。つまり、AND コントローラが起動するような状況(すべてのセンサが起動している)でも同様に起動します。排他的論理和には XOR コントローラを使ってください。

例

EscかQを押せばゲームが終了するようにしたい場合、それぞれのキーに対して [Keyboard センサ](#)を設定します。それを同じORコントローラにつなぎ、そのコントローラを [Quit this game アクチュエータ](#)につなぎます。これで、どちらかのキーが押されればゲームは終了します。

オプション



OR Controller

Controller Type メニュー

コントローラの種類を指定します

Controller Name

コントローラの名前で、ユーザーが指定できます。オブジェクト内で重複しない名前にする必要があります。Pythonでコントローラにアクセスする際に使われます。

State Index

このコントローラが実行されるステートを設定します。

Preference ボタン

オンにすると、オフにしている他のすべてのコントローラより先に実行されます(開始時スクリプトに便利です)

X ボタン

センサーを削除します

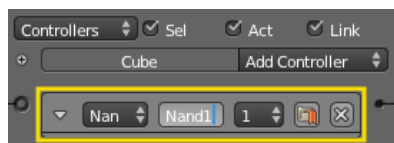
NAND Controller

NAND は Not And の略です。論理学では、Not は結果を反転します。つまり Not And は AND コントローラの正反対の働きをします。AND コントローラはすべてのセンサが発信しているときだけアクチュエータを起動します。それと反対に、NAND コントローラはすべてのセンサが起動してはいない場合のみアクチュエータを起動します。センサが一つだけ接続されている場合は、センサが発信していないときだけアクチュエータを起動します（単純な NOT と同じように働きます）。

例

---製作中---

オプション



NAND Controller

Controller Type メニュー

コントローラの種類を指定します

Controller Name

コントローラの名前で、ユーザーが指定できます。オブジェクト内で重複しない名前にする必要があります。Pythonでコントローラにアクセスする際に使われます。

State Index

このコントローラが実行されるステートを設定します。

Preference ボタン

オンにすると、オフにしている他のすべてのコントローラより先に実行されます（開始時スクリプトに便利です）

X ボタン

センサーを削除します

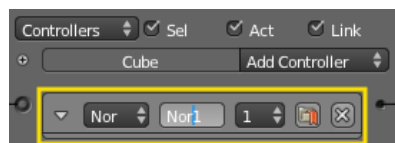
NOR Controller

NOR コントローラは OR コントローラの反対です。センサが一つも発信していない場合だけアクチュエータを起動します。

例

---製作中---

オプション



NOR Controller

Controller Type メニュー

コントローラの種類を指定します

Controller Name

コントローラの名前で、ユーザーが指定できます。オブジェクト内で重複しない名前にする必要があります。Pythonでコントローラにアクセスする際に使われます。

State Index

このコントローラが実行される状態を設定します。

Preference ボタン

オンにすると、オフにしている他のすべてのコントローラより先に実行されます (開始時スクリプトに便利です)

X ボタン

センサーを削除します

XOR Controller

XOR コントローラは、排他的論理和のコントローラです。

このコントローラは次の条件を満たすとき正の (TRUE/真の) 信号を出力します :

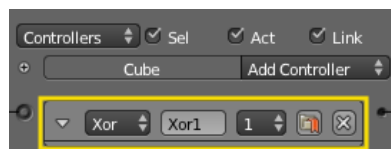
- 入力の一つ(一つだけ)が TRUE、かつ オブジェクトが指定状態 (Designated State) であるとき

これ以外では負の (FALSE/偽の) 信号を出力します。

例

--製作中--

オプション



XOR Controller

Controller Type メニュー

コントローラの種類を指定します

Controller Name

コントローラの名前で、ユーザーが指定できます。オブジェクト内で重複しない名前にする必要があります。Pythonでコントローラにアクセスする際に使われます。

State Index

このコントローラが実行されるステートを設定します。

Preference ボタン

オンにすると、オフにしている他のすべてのコントローラより先に実行されます (開始時スクリプトに便利です)

X ボタン

センサーを削除します

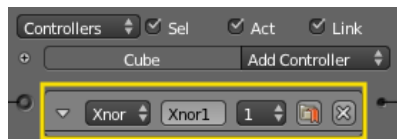
XNOR Controller

XNOR コントローラは、混乱するような名前ですが、単純に XOR コントローラの反対です。一つより少ないまたは多数のセンサが発信しているときだけアクチュエータを起動します。

例

---製作中---

オプション



XNOR Controller

Controller Type メニュー

コントローラの種類を指定します

Controller Name

コントローラの名前で、ユーザーが指定できます。オブジェクト内で重複しない名前にする必要があります。Pythonでコントローラにアクセスする際に使われます。

State Index

このコントローラが実行される状態を設定します。

Preference ボタン

オンにすると、オフにしている他のすべてのコントローラより先に実行されます(開始時スクリプトに便利です)

ボタン

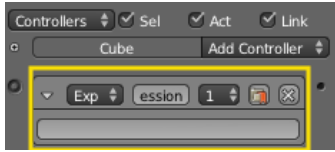
センサーを削除します

Expressions Controller

(式コントローラ)

コントローラは式を評価して正または負のパルスをアクチュエータに送ることができます。

- 式の評価が**True**(真)の場合、正のパルスを送ります。
- 式の評価が**False**(偽)の場合、負のパルスを送ります。



Expression Controller

Expression (式)

枠内に記述する式には変数 (variable)、定数 (constant)、および 演算 (operator) を組み合わせることができます。以下のルールに従う必要があります。

変数

使えるのは:

- **sensors names:** センサの名前
- **properties:** プロパティをオブジェクトに登録して、式の中で使えます

これらの名前にはスペースを含んではいけません。

演算

算術演算

演算子: *, /, +, -

返り値: 数値

例: 3 + 2, 35 / 5

論理演算

- 比較演算子: <, >, >=, <=, ==, !=
- ブーリアン演算子: AND, OR, NOT

返り値: 真または偽

例: 3 > 2 (True), 1 AND 0 (False)

条件文 (if)

以下の文を使います。

```
if( expression, pulse_if_expression_is_true, pulse_if_expression_is_false )
```

式 (expression) の評価が真の場合、

- **pulse_if_expression_is_true** が真ならアクチュエータに正パルスを送る。
- **pulse_if_expression_is_true** が偽ならアクチュエータに負パルスを送る。

式の評価が偽の場合、

- **pulse_if_expression_is_false** が真ならアクチュエータに正パルスを送る。
- **pulse_if_expression_is_false** が偽ならアクチュエータに負パルスを送る。

例

たとえば、オブジェクトに `coins` というプロパティをつくり、値は 30 にしておきます。

```
coins > 20
```

これは真を返します (アクチュエータに正パルスを送ります)。

たとえば、オブジェクトが

- **Key_Inserted** というセンサを持ち、値は真
- **Fuel** というプロパティを持ち、値は偽

という場合、

Key_Inserted AND Fuel

これは偽を返します (アクチュエータに負パルスを送ります)。

これは以下と同じです。

```
if (Key_Inserted AND Fuel, True, False)
```

また、

```
if (Key_Inserted AND Fuel, False, True)
```

こう書くと、**Key_Inserted AND Fuel**が偽の場合に正パルスが送られます。

さらに、

```
if ((Key_Inserted AND Fuel) OR (coins > 20), True, False)
```

これは真を返します。この場合はアクチュエータに正パルスが送られます。

Python Controller

Pythonコントローラはユーザがプログラムしたスクリプトに従って入力を処理するコントローラです。Pythonスクリプト、またはPythonコードの含まれるファイルを使います。Pythonコントローラには二つのモードがあります。ScriptとModuleです。どちらもテキストエディタで書いて.blendファイルの内部に保存することもできるし、外部ファイルにすることもできます。

BGEにおけるPythonについては[ここ](#)を参照。

BGEのPython APIについては[ここ](#)を参照。

Script Mode

Scriptモードでは、コントローラはスクリプトにリンクされ、ロジックが進む前にスクリプト全体が実行されます。スクリプトに記述されていることはすべて同じフレームにおいて起こります。そのため、処理の中で同じ値または属性が何度も変わった場合、最終的な結果だけがゲームとして表示されます。たとえばオブジェクトがまず(100.0,100.0,100.0)に移動し、それから(0.0,0.0,0.0)に移動した場合、表示されるのは(0.0,0.0,0.0)への移動だけです。

Module Mode

Blender 2.49からはPythonモジュールコントローラが加わりました。PythonコントローラのドロップダウンメニューでScriptのかわりにModuleを選んでください。そして、そのモジュールのなかで関数を定義し、その関数をコントローラで呼び出します。エディットボックスには“myScript.py”のかわりに“myModule.myFunc”という風に指定します。その関数はコントローラが呼び出されるたびに処理されます。しかし、myFuncのスコープの外にある関数や変数は一度だけしか処理されません。これは最初に一度だけ変数を初期化したい場合に便利です。

Pythonモジュールコントローラが扱える属性の数に制限はありません。つまりパッケージが自動的にサポートされるということです。“myModule.myFunc”と同様に“myPackage.myModule.myFunc”とネストされたパッケージも動作します。“myPackage.myModule.myInstance.myMethod”のようなメソッドも動作します。Pythonコントローラは、一つの引数をとる関数に対す引数として関数に渡されます。

このためライブにスクリプトを編集できます。Pythonモジュールコントローラについてさらに学ぶには、

*Thread <http://blenderartists.org/forum/showthread.php?t=156672>

*Video http://download.blender.org/apricot/live_bge_edit.ogv

アクチュエータ

アクチュエータは、移動、オブジェクトの生成、サウンドの再生などのアクションを行います。接続しているコントローラから一つ(またはそれ以上)の正パルスを受け取ると、アクチュエータはその関数を初期化します。

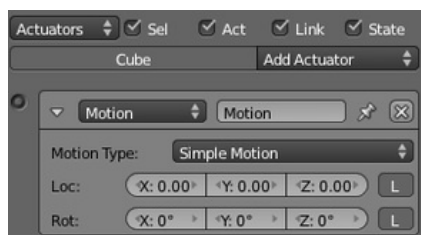
アクチュエータ用のロジックブロックは Logic Editor を使って構築し、変更することができます。この手順の詳細は[アクチュエータの編集](#)にあります。

以下のアクチュエータが使えます:

- [Action](#) アーマチャのアクションを操作します。アーマチャが選択されているときのみ表示されます。
- [Camera](#) 滑らかにオブジェクトを追跡するオプションがあります。主にカメラオブジェクトのためのものですが、どんなオブジェクトでも使えます。
- [Constraint](#) オブジェクトの位置、距離、回転を制限します。ゲームの中の物理をコントロールするのに便利です。
- [Edit Object](#) オブジェクトのメッシュの編集、オブジェクトの追加または削除、またメッシュを変更することもできます(すると衝突メッシュも再生成されます)。
- [Filter 2D](#) 画面をセピア色にしたり青みがかった感じにしたりなどの特殊効果フィルタです。
- [Game](#) ゲーム全体の操作、リスタート、終了、ロード、セーブなどをします。
- [Message](#) メッセージを送り、受け取ったオブジェクトを起動したりできます。
- [Motion](#) オブジェクトを動かしたり回転させたりします。瞬間移動させるのではなく、物理的に押ししたり回転させたりするオプションもあります。
- [Parent](#) オブジェクトを親にしたり、それを解除したりできます。
- [Property](#) オブジェクトのプロパティを操作します。設定、加算、コピーなど。
- [Random](#) ランダムな値を生成します。プロパティに格納することもできます。
- [Scene](#) .blendの中のシーンをあつかいます。そのシーンをゲームのステージやUI、背景などに使います。
- [Sound](#) ゲーム内でサウンドを再生するのに使えます。
- [State](#) オブジェクトのステートを切り替えます。
- [Steering](#) オブジェクトの経路検索のオプションです。
- [Visibility](#) オブジェクトの表示・非表示を変更します。

Actuator Editing

(アクチュエータの編集)



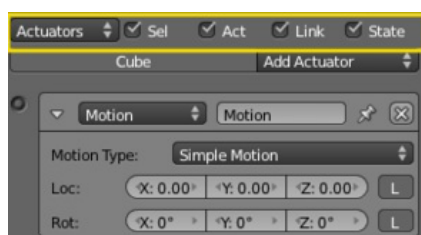
典型的なアクチュエータのあるアクチュエータ列

アクチュエータは Logic パネルの右の列で作り、編集できます。このページでは通常の列コントロールと、個々のアクチュエータで一般的に使われるパラメータについて説明します。

図は、アクチュエータの一つ持つ、典型的なアクチュエータ列を示しています。列の最上部にある列見出しには、現在の Game Logic にあるすべてのアクチュエータを操作するためのメニューやボタンが表示されています。

Column Heading

(列見出し)



アクチュエータ列の見出し

列見出しのアクチュエータ列には、アクチュエータの選択やその表示詳細度を設定するためのコントロールがあります。必要のないアクチュエータを隠して、必要なものを見つけやすくするのに役立ちます。それぞれ別々に操作できます。

Actuators

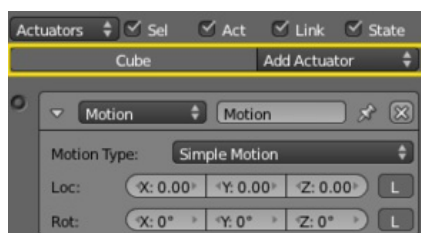
Show Objects	全オブジェクトを展開します。
Hide Objects	名前を表示するバーだけになるように全オブジェクトを折りたたみます。
Show Actuators	全アクチュエータを展開します。
Hide Actuators	名前を表示するバーだけになるように全アクチュエータを展開します。

ヘッダーにある4つのボタンを使って、表示されるアクチュエータを抽出できます:

Sel	選択中のオブジェクトの全アクチュエータを表示
Act	アクティブオブジェクトに属するアクチュエータのみ表示
Link	コントローラーに繋がっているアクチュエータのみ表示
State	アクティブなステートを持ったコントローラーに繋がっている、アクチュエータのみを表示

Object Heading

(オブジェクト見出し)



オブジェクト見出し

アクチュエータ列には、アクチュエータがオブジェクトごとにグループ分けされて並んでいます。デフォルトでは、選択中のオブジェクトのアクチュエータがすべて並びますが、列の見出し部分にある抽出機能を使って変更可能です。

オブジェクトのアクチュエータ一覧の見出し部分には、二つのボタンがあります:

Name

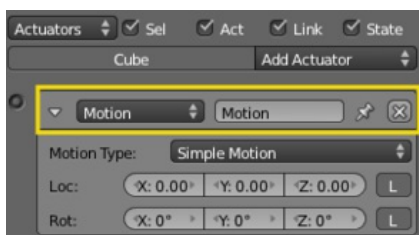
オブジェクト名

Add

クリックすると利用可能なアクチュエータの選択メニューが現れます。選択して、オブジェクトに新たにアクチュエータを追加します。利用可能なアクチュエータの一覧は、[Actuators](#) をご覧ください。

Actuator Common Options

(アクチュエータの共通オプション)



アクチュエータの共通オプション

すべてのアクチュエータに共通する、次のようなボタン、入力欄、メニューがあります:

三角形のボタン

アクチュエータ情報を 1 行に折りたたみます。再度押すと展開します

アクチュエータ種類 メニュー

アクチュエータの種類を選びます

アクチュエータ名

アクチュエータの名前です。ユーザーが定義できます。Python でアクチュエータにアクセスする際に使われます。選択されたオブジェクト間で重ならない名前にする必要があります。

X ボタン

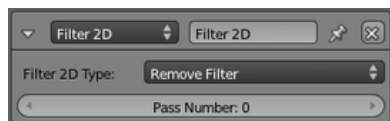
アクチュエータを削除します。

2D Filter アクチュエータ

2D Filterはイメージフィルタのアクチュエータです。最終的にレンダリングイメージに対して適用されます (FPSゲームのサンプル画像を提供してくれた人々に感謝)。

2D Filter の種類

次のフィルターが用意されています



2D Filters

- Custom Filter
- Invert
- Sepia
- Gray Scale
- Prewitt
- Sobel
- Laplacian
- Erosion
- Dilation
- Sharpen
- Blur
- Motion Blur
- Remove Filter
- Disable Filter
- Enable Filter

どのフィルターもパラメータはひとつだけです

Pass Number

フィルターを使うパス番号

フィルターの詳細については以下の説明をご覧ください。

Motion Blur

(モーションブラー)

Motion Blur を使って動きの表現をするには、以前のフレームの情報が必要です。図"2D Filters: Gmae Logic"はBlenderのウィンドウに表示されたロジックブリックです。



2D Filters: Motion Blur.



2D Filters: Game Logic.

フィルタを有効にするには:

1. 適切なセンサとコントローラを追加する。
2. 2D Filterアクチュエータを追加する。
3. ドロップダウンリストからMotion Blurを選択する。
4. Value (モーションブラーの度合い)

フィルタを無効にするには:

1. 適切なセンサとコントローラを追加する。
2. 2D Filterアクチュエータを追加する。
3. ドロップダウンリストからMotion Blurを選択する。
4. Enableボタンを押して無効モードにする。

Pythonコントローラを使ってモーションブラーを有効にすることもできます:

```
from bge import render
render.enableMotionBlur(0.85)
```

無効にするには:

```
from bge import render
render.disableMotionBlur()
```

註釈

グラフィックハードウェアとOpenGLがアキュムレーションバッファをサポートしている必要があります (glAccum関数)。

Built-In 2D Filters

(組み込みの2次元フィルター)

2D Filterアクチュエータで見ることができるフィルタはすべて同じアーキテクチャを用いています。すべての内蔵フィルタは最終的なイメージを生成するためにフラグメントシェーダを使います。よって、ハードウェアがシェーダをサポートしている必要があります。



2D Filters: Motion Blur.



2D Filters: Sepia.



2D Filters: Sobel.

Blur, Sharpen, Dilation, Erosion, Laplacian, Sobel, Prewitt, Gray Scale, Sepia, Invertが内蔵フィルタとして使えます。これらのフィルタはいくつかのパスで使用できます。

フィルタを使うには:

1. 適切なセンサとアクチュエータを作成する。
2. 2D Filterアクチュエータを作成する。
3. 使用するフィルタ、たとえばBlurを選択する。
4. フィルタを適用するパスの番号を指定する。

特定のパスでフィルタを削除するには:

1. 適切なセンサとアクチュエータを作成する。
2. 2D Filterアクチュエータを作成する。
3. Remove Filterを選択する。
4. フィルタを削除するパスの番号を指定する。

特定のパスでフィルタを無効にするには:

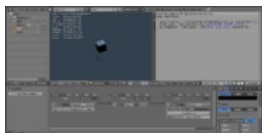
1. 適切なセンサとアクチュエータを作成する。
2. 2D Filterアクチュエータを作成する。
3. Disable Filterを選択する。
4. フィルタを無効にするパスの番号を指定する。

特定のパスでフィルタを有効にするには:

1. 適切なセンサとアクチュエータを作成する。
2. 2D Filterアクチュエータを作成する。
3. Enable Filterを選択する。
4. フィルタを無効にするパスの番号を指定する。

Custom Filters

(カスタムフィルター)



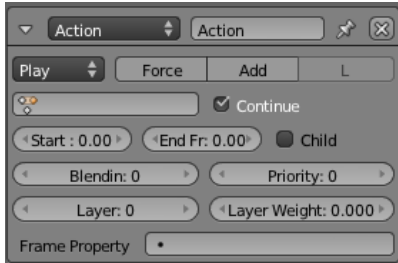
2D Filters: Custom Filter.

カスタムフィルタによって、GLSLを使って好きな2Dフィルタを定義することができます。使い方は内蔵フィルタと同じですが、2D FilterアクチュエータでCustom Filterを選択する必要があります。そしてテキストエディタでシェーダプログラムを書き、アクチュエータにスクリプト名を設定します。

青セピアの例:

```
uniform sampler2D bgl_RenderedTexture;
void main(void)
{
    vec4 texcolor = texture2D(bgl_RenderedTexture, gl_TexCoord[0].st);
    float gray = dot(texcolor.rgb, vec3(0.299, 0.587, 0.114));
    gl_FragColor = vec4(gray * vec3(0.8, 1.0, 1.2), texcolor.a);
}
```


Action アクチュエータ



Action Actuator

Actionアクチュエータはアーマチュアが選択されているときのみ表示されます。アクションはアーマチュアに格納されているからです。

アクチュエータ共通のオプションについては [こちら](#) をご覧ください。

専用オプション:

Action Playback Type

Play -

TRUE パルスを受けた時 IPO の開始フレームから終了フレームまで一度だけ再生します

Ping Pong

Flipper

TRUE パルスを受けた時 IPO の開始フレームから終了フレームまで一度だけ再生します (FALSE パルスを受けた時は逆方向に再生します)。

Loop End

TRUE パルスを受けた時 IPO の終了フレームから開始フレームまでを繰り返し再生します

Loop Start

TRUE パルスを受けた時 IPO の開始フレームから終了フレームまで繰り返し再生します

Property

Action

使用するアクション

Continue

オン・オフを切り替える際、前回のフレームを記憶します。無効にすると毎回最初から再生されます。

Start Frame

再生を始めるアクションのフレーム

End Frame

再生を終えるアクションのフレーム

Child Button**Blendin**

モーションブレンディングに使うフレーム数

Priority

実行の優先順位。低い数値にすると、高い数値のアクションをオーバーライドします。2つまたはそれ以上のアクションが同時に起こるときは、優先順位の高いチャンネルはスタックの中で低い位置になければいけません。

Frame Property

アクションのカレントフレーム番号をこのプロパティに登録します。

Property

ここに指定したプロパティに基づいてアクションの位置を決めます。再生モードがPropertyの場合のみです。

Layer**Layer Weight**

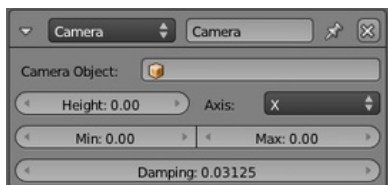
Camera Actuator

カメラにオブジェクトを追従させます。

オプション

アクチュエータ共通のオプションについては [こちら](#)をご覧ください。

専用オプション:



Camera Actuator

Camera Object

カメラが追従する Game Object の名前

Height

Game Objectの中心からの高さ。カメラがこの位置に留まり続けようとする

Axis

カメラの追従する軸 (X か Y)

Min

Game Object を追従するカメラの最小距離

Max

Game Object を追従するカメラの最大距離

Damping

カメラを対象の後ろに動かす強制力。0 から 10 の範囲で指定します。値が高いほどカメラが制約範囲内 (Min/Max/Heightによるもの) に位置調整するのが速くなります。

関連項目

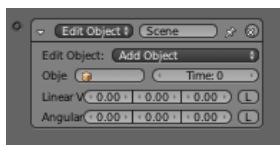
- [Camera](#)
- [Fisheye Dome camera](#)

Edit Object(オブジェクト編集)アクチュエーター

Edit Object(オブジェクト編集)アクチュエーターで、ゲーム中のオブジェクトの設定を編集できます。

共通のオプションは[アクチュエーター共通のオプション](#)を参照してください。

特別なオプション:

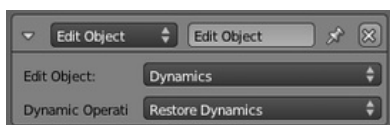


Edit Object actuator

Edit Object(オブジェクト編集)

Edit Object(オブジェクト編集)アクチュエーターのオプションのメニュー

- Dynamics(力学処理)
- Track To(追跡)
- Replace Mesh(メッシュ切替)
- End Object(エンドオブジェクト)
- Add Object(オブジェクト追加)



Edit Object(オブジェクト編集)アクチュエーター - Dynamics(力学処理)

Dynamics(力学処理)

オブジェクトの力学処理のオプションを設定するDynamic Operation(力学処理の操作)メニューを提供します。

Set Mass(質量を設定)

現在のオブジェクトの物理演算用の質量を設定できます(範囲:0 - 10,000)。

Disable Rigid Body(剛体力学を無効化)

剛体状態を無効にします。衝突が無効になります。

Enable Rigid Body(剛体力学を有効)

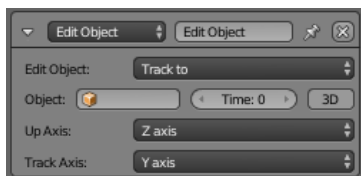
剛体状態を有効にします。衝突が有効になります。

Suspend Dynamics(力学処理を一時停止)

オブジェクトの力学(速度)を中断します。

Restore Dynamics(力学処理を再開)

オブジェクトの力学(速度)を再開します。



Edit Object(オブジェクト編集)アクチュエーター - Track to(追跡)

Track to(追跡)

そのオブジェクトが他のオブジェクトを2Dまたは3Dで「注視」するようにします。

Object(オブジェクト)

追跡するオブジェクト。

Time(時間)

ターゲットオブジェクトの方を向くのにかかるフレーム数(範囲:0-2000)。

3D(トグル)

2D (X,Y) または 3D (X,Y,Z) での追跡を切り替えます。

Up Axis(上方向の軸)メニュー

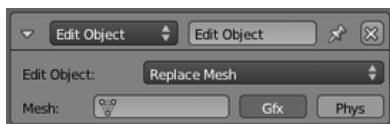
上を指し示す (X、Y、Z)軸(デフォルトはZ軸)。

Track Axis(トラック軸)メニュー

ターゲットオブジェクトを指し示す (X、Y、Z、-X、-Y、-Z)軸(デフォルトはY軸)。



Up Axis(上方向の軸)メニューとTrack Axis(トラック軸)メニューで同じ軸を指定した場合、機能しくなくなります。



Edit Object(オブジェクト編集)アクチュエーター - Replace Mesh(メッシュ変更)

Replace Mesh(メッシュ変更)

メッシュを他の物で置き換えます。表示用と物理演算用の両方を一緒、または別々に変更できます。

Mesh

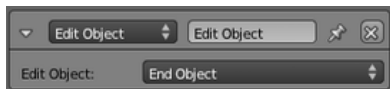
現在のメッシュを置き換えるメッシュ名。

Gfx Button

表示するメッシュを置き換えます。

Phys Button

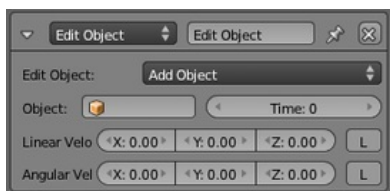
物理演算に使うメッシュを置き換えます(「組み合わせ」は除く)



Edit Object(オブジェクト編集)アクチュエーター - End Object(エンドオブジェクト)

End Object(エンドオブジェクト)

現在のオブジェクトを消失させます。



Edit Object(オブジェクト編集)アクチュエーター - Add Object(オブジェクト追加)

Add Object(オブジェクト追加)

オブジェクトを現在のオブジェクトの中心に追加します。追加されるオブジェクトは、別の非表示レイヤに置いておく必要があります。

Object(オブジェクト)

追加するオブジェクト名。

Time(時間)

そのオブジェクトが消えるまで生存し続ける時間(フレーム数)。0で無限。

Linear Velocity(線形速度)

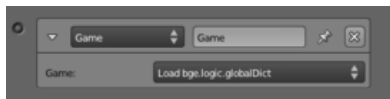
作成されたオブジェクトの線形速度。オブジェクト射出時の、初期速度の設定に便利です。

Angular Velocity(角速度)

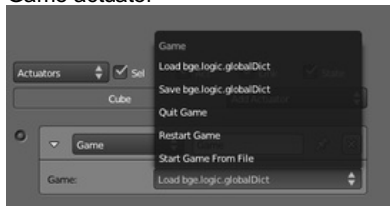
作成されたオブジェクトの角速度。

Game アクチュエータ

Gameアクチュエータはゲームに関連した機能、たとえばリスタートや終了、ロードなどを行います。



Game actuator



Game

種類

Load bge.logic.globalDict

.bgeconfからbge.logic.globalDictをロードします。

Save bge.logic.globalDict

.bgeconfにbge.logic.globalDictをセーブします。

Quit Game

このアクチュエータが起動するとBlender playerが終了するようにします。

Restart Game

このアクチュエータが起動するとBlender playerがリスタートするようにします(ファイルをリロードします)。

Start Game From File

このアクチュエータが起動するとBlender playerは指定されたパスの.blend ファイルをスタートします。

File

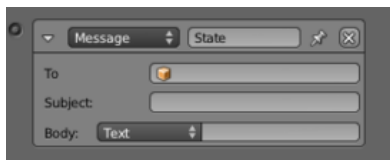
ロードする.blendファイルのパス。

使用する際の注意

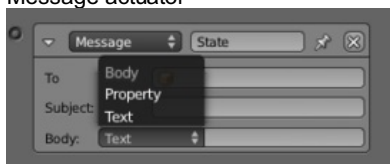
keyboardセンサでEscを検出するようにしている場合、Pythonファイルでのエラー等が起きてQuit Gameアクチュエータが失敗すると、ゲームが終了できなくなります。データはquit.blend (File » Recover Last Session)から復旧できるかもしれません。

Message アクチュエータ

Messageアクチュエータは、シーン内で、またはシーン同士の間でデータを送ることができます。



Message actuator



Message actuator Options

To
メッセージを送る相手のオブジェクト。すべてのオブジェクト(または他のシーン)にメッセージを送る場合はこの欄は空にします。

Subject
メッセージの件名。たとえば"end-game"というメッセージを送る場合、messageセンサーで"end game"を検出し、ANDコントローラを介してQuit Gameアクチュエータを起動する、という使い方ができます。

Body
メッセージの本文(これを読めるのはPythonのみです)。

Text

ユーザの指定したテキスト。

Property

指定したプロパティ。たとえばハイスコアなど。

使用する際の注意

Messageアクチュエータは、たとえばオブジェクトにスコアを送ったりするのに使えますが、シーンをまたいでデータを送るのにも使えます(もう一つの方法としては**bge.logic.globalDict**を使います)。

Motion Actuator

このアクチュエータはオブジェクトを移動/回転させます。移動、回転、または動的な移動を行う、Simple および Servo の二つの操作方法が用意されています。Simple モードの処理はオブジェクトの物理設定に左右されます。

アクチュエータ共通のオプションについては [こちら](#) をご覧ください。

専用オプション:

Motion Type

動きの種類を決めます

Simple Motion

さまざまな動きを直接適用します

Servo Control

目標速度と、その速度に到達するまでの早さを指定します

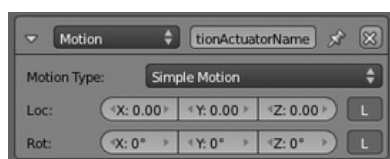


オブジェクトの衝突

Simple 動作モードではオブジェクトは始点と終点以外の座標を一切通らないため、他のオブジェクトを通り抜ける可能性があります。オブジェクトの物理設定がDynamic/Rigid Body/Soft Body に設定されると実行される Servo モードでは、この問題を回避できます。

Simple Motion

(単純運動)



Motion actuator for Simple Motion

Loc

オブジェクトはパルスを受信するたびに、各軸方向に指定値だけジャンプします。

Rot

オブジェクトはパルスを受信するたびに、各軸を中心にして指定量だけ回転します。1回転は値 7.2で表されます (0.02 で 1度)。



指定した座標がグローバル(灰色)とローカル(白色)のどちらなのか指定します。



Servo Control

Servo Control を動作させるには、Physics ウィンドウで Dynamic をオンにし、オブジェクトを Actor にする必要があります。

Servo Control

(サーボ制御)



Motion actuator for Servo Control

Servo control は物質界のものの動きを模倣する強力な手段です。Servo controller は、オブジェクトに与える力を調整し、指定した速度になるようにします。運動の比例 - 積分 - 微分 (PID) の数式を使います。

Reference Object

アクチュエータの所有者が運動の基準として使うオブジェクトを指定します。例えば動く足場などです。空欄にすると World を基準にします。

Linear Velocity

三軸それぞれにおける線速度で、オブジェクトの速度の目標値になります。



指定した座標がグローバル(灰色)なのかローカル(白色)なのかを指定します

X, Y, Z

オブジェクトに適用される力の最大/最小値を設定します。無効にすると(X、Y、Zボタンが灰色になります) 力の上限はなくなります。

Proportional Coefficient

比例係数を調整します。実際の速度と目標線速度との差への反作用を制御します。

Integral Coefficient

積分係数を設定します。この動きのこれまでの誤差の合計への反作用を制御します。

Derivative Coefficient

微分係数を設定します。反作用を制御します。

Parent Actuator

現在のオブジェクトの親を変えることができます。

アクチュエータ共通のオプションについては [こちら](#) をご覧ください。

専用オプション:



Parent Actuator

Scene

必要な操作を選びます

Set Parent

このオブジェクトを現在のオブジェクトの親にします

Parent Object

親オブジェクトの名前

Compound'

このオブジェクトの形状を親の形状に足します (親の形状がすでに compound されているときのみ)

Ghost'

親である間このオブジェクトを Ghost にします

Remove Parent

現在のオブジェクトの全ての親を取り除きます

Parent Object

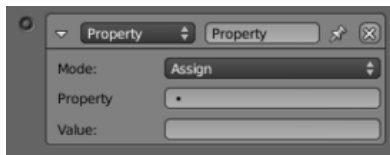
親オブジェクトの名前

Property(プロパティ)アクチュエーター

Property(プロパティ)アクチュエーターは、アクティブになると指定したプロパティの値を変更します。

共通のオプションについては[アクチュエーター共通のオプション](#)を参照してください。

特別なオプション:



Property(プロパティ)アクチュエーター

Mode(モード)

Assign(適用)

Property(プロパティ)に指定したターゲットプロパティを、Value(値)に指定した値にします。

Add(追加)

Value(値)をProperty(プロパティ)の値に加算(負の値を入れると減算)します。Bool(ブーリアン)では、0(または負の値)以外が True として加算されます。

Copy(コピー)

他のオブジェクトのプロパティからこのアクチュエーターのオーナーのプロパティにコピーします。

Toggle(切り替え)

0と1、そして0以外の数字の場合は0に切り替えます。ON/OFFスイッチに便利です。

Level(レベル)

アクチュエーターがアクティブになると1、非アクティブになると0になります。

Property(プロパティ)

このアクチュエーターが変更するターゲットプロパティ。

Value(値)

プロパティを変更するのに使用する値。

例

たとえばゲームのキャラクターが"HP"(ヒットポイント)というプロパティを持っているとします。あとどのくらいのダメージを受けたら死亡するかという数値を決める数値です。HPは整数で、100から始まるとします。

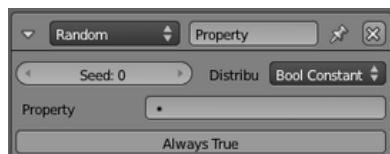
二つのCollision(コリジョン)センサーを追加します。一つは敵の弾、もう一つは回復薬の回収用です。最初のセンサーは、(ANDコントローラを介して)Add Property(追加プロパティ)アクチュエーターにつなぎ、変更するプロパティ名"HP"を指定して、Value(値)は-10に設定します。プレイヤーは敵の弾に当たるたびにHPを10失います。二つ目のセンサーは、(ANDコントローラを介して)Add Property(追加プロパティ)アクチュエーターにつなぎ、変更するプロパティ名"HP"を指定して、Value(値)は50に設定します。プレイヤーが回復薬にぶつかるたびにHPが50増えます。次に、上限を設定するためにProperty(プロパティ)センサーを設定し、interval(区間)モードで100以上のときの反応するようにします。そして(ANDコントローラを介して)Assign Property(適用プロパティ)アクチュエーターにつなぎ、プロパティを100に変えるように設定します。これでHPが100以上になってしまう場合は100に設定されます。

Random Actuator

オブジェクトのプロパティにランダムな値をセットします

アクチュエータ共通のオプションについては [こちら](#) をご覧ください。

専用のオプション:



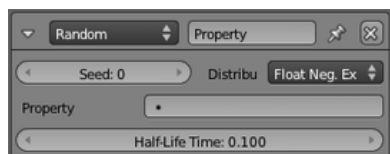
Random Actuator

Seed(シード)

乱数生成器の開始シード値です(1 - 1000 の範囲)

Distribution(分布)

乱数が抽出元となる分布のメニューです。デフォルトの Boolean Constant は True または False 値を返し、テスト目的に役立ちます。



Float Neg. Exp.

Float Neg. Exp.

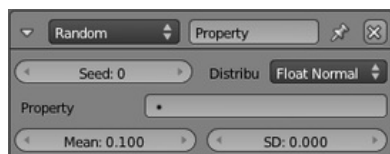
指定した半減期を持ち指数関数的に減衰する値

Property

値を受け取る Float 型のプロパティ

Half-Life Time(半減期)

半減期(範囲: 0.00 -10000.00)



Float Normal

Float normal

正規分布から得られる乱数

Property

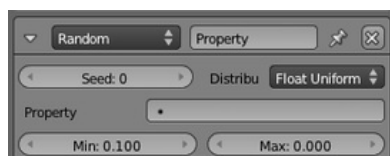
値を受け取る Float 型のプロパティ

Mean(平均)

正規分布の平均値(範囲: -10000.00 から +10000.00)

SD(標準偏差)

正規分布の標準偏差(範囲: 0.00 から +10000.00)



Float Uniform

Float uniform

最大/最小値間で均等に選ばれたランダムな値

Property

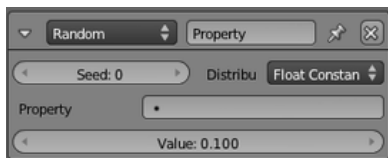
値を受け取る Float 型のプロパティ

Min

最小値(範囲: -10000.00 から +10000.00)

Max

最大値 (範囲: -10000.00 から +10000.00)



Float Constant

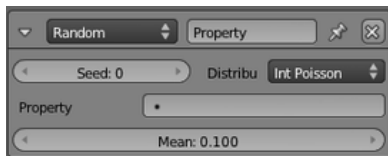
Float constant
定値を返します

Property

値を受け取る Float 型のプロパティ

Value

値 (範囲: 0.00 から +1.00)



Random Integer Poisson

Int Poisson
ポアソン分布から得られる乱数

Property

値を受け取る Integer 型のプロパティ

Mean (平均)

ポアソン分布の平均値 (範囲: 0.01 から +100.00)



Random Integer Uniform

Int uniform
最大/最小値間で均等に選ばれたランダムな値

Property

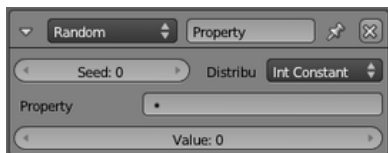
値を受け取る Integer 型のプロパティ

Min

最小値 (範囲: -1000 から +1000)

Max

最大値 (範囲: -1000 から +1000)



Random Integer Constant

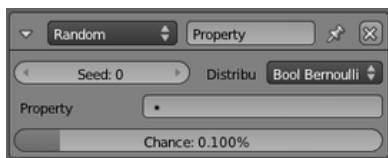
Int constant
定値を返します

Property

値を受け取る Integer 型のプロパティ

Value

値 (範囲: 0.00 から +1.00)



Random Bool Bernoulli

Bool Bernoulli

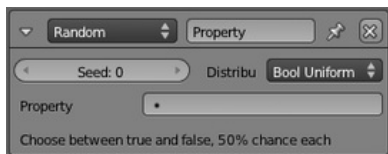
指定した比率で TRUE パルスを含む、ランダムな分布を返します

Property

値を受け取る Boolean 型のプロパティ

Chance (確率)

TRUE を受け取る比率



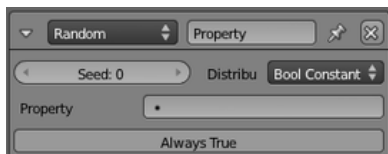
Random Bool Uniform

Bool uniform

50対50の確率で True か False を受け取ります

Property

値を受け取る Boolean 型のプロパティ



Random Bool Constant

Bool constant

定値を返します

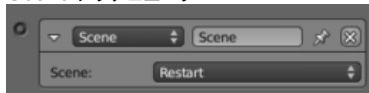
Property

値を受け取る Boolean 型のプロパティ

Value

値 (True または False)

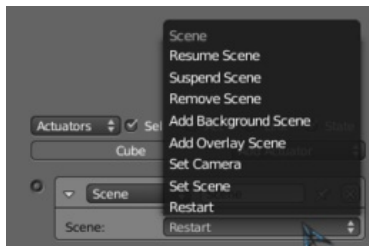
Scene アクチュエータ



Scene actuator

Sceneアクチュエータは.blendファイルの中のシーンを扱います。たとえばゲームのステージやUI、背景などをコントロールします。

このアクチュエータには8つのモードがあります。



Scene actuator options

Restart

現在のシーンをリスタートします。シーン内のすべてのものはリセットされます。

Set Scene

指定したシーンに変更します。

Set Camera

使用するカメラを変更します。

Add OverlayScene

別のシーンを追加し、現在のシーンの上に描画します。これはたとえばゲーム画面に体力や弾薬、スピードメーターなどの表示をオーバーレイして常に見えるようにする場合に便利です。

Add BackgroundScene

*OverlayScene*と反対に、現在のシーンの下に別のシーンを描画します。

Remove Scene

シーンを削除します。

Suspend Scene

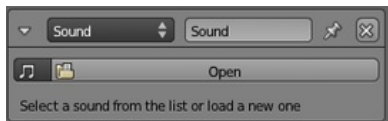
シーンを中断します。

Resume Scene

中断したシーンを再開します。

Sound Actuator

音声ファイルを一覧から選ぶか、外部の音声ファイルを新たにリンクします。



Sound Actuator

アクチュエータの共通のオプションについては [こちら](#) をご覧ください。

専用のオプション:

音声ファイル名

一覧から音声ファイルを選ぶか、**Open** を押して外部の音声ファイルをリンクします。

Play Mode: (再生方法)

Loop Bidirectional Stop: 正パルスの受信中のみ再生 + 逆再生します

Loop Bidirectional: 再生後に逆再生します

Loop End: ループ再生します

Loop Stop: 正パルスの受信中のみループ再生を続けます

Play End : 再生します

Play Stop: 正パルスの受信中のみ再生を続けます

Volume (音量)

再生音の音量を調整します

Pitch (音高)

再生音の高さを調整します

3D Sound

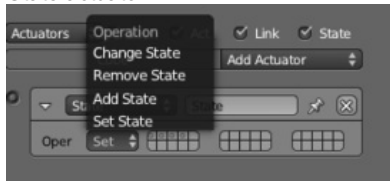
有効にすると、さまざまなパラメータを使って音の空間を表現できます

State アクチュエータ

Stateアクチュエータは簡単な操作で複雑なロジックを作成することができます。複数の違ったステート(状態)を持ち、それに対して処理をします。



State actuator



State actuator options

処理

Change State

現在のステートから指定したステートへ変更します。

Remove State

アクティブなステートのうち、特定のステートを削除します(停止します)。

Add State

アクティブなステートに、指定したステートを追加します(起動します)。

Set State

現在のステートから指定したステートに移動します。他のステートは停止されます。

Usage Notes

Stateアクチュエータを使うと、何百ものプロパティを扱わずにロジックの階層を作ることができます。うまくつかえば非常に有益ですが、pythonで問題を回避できる場合もあります。

Visibility アクチュエータ

Visibilityアクチュエータはオブジェクトの表示・非表示をコントロールします。



Visibility actuator

Visible

表示・非表示を切り替えます。

Occlusion

オクルージョンの有効・無効を切り替えます。Physicsタブで初期設定する必要があります。

Children

再起設定の有効・無効を切り替えます。有効にすると、visibilityおよびocclusionの設定が再帰的に(子オブジェクトやそのさらに子オブジェクトに)適用されます。

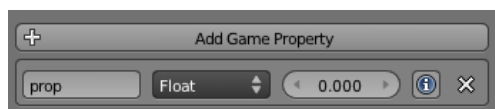
使用する際の注意

Visibilityアクチュエータを使うと描画を節約できますが、物理計算に関してはそうではありません。よってLOD(レベル・オブ・ディテール)という意味では限界があります。LODのためには[Edit Objectアクチュエータ](#)のreplace meshモードを見てください。ただし、ロジックがLODを反転してしまう可能性に気をつけてください。

プロパティ

プロパティは変数に相当するものです。これはオブジェクトに格納されており、たとえば弾薬、体力、名前などを表すことができます。

プロパティパネル



The properties panel.

View » PropertiesまたはNでプロパティを開きます。

プロパティパネルは6つの要素から成っています。

Add Property button

新しいプロパティを追加します。デフォルトではFloatプロパティで、名前はprop、すでに同じ名前がある場合は後ろに番号が付きます。

Name field

プロパティの名前。この名前を使ってPythonプログラムまたは式でプロパティにアクセスします。その場合ディクショナリ形式でアクセスします (GameObject["propname"])。名前は大文字と小文字が区別されます。

Property type menu

プロパティの種類 ([後述](#))。

Value field

プロパティの初期値。

Debug button

デバッグを有効にします。GameメニューのShow Debug Propertiesも有効にする必要があるので注意してください。Show Debug Propertiesを有効にすると、デバッグを有効にしてあるすべてのプロパティが、オブジェクト名、プロパティ名、数値、としてゲームの実行中に表示されます。プロパティが問題を起こしている疑いがある場合に便利です。

Delete button

それぞれのプロパティの横には削除ボタンがあります。

プロパティの種類

プロパティには5種類あります。

- Timer 初期値から始めて、オブジェクトが存在する限りカウントアップしていきます。たとえばゲームのステージをクリアするのにかかった時間などを調べるのにつかいます。
- Float -10000.000から10000.000までの少数。精密な数値が必要な場合に遣います。
- Integer -10000から10000までの整数。たとえば弾丸の数など、少数が必要ない場合に使います。
- String テキストを値として持ちます。128文字まで使えます。
- Boolean 真か偽の二値を持ちます。明かりのスイッチなど、二つしか状態がないようなものに便利です。

プロパティを使う

ロジックエディタではプロパティセンサとプロパティアクチュエータが使えます。

Property Sensor

Property センサには4つのモードがあります

Equal

プロパティ値とセンサの値が等しい場合に正パルスを発信します。

Not Equal

プロパティ値とセンサの値が異なる場合に正パルスを発信します。

Interval

プロパティ値がセンサのMinからMaxの値の範囲にある場合に正パルスを発信します。
最低値のみを指定したい場合は、自身のプロパティ名をMax欄に、パルスを送る最低値をMin欄に入力します (訳注: 最高値は自分自身の値なので、常に範囲内にある)。
最高値のみを指定したい場合は、自身のプロパティ名をMin欄に、パルスを送る最高値をMax欄に入力します。
値を比べるために他のプロパティ名を入力することもできます。

Changed

プロパティの値が変わったときに正パルスを発信します。

センサの基本的な使い方については[センサ](#)を、プロパティセンサについては[プロパティセンサ](#)を参照してください。

Property Actuator

Propertyアクチュエータには4つのモードがあります。

Assign

Value欄の値をProp欄に与えます。同じオブジェクトのプロパティのみに使えます。

Add

プロパティの値を増加させます。マイナスの数値をしてすれば減少させます。Boolに関しては0以外の値(マイナスでも)は真とみなされます。

Copy

他のプロジェクトのプロパティをアクチュエータのオーナーのプロパティにコピーします。

Toggle

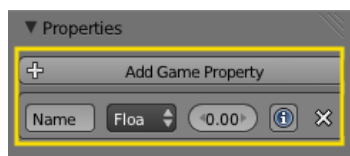
0は1に変え、0以外の値は0に変えます。オン/オフのようなスイッチに便利です。Stringプロパティには使えません。

アクチュエータの基本的な使い方については[アクチュエータ](#)を、プロパティアクチュエータについては[プロパティアクチュエータ](#)を参照してください。

Property Editing

(プロパティの編集)

Logic Editor パネル左側にあるパネルを使って、ロジックのプロパティを作成したり編集したりできます。最上部のメニューでは利用できるプロパティの型の一覧が見られます。



Property Panel

Add Property ボタン

一覧に新たなプロパティを加えます。デフォルトは“prop”という名前の Float 型のプロパティで、同名のプロパティがすでにあれば、後ろに数字がつきます。

Name 欄

プロパティの名前を入力します。Python や Expression(式)からアクセスする際に使われます。Python では辞書型の参照方式を使います (`((GameObject["propname"])`)。名前の大文字小文字は区別されます。

Type メニュー

プロパティの型を決めます(下記参照)。

Value 欄

プロパティの初期値を決めます。

I 情報ボタン

デバッグ時にプロパティを表示します。デバッグをオンにすると、ゲーム実行中画面の左上隅にプロパティの値が表示されます。デバッグをオンにするには、Game メニューにある Show Debug Properties にチェックをつけてください。デバッグ対象となったプロパティはすべて、ゲームプレイ中オブジェクト名、プロパティ名、プロパティ値で表示されます。プロパティが何か問題を起こしていると思える場合に役立ちます。

X

プロパティを削除します。

States


(ステート/状態)

BGE では、オブジェクトは複数の「ステート」を持つことができます。ゲームのプレイ中はいつでも、オブジェクトの現在のステートによって行動が決まります。例えば、ゲーム内のキャラクターが覚醒、睡眠、死亡の各ステートを持っているものとします。大きな音がしたときの彼の行動はいつも、現在の状態によって決まります。うずくまる(覚醒時)、目を覚ます(睡眠時)、何もしない(死亡時)といった具合です。

How States Operate

(ステートの操作方法)

ステートはコントローラーを通じて作成し、使用します。ステートのシステムに直接操作されるのはアクチュエータやセンサーではなく、コントローラーだけであることに注意してください。各オブジェクトは複数のステートを持つことができます(上限は 30 で、デフォルトは 1 です)、ある時点でのオブジェクトのステートは必ずひとつに決まります。コントローラーは常に、操作対象となるステートを定義しなければなりません。a) ロジックの条件を満たし、かつ b) オブジェクトが現在、指定したステートにある場合にだけ、出力パルスを送ります。ステートはオブジェクトのコントローラー設定で作成し、編集します(詳細は下記をご覧ください)。

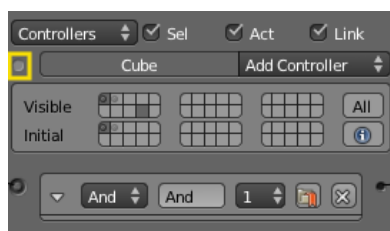
 単純なゲームでは、ステート設定は自動で行われます。デフォルトで、各オブジェクトはひとつのステートを持ち、すべてのコントローラーがステート 1 を使うよう設定されます。したがってゲームに複数のステートが必要ないのなら、ステートをあえて設定しなくともすべて動作するでしょう - ステートについて頭を悩ます必要はまったくありません。

{{{2}}}

アクチュエータの一つである State アクチュエータは、オブジェクトのステートビット値を設定したり解除したりでき、センサーの信号に対するオブジェクトの反応を現在のステートによって決めることができます。上の例では、キャラクターには「大きな音」センサーにつながった「覚醒時」「睡眠時」「死亡時」ステート用のコントローラーがあります。これでキャラクターの現在のステートによって異なった動作を行わせることができ、中にはある条件下でキャラクターのステートを切り替えるものがあるかもしれません。

Editing States

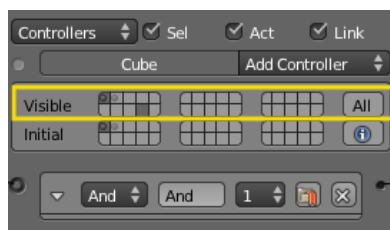
(ステートの編集)



State Panel Button

ステートは、Game Logic パネルのコントローラー(中央)列を使って作成し、編集します。ステートパネルを表示するには、ステートパネルボタンをクリックします。パネルには利用できるステートが30ずつ、二つにわけて表示されます。Visible ステートと Initial ステートです(下記参照)。ゲームのステートシステムを設定するには、オブジェクトのロジック内で、各コントローラーに妥当なステートを選びます。

オブジェクトのステートやその関連項目の表示はそのオブジェクトのステートパネルを使って行います。ボタンを使ってオンとオフを切り替えます。パネルは Visible および Initial のふたつの部分に分かれています。

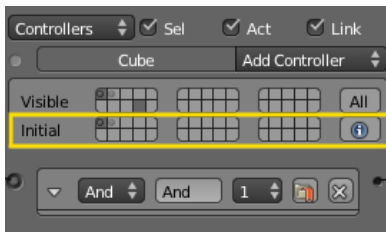


State Panel Visible

Visible ステート

Visible 領域では、利用できる30のステートがそれぞれ、薄い灰色の四角形で表現されています。このパネルは、オブジェクトのロジックブロックで見えるロジックを示しています。右側にある All ボタンをクリックすると、オブジェクトの論理ブロックのすべてが表示され(もう一度クリックすると非表示)、ステートパネルの四角はすべて薄い灰色になります。もしくは、個々のステートをクリックしてそのロジックを目に見えるようにすることができます(複数の四角をクリックできる点に注意してください)。もう一度四角をクリックすればステートの選択を解除します。

使用中のステート(=そのステートで実行されるコントローラーがオブジェクトにある場合)にはドットがつき、Game Logic にコントローラーを表示すると、四角は濃い灰色になります。列見出し部分にあるステートボタンを使って、連結されたセンサーやアクチュエータの表示を切り替えられます。



State Panel Initial

Initial ステート

Initial 領域でも、利用できる30のステートが薄い灰色で表現されています。ゲームが実行されたとき、オブジェクトの開始時のステートとして、このうち一つをクリックできます。

右側には I (Information) ボタンがあります。クリックして Game メニューの「Show Debug Properties」メニューをクリックすると、ゲームの実行中、そのオブジェクトの現在のステートが画面左上隅に表示されます。

BGE (Blender Game Engine)におけるカメラ

ゲームエンジンの内部で、カメラはデフォルトの特性(動き方)をさせることもできるし、オブジェクトを追跡することもできます。それにはオブジェクトまたはその頂点を親にするか、Cameraアクチュエータを使います。

Note

どのオブジェクトでもカメラにすることができます(後述)。またカメラにオブジェクトを追跡させる場合、どのオブジェクトでも使うことができます(Emptyなど)

カメラを使うには、カメラの視点(0 NumPad)でゲームエンジンをスタートさせます。またカメラがゆがむのを防ぐため、常にカメラオブジェクトがビューいっばいに表示されるようにズームしてください。

デフォルトのカメラ

デフォルトではカメラは動きません。

オブジェクトを親にする

カメラにオブジェクトを追跡させます。まずカメラを選択し、つぎにオブジェクトを使い選択し、CtrlP → Make Parent。

この場合、オブジェクトが回転するとカメラも回転します。それを避けるには、後述のように頂点を親にしてください。

頂点を親にする

最も簡単な方法として、オブジェクトを選択して⇔ TabでEditモードに入り、頂点を選択してObjectモードにもどります。

次に、何もオブジェクトを選択していない状態でカメラを選択し、⇧ Shiftを押しながらオブジェクトを選択します。そして⇔ TabでEditモードに入り、CtrlPを押してMake vertex parentを選びます。

これでカメラはオブジェクトを追跡するようになり、またオブジェクトが回転してもカメラは回転しません。

オブジェクトをカメラにする

どんなオブジェクトでもカメラにすることができます。またどのようなプロパティを持っていても良いです。

オブジェクトをカメラにするには、Objectモードでオブジェクトを選択し、{{Shortcut|ctrl|pad0}}を押します。

元に戻すには、カメラを選択して再度{{Shortcut|ctrl|pad0}}を押します。

レンズシフト

BlenderのインターフェースではカメラビューのXY平面上でビューをずらすオプションがあります。これはビデオプロジェクターのレンズシフト機能に似ています。この機能は通常、Y軸に沿って画像を動かします(それによって、たとえばテーブルにプロジェクターを置いたときに映像の下半分がテーブルにかかってしまう、ということを防ぎます)。

残念ながら、このパラメータはゲームエンジンで使えません。

投影を操作するには、Pythonでカメラの投影マトリックスを修正します。

```
import bge
scene = bge.logic.getCurrentScene()
cam = scene.active_camera
# get projection matrix
camatrix = cam.projection_matrix
#modifying the camera projection matrix by modifying the x and y terms of the 3rd row to obtain a shift of the rendered area
camatrix[2][0] = 2*shiftx
camatrix[2][1] = 2*shifty
cam.projection_matrix = camatrix
```

shiftxとshiftyは視野を単位とします。よってビューを半分上にずらすときは、shiftyを0.5にセットします。

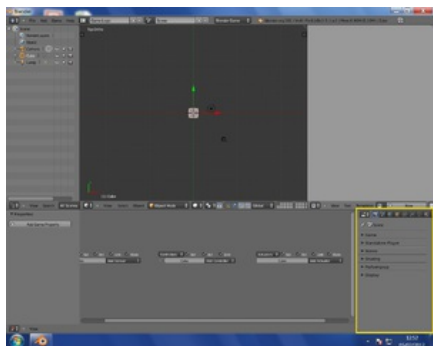
カメラの投影マトリックスの属性は初期化スクリプトが実行されたあとで設定してください。ゲームが始まった瞬間に実行すると、投影マトリックスがぐちゃぐちゃになってしまいます。

関連項目

- [Fisheye Dome camera](#).

Camera Editing

(カメラの編集)



Camera Properties

Blender Game で使われるカメラは、ゲームの描画や表示方法に幅広い影響を持ちます。設定のほとんどは、ゲームで使用するカメラのプロパティパネルから操作します。

💡 レンダリングエンジン

これらの操作を行う際には、レンダリングエンジンが Blender Game であることを確かめてください。そうでないと、実際の画面と説明が一致しなくなるでしょう！

カメラのプロパティでは六つのパネルが利用できます。それぞれ、いつもの三角ボタンを使って伸縮できます。各パネルの機能の詳細を以下で説明します。

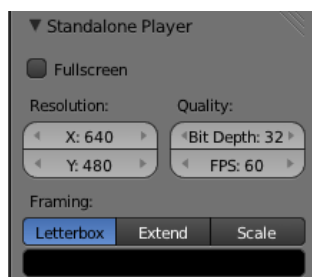
Game



Game Panel

Start ボタン - Game Engine を開始します。

Standalone Player



Standalone Panel

単体の (Standalone) ゲームプレイヤーの情報を設定するパネルです。単体のゲームプレイヤーは Blender なしで実行できます。詳細は [Standalone Player](#) をご覧ください。

Fullscreen - (フルスクリーン)

Off - 新しいウィンドウでゲームを実行します。
On - フルスクリーンでゲームを実行します。

Resolution (解像度)

X

フルスクリーン表示での描画領域の X 方向の大きさです。

Y

フルスクリーン表示での描画領域の Y 方向の大きさです。

Quality (画質)

Bit Depth (ビット深度)

フルスクリーン表示で、各ピクセルの色の表現に使われるビット数です

FPS

フルスクリーン表示での毎秒ごとのフレーム数です

Framing

描画領域にどのように表示を合わせるかを指定します

Letterbox

ウィンドウに描画領域全体を表示し、残った部分を Bar の設定色で埋めます。

Extend

視界を広げることでウィンドウに描画領域全体を合わせます。

Scale

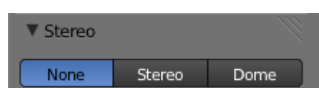
描画領域全体がきっちりウィンドウにおさまるように X か Y 方向に伸縮します

Bar Color

描画領域の周囲の余白を塗りつぶす色です。デフォルトは黒です。

カラー方式を選んでから(RGB, HSV or Hex)、スライダーやホイールを使って余白の色を選びます。

Stereo(立体視)



Stereo パネル

ゲームの立体画像を作るのに使われる、立体視の方式を選びます(もちろん、この立体表示はスタンドアロンプレイヤーで画像をレンダリングするのにも使われます)。

None

立体視効果なしで一枚の画像をレンダリングします。

Stereo

専用アルゴリズムにより、立体視用の二重の画像をレンダリングします。利用できるオプションの詳細は [立体視カメラ](#) をご覧ください。

Dome

没入型のドーム状環境の内部からゲームを見る機能です。利用できるオプションの詳細は [ドームカメラ](#) をご覧ください。

Shading



Shading Panel

ゲームのレンダリング時に使われる陰影処理の方式を指定します。[マテリアル](#) と [テクスチャ](#) で使われる陰影処理機能は、Blender Game Engine のものと基本的に同じです。ただ、リアルタイム表示による制約で、一部機能が利用できないことがあります。

Single Texture

一枚のテクスチャ用の機能を使います。

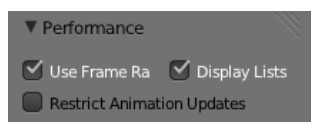
Multitexture

複数枚テクスチャの陰影処理を使います。

GLSL

GLSL の陰影処理を使います。リアルタイムな画像レンダリングの可能性があれば、常に GLSL を使うとよいでしょう。

Performance



Performance パネル

Use Frame Rate

レンダリングのフレーム数を可能な限り増やす代わりに、フレームレートを重視します。

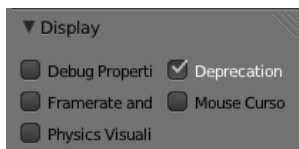
Display Lists

GPU にジオメトリを維持してレンダリングを高速化します。

Restrict Animation Updates

アニメーションの更新数をアニメーションの FPS に制限します(パフォーマンスはよくなりますが、滑らかな再生に問題を起こす可能性があります)。

Display



Display パネル

Game Engine 実行時の表示のオプションがあります。

Debug Properties (デバッグ用のプロパティ)

ゲーム実行中にデバッグ用に印をつけられたプロパティを表示します。デバッグ用プロパティは入力として指定する必要があることに注意してください(例えば Game Properties パネルで **I** ボタンを使います)。Blender 内でゲームを実行するときだけ利用できます(スタンドアロンプレイヤー版では使えません)。

Framerate and Profile

ゲーム実行中にフレームレートと分析情報を表示します。Blender 内でゲームを実行するときだけ利用できます(スタンドアロンプレイヤー版では使えません)。

Physics Visualization

ゲーム実行中に、物理演算で使われる境界線や相互作用を表示します(Blender 内とスタンドアロン版の両方で使えます)。

Deprecation Warnings

Python API の非推奨機能を使うと警告を表示します。Blender 内でゲームを実行するときだけ利用できます(スタンドアロンプレイヤー版では使えません)。

Mouse Cursor

ゲーム実行中にマウスカーソルを表示します(Blender 内とスタンドアロン版の両方で使えます)。

ステレオカメラ

ステレオカメラは、特殊な眼鏡を通すと3次元的に見えるような画像をレンダリングできます。そのために、カメラからわずかな距離だけ離れた二つの画像がレンダリングされます。それによって、われわれの両目から見た映像をシミュレートします。ステレオイメージを見るときには、片方の目は一方の画像だけを見るように制限され、もう片方の目はもう一方の画像だけを見ます。われわれの脳はその二つのイメージを合成して3次元の物体を見ているように感じるのです。

Stereo Settings

Stereo Mode

ステレオカメラの種類をさせています。これはあとで説明します。

Eye Separation

この値は非常に重要です。これは二つのイメージがどのくらいはなれているか、どのくらい「3D」であるか、ということを決めます。あまり高い値にすると頭痛が起こったり目が疲れやすくなる可能性があります。

Stereo Modes

Quad Buffer

Above-Below

Interlaced

Anaglyph

Side by Side

Vinterlaced

Blender Game Engine のドームモード

この機能は、インタラクティブなプロジェクトを没入型ドーム環境として視覚化できます。拡張性の高いツールとするため、完全なドーム、端を切り落としたドーム(ツ前方および後方)、プラネタリウム、球面ミラーによるドームをサポートしています。

この機能はBlender 2.49で追加されました。これはPaul Bourkeが開発したマルチパス・テクスチャ・アルゴリズムを使用し、**SAT** (Society for Arts and Technology)の後援のもと、**SAT Metalab** immersion research program[1]の一環としてDalai Felintoによって実装されました。簡潔に説明すると、これはシーンを4回レンダリングして、平行投影カメラで見たときに魚眼レンズのような投影に見えるようにデザインされたメッシュに貼り付けます。

注意

最大の大きさで投影するために、Blenderをフルスクリーンモードにして使ってください。 そのためには、コマンドラインで-WをつけてBlenderを起動します。また、トップメニューを消すためにすべてのウィンドウを結合します。そうでなければ、(Ctrl+Up)で最大化しても完全に全画面を使えません(トップバーメニューは約20ピクセルあります)。

Dome Camera Settings

Dome Type

メニューで使用するドームカメラの種類を選択します。それぞれのカメラの概要は後述します。また設定項目も後述します。

- [Fisheye Dome](#)
- [Front-Truncated Dome](#)
- [Rear-Truncated Dome](#)
- [Cube Map](#)
- [Full Spherical Panoramic](#)

使用できる設定項目はカメラの種類によってかわります:

Resolution

バッファの解像度。この数値を下げるとスピードが上がりますが、品質は下がります。

Tessellation

4がデフォルトです。これはメッシュの分割レベルです(Cube Mapでは使えません)。

Angle

視野の角度。範囲は90°から250°です(FisheyeとTruncatedで使えます)。

Tilt

水平軸におけるカメラの回転(FisheyeとTruncatedで使えます)。

[Warp Data](#)

映像を湾曲させるのにカスタムメッシュを使います。

Fisheye Mode

視野90°から250°までの正射影魚眼映像。

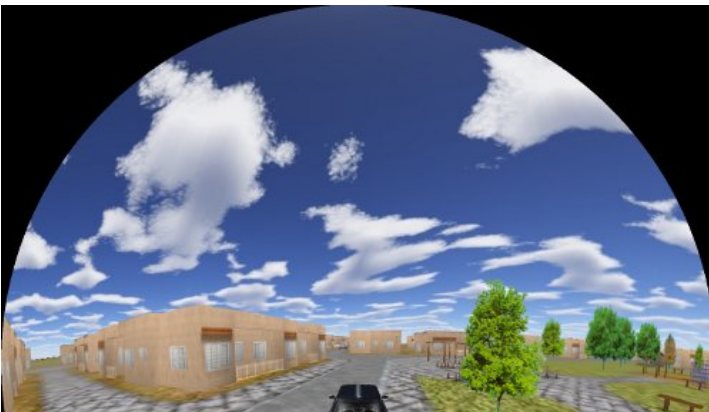
- 90°から180°までの場合、4回レンダリングします。
- 181°から250°までの場合、5回レンダリングします。



Front-Truncated Dome Mode

端を切り落としたドーム用にデザインされています。このモードでは魚眼映像がウィンドウの上部および左右に接するように配置されます。

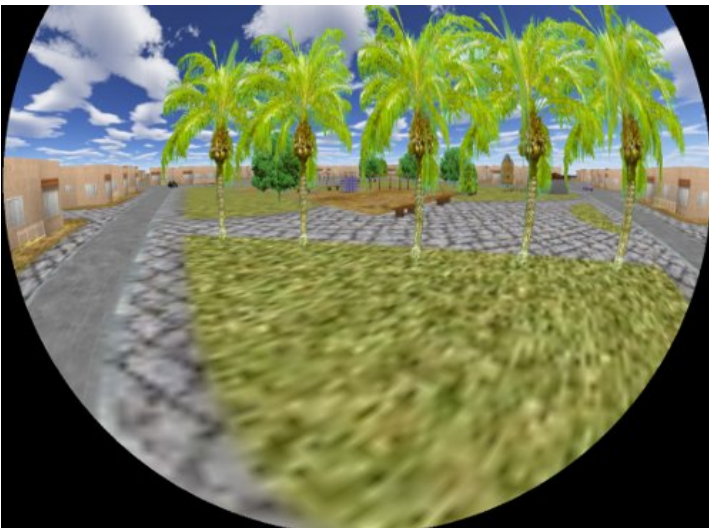
- 90°から180°までの場合、4回レンダリングします。
- 181°から250°までの場合、5回レンダリングします。



Rear-Truncated Dome Mode

端を切り落としたドーム用にデザインされています。このモードでは魚眼映像がウィンドウの下部および左右に接するように配置されます。

- 90°から180°までの場合、4回レンダリングします。
- 181°から250°までの場合、5回レンダリングします。



Cube Map Mode

このモードはキューブマッピングのプリレンダーアニメーションに使えます。

- これは6回レンダリングします。イメージの順番はBlenderの環境マップのフォーマットと同じです。
 - 1段目: 右、後、左
 - 2段目: 下、上、前



Spherical Panoramic

球面パノラマモードです。

- 6回レンダリングします。
- 上部と下部は**Definition**を5以上に設定すると正確になっていきます。



Warp Data Mesh

さまざまな投影環境を使用する場合、フラットなスクリーンとは違ってシンプルな遠近法ではないイメージが必要となることがあります。円筒ディスプレイ、プラネタリウムドームのための新しい方式、バーチャルリアリティ用の直立型ドームなどのために補正されたイメージです。

メッシュのフォーマットについては[Paul Bourkeの記事](#)を参照。



こういったイメージを生成するために、特別なフォーマットを用いています。

テンプレートは::

```
mode
width height
n0_x n0_y n0_u n0_v n0_i
n1_x n1_y n1_u n1_v n1_i
n2_x n2_y n2_u n2_v n2_i
n3_x n3_y n3_u n3_v n3_i
(...)
```

1行目はメッシュに適用されるイメージの種類です: **2** = **rectangular**(矩形), **1** = **radial**(放射状) 2行目はメッシュの寸法をピクセルに換算した数値。残りの行はメッシュの結節点 (nodes) です。

それぞれの行は **x y u v i** からなっています。(x,y)は正規化されたスクリーン座標、(u,v)はテクスチャ座標、iは乗算する強度ファクターです。

xの範囲は(-1 x アスペクト比)から(アスペクト比)まで、yは-1から1、uとvは0から1、iは0から1です。負の数が入っているとそのノードは描画されません。

- Warp Meshデータファイルを使うためには、ファイルを作成してテキストエディタに追加する必要があります。
- テキストエディタを開きます (テキストエディタウィンドウ)。
- テキストエディタで、メッシュデータファイル(たとえばmyDome.data)を開きます (Text/Open または Alt+O)。
- ゲームフレーミングセットアップに移動します (Window Types/Buttons Window/Scene Page または F10)。
- ドームモードを有効にします。
- Warp Data欄にファイル名(たとえばmyDome.data)を入力します。

Warp Mesh(イメージを湾曲させるのに使うメッシュ)の作成にはmeshmapperというインタラクティブなツールが使えます。これは[Paul Bourke's Warpplayer](#)ソフトウェアパッケージに含まれています(フルバージョンが必要です)。

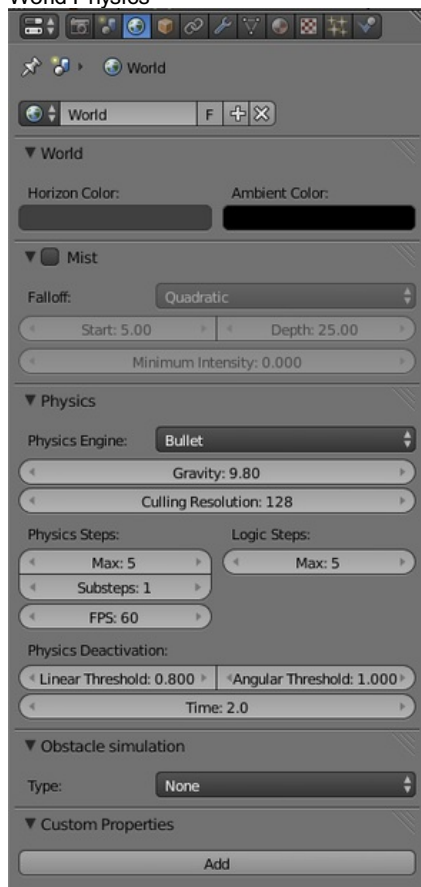
サンプルファイル

[Spherical Mirror Dome 4x3](#), [Truncated Dome 4x3](#), [Sample Fullscreen File 4x3](#), [Sample Fullbuffer File 4x3](#).

Note

重要: ビューポートはキャンバスの横／縦の比を使って計算されます。したがって、画面のサイズが変わると必要なメッシュファイルも変わります。また正しい縦横比を得るために、Blenderをフルスクリーンで使用してください。

World Physics



BGE World Panel (fully expanded)

World 設定では、すべてのシーンに作用する基本的な効果を設定でき、ゲーム全体に統一感や連続性を与えます。たとえば環境光、奥行き効果/霧、グローバルな物理設定などがあります。これらの効果は Blender レンダラーで利用できる効果の限定的なサブセットになります ([World](#) をご覧ください)。

💡 ノード合成

World 設定はシーンに簡単にエフェクトを足すことができます。一方 [compositing nodes/ノード合成 \(eng\)](#) はもっと多くのコントロールやオプションを持ち、習得が難しくなりますが、よく使われます。たとえば、Z値 (カメラからの距離) や法線 (面の方向) を compositing nodes を通じてフィルターすれば、シーンの奥行きや空間的な明瞭さを増大させることができます。

World

二種類の色の設定を使って、ゲームに汎用的な照明効果を与えられます。

Horizon Color

地平線の RGB カラー値です。明示的に色の塗られない場所の色および強度 (intensity) に使われます

Ambient Color

表面拡散で得られる背景照明全体を真似るものです ([Ambient Light](#)、[Exposure \(eng\)](#) や [Ambient Occlusion \(eng\)](#) をご覧ください)。標準となる色/強度をこのコントロールで設定します。

Mist

Mist (霧/かすみ) でレンダリングの奥行きを大きく錯覚させることができます。Mist を作るため、Blender は離れたオブジェクトほど透明度を上げて (アルファ値を下げて)、オブジェクトの色と背景色が混ざり合うようにします。詳細は [Mist](#) をご覧ください。

Mist

霧機能の有効無効を切り替えます

Falloff

霧の減衰形状を決めます

Start

霧効果の開始距離です。この距離より近いオブジェクトには霧がかかりません。

Depth

オブジェクトの不透明度がゼロになる奥行きです

Minimum intensity

霧全体の強度の最小値です

Game Physics

World パネルにある Game Physics では Gmae Engine のシーンを支配する物理ルールや使用する重力の値を決めます。選択した物理エンジンに基づいて、ゲームエンジンの物理シミュレーションにより、Blender は自動的に Actor (演者/物理演算の対象となるオブジェクト) を下方に (-Z) 動かします。Actor を配置し望みどおりに動いたら、その動きを固定された lpo 曲線に焼き込むことができます (詳しくは [Logic actors](#) をご覧ください)。

Physics Engine

使用する物理エンジン

Bullet

現在も開発が進行中の、デフォルトの物理エンジンです。運動と衝突検知を扱います。衝突したものの運動量を衝突されたものに伝えます。

None

物理をしません。ものは重力に影響されず、仮想空間内を飛び回ることができます。運動中のオブジェクトはその運動を続けます。

Gravity

重力加速度です。単位はこの World のメートル/秒⁻²です (m.s⁻²)。Actor であるオブジェクトはすべて質量と大きさのスライダーを持ちます ([Object Physics](#) をご覧ください)。フレームレート ([Render](#) をご覧ください) と組み合わせて、Blender はこの情報を使ってオブジェクトの下方への加速を計算します。

Culling Resolution

ピクセル単位で occlusion culling buffer の大きさを指定します。値が大きいくほど精度がよくなりますが遅くなります。視覚錐台/隠面消去 (view frustum and occlusion culling) 用に最適化された Bullet DBVT がデフォルトで内部実行されています。

Physics Steps

Max

グラフィックスがゲームを遅延させるときの物理演算の最小ステップを指定します。値が高いほど物理演算が現実の時間に合わせられます。

Substeps

物理演算の timestep ごとのシミュレーションのサブステップの数を指定します。値が高いほど精度が上がります。

FPS

毎秒ごとのゲームフレーム数の基準値を指定します。物理演算固定の timestep = 1/fps で、実際のフレームレートとは独立しています。

Logic Steps

グラフィックスがゲームを遅延させるときの、ゲームフレームごとの最大論理フレーム数を指定します。値が高いほど物理演算との同期がうまくいきます。

Physics Deactivation

物理演算が非アクティブになるしきい値を設定します。ゲーム中の物理シミュレーションに費やされる処理を抑えるのに役立ちます。

Linear Threshold

rigid body は、速度がこのしきい値以下になり、非アクティブ化までの時間が経過するとスリープ (運動停止) 状態に入ります (非アクティブ化までの時間を 0 にするとスリープは無効になります)。

Angular Threshold

Linear Threshold の回転版です。単位は rad/s です。

Time

このしきい値を越える時間オブジェクトが運動しない場合に物理演算を無効化/スリープします (0.0 でスリープを無効にします)。

Obstacle Simulation

(障害物シミュレーション) Game Engine で障害物の回避に使われるシミュレーションで、RVO (Reciprocal Velocity Obstacles) の仕組みに基づいています。目的は一つ以上の Actor が障害物と衝突することを避けることです。詳しくは [経路検索と進路決定の動作 \(eng\)](#) をご覧ください。

Type

None

障害物シミュレーションは無効で、Actor は障害物の回避能力を持ちません。

RVO (cells)

障害物シミュレーションが [RVO method \(eng\)](#) with cell sampling に基づくものになります。

RVO (rays)

障害物シミュレーションが [RVO method \(eng\)](#) with ray sampling に基づくものになります。

Level height

障害物が相互に作用できる高さの差の最大値です。障害物の片方がもう片方の上にあるとみなす最低距離 (高さ) の定義に使われます。上下にあるとみなされた酒害物は、お互いに影響しません

Visualization

デバッグ用に障害物シミュレーションの視覚化を行います

Game Physics

ゲーム用の物理処理の設定はWorldパネルにあります。その設定によってゲームエンジン内のシーンにおける物理のルールおよび重力が決まります。選択された物理エンジンに基づいてBlenderは自動的にアクターを下方に動かします。望みの動きになるようにアクターを調整したあと、その動きを固定したIpoカーブとしてバイクすることもできます(詳細は[ロジックのアクター](#)を参照)。

オプション

Physics Engine

使用する物理エンジンの種類を選択します。

Bullet

デフォルトの物理エンジンです。活発に開発中です。これは移動と衝突検知をあつかいます。衝突によって運動量が移動します。

None

物理計算をしません。物体は重力の影響を受けることはなく、移動しているオブジェクトはその動きを続けます。

Gravity

重力加速度です ($m \cdot s^{-2}$)。このオブジェクトは質量と大きさのスライダーを持っています([オブジェクトの物理](#)を参照)。フレームレートを考慮する際に([Render](#)を参照)、Blenderはこの情報を使ってオブジェクトがどのくらいの速さで下方に加速されるかを計算します。

Physics Steps

Max

描画のためにゲームスピードが遅くなる場合の、1ゲームフレームあたりの物理計算ステップの最大値。高い数値にすると物理計算がリアルタイムに追いつきやすくなります。

Substeps

物理計算ステップあたりのシミュレーションサブステップの数。高い数値にするほど物理的に正確になります。

FPS

ゲームフレームレートの設定値。物理計算の固定タイムステップの長さは実際のフレームレートとは関係なくこの設定値の逆数になります。

Logic Steps Max

描画のためにゲームスピードが遅くなる場合の、1ゲームフレームあたりのロジックフレーム数の最大値。高い数値にするほど物理計算と同期します。

Occlusion Culling

オクルージョン・カリング(隠れたポリゴンを省略する技術)と視野錐台に最適化されたBullet DBVTを使います。

Resolution

オクルージョンバッファのピクセルサイズ。高い数値にすると正確になります(しかし遅くなります)。

Static(静止)オブジェクト

物理計算には関わりますが、重力の影響は受けないオブジェクトです。

Actor

有効にすると、物理エンジンの中でオブジェクトは世界(単純な壁や床)とは違った扱いになります。このオブジェクトはほかのアクティブなオブジェクトから見えるようになります。

この時点では、オブジェクトは通常の物理効果(たとえば重力)の影響は受けないがただの背景とも違った見られ方をする、という状態です。これは物理的な性質(重力など)をもつ必要がないオブジェクトの場合に便利です。たとえば壁についているボタンなど。

Ghost

有効にすると、他のオブジェクトをすり抜けるようになります。これは反発や摩擦などの物理計算の時間を節約できます。他のオブジェクトとの衝突は依然として検出され、ゲームロジックに報告されます。

Invisible

有効にするとオブジェクトが「透明(Invisible)」になり、レンダリングされなくなります。

Radius

境界とする球形およびmaterial physicsの半径。

Anisotropic(異方性)

X、Y、Zそれぞれの軸における摩擦力をコントロールします。異方性は特定の方向における摩擦に関係するものです。たとえば、スケートボードのホイールは前方には転がりやすいですが、横方向には滑りにくいです。

No Collision

オブジェクトの衝突を無効にします。他のオブジェクトから決して触れられないようなオブジェクトにこれを適用すると、パフォーマンスの節約になります。

Invisible

[Static](#)を参照。

Dynamic

有効にすると、オブジェクトはリアルな物理的性質を持ちます。重力に引かれたり、他のオブジェクトにぶつかってはね返ったり、質量をもったり、摩擦によってスピードダウンしたりします。

設定

Actor

[Static](#)を参照。

Ghost

[Static](#)を参照。

Invisible

[Static](#)を参照。

Use Material Force Field

有効にすると、オブジェクトは[Dynamic Materials](#)と相互作用します。現在これはデフォルトのbullet物理エンジンではサポートされておらず、Sumoエンジンでも完全にはサポートされていません。Blender2.24では完全にサポートされています。

Rotate From Normal

オブジェクトを回転させるために法線を使います。サーフィスの方向とは違う方向を指します。

No Sleeping

デフォルトでは、回転の物理計算の適用されているオブジェクトの動きが止まると、回転物理は無効にされます。そのオブジェクトの動きを計算する必要はなくなったと物理エンジンが判断するためです。これはある場合には正しく、またコンピュータの処理能力を節約することになります。しかし別の場合、たとえばサッカーゲームなどでは、誰かがボールをまた蹴って動かすまで物理計算を維持する必要があります。

Attributes

Mass

質量は、オブジェクトを動かすのに必要な力の大きさに関わります。質量が大きいほど、オブジェクトを動かすのにより大きな力が必要になります。質量は物体が落ちる速さには関係しません。速度を落とすには[dampening](#)を使ってください。物体が落ちる速度を上げるには[gravity](#)を上げてください(通常は9.81です)。

Radius

球形の境界(bound)を設定した場合の球の半径、およびFh/FhRotを決めます。オブジェクトの見たと実際の大きさの関係を変えるためにこの数値を変えます。境界はBoundsで有効になり、そのデフォルトモードはBoxです。

境界の半径を変える際、画面上に円が表示され効果を確認できます。この瞬間は3Dウィンドウにおいてすべての境界形状が円として反映されます。それはゲーム中の反応とは違う場合があります。

Form Factor

剛体(rigid body)オブジェクトがコントロールされる度合いです。数値を高くすると、オブジェクトが転がりにくくなります。とくに平らな表面上において効果ができます。

Anisotropic Friction

[Static](#)を参照。

Velocity (速度)

Minimum

速度の最小値を設定します。

Maximum

速度の最大値を設定します。

Damping (減衰)

Translation

減衰はオブジェクトの動きの自由度に影響します。宇宙空間では減衰はほぼまったくないと考えられます。一方、水中の場合は減衰の効果を非常に高くするべきです。

Rotation

通常の減衰と似ていますが、これはオブジェクトの回転に影響します。数値を高くしても、他の種類の運動量には影響しません。

Lock Translation and Rotation

オブジェクトの回転および移動を、任意のX、Y、Z軸の組み合わせのみに制限できます。

Collision Bounds

[Object Type](#)を参照。

Rigid Bodies

設定項目は[Dynamic Objects](#)と同じです。

これは回転の物理計算を有効にして、オブジェクトに生き生きした感じを与えます。たとえばボールが床に落ちた場合、床が水平だとしてもその場で止まるのは不自然でしょう。回転の計算を有効にすると、たとえ完璧に水平な面にオブジェクトが落ちたとしても自然な感じに転がるようになります。

Soft Body

衝突したときに変形するオブジェクトです。

Settings

Actor

[Static](#)を参照。

Ghost

[Static](#)を参照。

Invisible

[Static](#)を参照。

Attributes

Mass

[Dynamic](#)を参照。

Welding

結合のしきい値。これよりも近い距離にある頂点は同じものであるとみなされます。0.0に設定すると、チェックが無効になり、シーンのローディングが速くなります(重複した頂点がなければ無効にして大丈夫です)。

Position Iterations

位置の計算を反復する回数。数値を大きくすると正確になります。

Linear Stiffness

柔体のリンクの直線的な硬さ。

Friction

動摩擦です。

Margin

衝突計算の余裕。小さい値にするとアルゴリズムが不安定になります。

Bending Constraints

コンストレイントの制約も曲げます。

Shape Match

Thresholdにもとづいてシェイプマッチングします。

Cluster Collision

Rigid to Soft Body

柔体と剛体の間でcluster collisionを有効にします。

Soft to Soft Body

柔体同士の間でcluster collisionを有効にします。

Iterations

クラスタの反復数。

Collision Bounds

[Object Type](#)を参照。

Occluder

選択的オクルージョン(occlusion=ふさぐ)とは、オブジェクト(オクルーダー)が他のオブジェクトを隠す(GPUに送らないようにする)能力のことです。実装はtree selectionと同じです。

オクルージョンはデフォルトでは有効になっていません。これを有益に使うためにはよく理解する必要があるからです。適切に使用されない場合、ゲームがスローダウンすることがあります(ただしそれほど多大にはありませんが。処理に使う時間は自動的に調整されます)。

動作の仕組み

オクルージョンはすくなくとも一つのオクルーダーが存在する場合に起動します。オクルーダーの設定はphysicsパネルにあります。

オクルーダーはネットワークベースのオブジェクトのみに使われなくてはなりません。これは物理計算に関する限り、衝突計算をしないということと同じです。オクルーダーモードが他の物理モードと相互排他的である理由は、オクルーダーはそれ専用でデザインされるべきで、すべてのメッシュをオクルーダーにするべきではない、ということを強調するためです。しかしながら、Pythonとロジックブリックを使って、他の物理オブジェクトにオクルーダーの機能を与えることも可能です。これについては後で触れます。

オクルーダーオブジェクトが視野錐台に入ってくると、ゲームエンジンはそのオブジェクトの面からZデプスバッファを作成します。面が片面表示か両面表示かは重要です。表側が見えている面および両面表示になっている面だけがZデプスバッファの作成に使われます。複数のオクルーダーが視野錐台内にあるときは、ゲームエンジンはそれらを結合して最も手前にある面を残します。

Zデプスバッファの解像度はWorld設定の"Occlu Res"ボタンでコントロールできます。

デフォルトでは、解像度はビューの最も長い辺に対して128ピクセル、他の辺に対しては長さに応じて設定されます。128というのは非常に低い解像度ですが、カリング(描画から除外すること)のためには十分です。解像度は1024まで上げられますが、CPUの使用も増大します。

ゲームエンジンはDBVTツリーを走査して、個々のノードが完全にオクルーダーに隠されるかをチェックします。隠される場合は、そのノード(およびそれが内包するすべてのオブジェクト)を消去します。

更なる最適化のため、ゲームエンジンはすくなくとも一つのオクルーダーが視野錐台にあるときのみ、Zデプスバッファを作成、使用します。それ以外の場合はオクルージョンを使わない場合と比べてパフォーマンスの低下はありません。

使い方

オクルージョンカリングを使うのが有益でない場合もあります:

- オクルーダーが小さく、多くのオブジェクトを隠さない場合。
この場合、オクルージョンカリングはCPUを遅くするだけです。
- オクルーダーが大きいが、隠されるオブジェクトがシンプルな場合。
この場合、オブジェクトをGPUに送ってしまったほうが良いでしょう。
- オクルーダーが大きく、多数の複雑なオブジェクトを隠すが、それが非常に予想しやすい状況の場合。
たとえば、複雑なオブジェクトがたくさん詰まった家などです。ここでオクルージョンカリングを使ってもうまく働きますが、特定のロジックを作って表示・非表示を切り替えたほうがもっとパフォーマンスはあがります。たとえば、カメラが屋内にあるときだけ表示されるようにオブジェクトを設定するなど。

オクルージョンカリングが最も意義を持つのは、オクルーダーが大きく(ビルや山など)、多数の複雑なオブジェクトを隠し、それが結果を予想しにくい状況である場合です。しかしながら、パフォーマンスをそれほど気にし過ぎなくても良いです。適切でない使い方をしたとしても、パフォーマンスの低下はある程度に抑えられるアルゴリズムになっています。

オクルーダーは視覚的に描画されるオブジェクトであってもかまいませんが、面が多すぎるとZデプスバッファの作成に時間がかかるので注意してください。たとえば、地表はオクルーダーの候補としてはあまり良くありません。面や重なっている部分が多すぎます。オクルーダーをシンプルかつ透明(invisible)にして、複雑なオブジェクトの内部に配置することもできます(たとえば複雑な建物の壁の内側に沿って)。オクルーダーに穴をもたせて、そこからオブジェクトが見えるようにすることもできます。

複雑な地形からオクルーダーを作るときにありうる手順を以下に示します。

- 地形を複製して、低ポリゴンに作り直す。
- 水平になっているパーツをすべて削除して、急勾配な部分だけを残す。
- すべての面を透明にして、オクルーダーモードをセットする。

さらに最適化するため、オクルーダーをいくつかのオブジェクトに分割して(個々の山や壁からそれぞれオクルーダーを作る)、地表と同じ作業をします。このようにすると、オクルーダーに隠された地形がレンダリングされなくなります。

ワイヤーフレームモードでゲームを実行すると、オクルージョンが動作していることを確認できます。

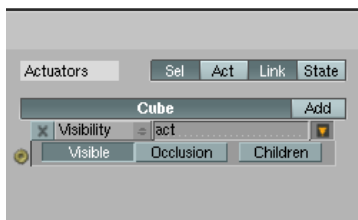
ゲーム内でのコントロール

Pythonを使って、オブジェクトのオクルージョン能力を有効/無効にできます。

```
obj.occlusion = True
obj.setOcclusion(True, False) #param1=occlusion, param2=recurse in children
```

オブジェクトは必ずしもGUI上でオクルーダーとして設定されていなくてもかまいません。どのようなメッシュでも、staticオブジェクトでもdynamicオブジェクトでも、オクルージョン能力を有効にできます。

Visibilityアクチュエータを使うこともできます:



アクチュエータのレイアウトが変わったので注意してください。排他選択のVisibleボタンとInvisibleボタンは、別々の機能に関する二つのオン・オフボタンに置き換えられました (Visible/InvisibleとOccluding/Non-occluding)。

アクチュエータを使う場合は、VisibilityとOcclusion両方の設定 (有効／無効) を指示したことになります。個別に設定する場合はPythonを使ってください。

既知の制限

- オクルージョンリングを使う場合、ゲームが始まった時点で少なくとも一つのオクルーダーが存在していなくてはなりません。オクルーダーはアクティブなレイヤーにある必要はありません。その後は、どのメッシュオブジェクトに対してもオクルージョンの有効・無効をゲーム内で設定できます。

Sensor

Sensorはstaticおよびdynamicオブジェクトを検知しますが、他の衝突、センサ、オブジェクトは検知しません。これはNearセンサやRadarセンサを設定した物理オブジェクトに似ています。NearオブジェクトやRadarオブジェクトと同様に、

- staticかつghostです。
- デフォルトでinvisibleです。
- 正しく衝突を検出するため、常にアクティブです。
- staticオブジェクトおよびdynamicオブジェクトを検出します。
- その親オブジェクトとの衝突は無視します。
- broadphaseフィルタリングします。そのベースは、
 - Actor option: 衝突が検出されるためにはActorを有効にしている必要があります。
 - property/material: 適用されている衝突センサにおいて指定されているもの。

Broadphaseフィルタリングはパフォーマンスのために重要です。衝突点はbroadphaseフィルタを通ったオブジェクトの場合だけ計算されます。

- 衝突センサがアクティブでない場合は自動的にシミュレーションから除去される。

NearオブジェクトやRadarオブジェクトとは違う点は、

- どのような形状をとることもできます。triangle meshでも良いです。
- デバッグのために視覚化できます (Visibleアクチュエータを使ってください)
- 複数の衝突センサがこれを使えます。

その他の点として、sensorオブジェクトはふつうのオブジェクトです。字通に動かしたり、親子の関連付けをしたりできます。dynamicオブジェクトを親にすると、そのオブジェクトに対して高度な衝突コントロールを提供できます。

衝突検知の種類は形状によって異なります。

- box、sphere、cylinder、cone、convex hullではボリュームの衝突検知をします。
- triangle meshではサーフィスの衝突検知になりますが、マージンを設定すればサーフィスにボリュームを与えることもできます。マージンはサーフィスの両面に設定されます。

パフォーマンスのヒント:

- SensorオブジェクトはNearやRadarよりも上手く動作します。シーングラフ最適化によってsynchr

onizationsが少なく、また複数の衝突センサを持つことができます (たとえばそれぞれ違ったプロパティによってフィルタリングしたりなどできます)

- 可能な場合はシンプルな形状 (boxやsphere) を使ってください。
- 常にbroadphaseフィルタリングを使用してください (フィルタリングするプロパティやマテリアルを設定せずにセンサを使わないでください)。
- 衝突センサは必要なときだけ使用してください。センサオブジェクト上で衝突センサが無効になっているときは、シミュレーション内から削除され、CPUを消費しません。

わかっている制限:

- Blenderをデバッグモードで実行している場合、このような警告が出ます。
"warning btCollisionDispatcher::needsCollision: static-static collision!"
リリースモードではこのメッセージは出ません。
- sphere、cone、cylinderでは衝突マージンは効果がありません。

設定

Invisible

[Static](#)を参照。

Collision Bounds

[Object Type](#)を参照。

Performance

(性能)

ゲームを開発するときにゲーム開発者やソフトウェア/ハードウェア開発者は、ユーザーがそのゲームで最高の体験ができる基本利用環境を決め、いくつかのツールを使ってゲームを特定のプラットフォームや OS にあわせて最適化します。

こうしたツールのほとんどは、ゲームが開発実行される特定のゲームエンジンに向けたソフトウェアです。

Blender Game Engine にも開発中のゲームを最適化するための視覚的なツールがあり、ゲーム開発者は最適な利用環境と、ゲーム実行に最低限必要なソフトウェアおよびハードウェアをテストすることができます。

こうしたツールを、Blender では Properties Window の Render コンテキストにある System および Display から利用できます。特定の性能調整と計測用オプション、ゲーム実行中のフレームレートや Blender ウィンドウ(ゲームビューポート)へのレンダリング方法の調整オプションの他、グラフィックカードのメモリに割り当てられたジオメトリの管理用コントロールがあります。

Blender Game Engine レンダリングシステムのコントロール

[System/システム](#) - ゲーム実行中の、シーンのレンダリング用コントロール

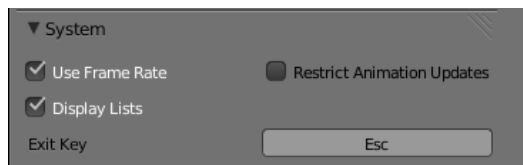
Blender Game Engine 性能の計測

[Display/表示](#) - ゲーム実行中の、性能に関する特定データの表示用コントロール

System

プロパティウィンドウの Render タブにある System では、ゲーム開発者はフレーム廃棄やフレーム描画の制限に関わるシステム性能のオプション、Blender Game Engine を止めるキー、ジオメトリの管理をグラフィックカードの内蔵メモリで行う、といったオプションを指定できます。

Options



System tab at the Render Context

Use Frame Rate (フレームレートの使用)

フレームレートの制限なしに実行できるかどうかを決めます。フレームレートはプロパティウィンドウの Render タブにある Display で指定します。フレームレートに関するさらなる情報は [Display](#) をご覧ください。

Display Lists (表示一覧)

有効にすると、Blender は GPU メモリに割り当てられているメッシュジオメトリの一覧を管理します。ジオメトリとテクスチャを割り当て可能な GPU メモリがあれば、ゲーム実行中のビューポートレンダリングを高速化できます。

Restrict Animation Updates (アニメーション更新の制限)

有効にすると、[Display](#) タブで指定された値よりも GPU のフレームレートが大きくなると、ゲームエンジンがフレームを廃棄します (再描画の途中であっても廃棄し、画像が「引き裂かれたように」ひずむことがあります)。

Exit Key

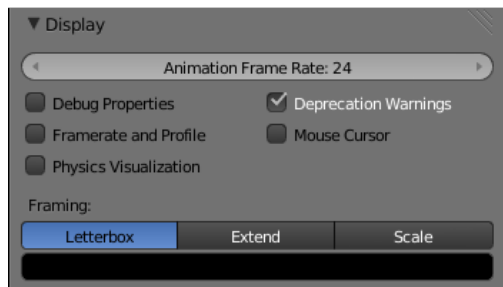
実行中のゲームエンジンを終了するキーを設定できます。このボタンをクリックして、終了に使用したいキーを入力します。デフォルトは Esc キーです。

Display

(表示)

プロパティウィンドウの Render にある Display では、ゲーム実行中のアニメーションの最大フレームレート、フレームレートや分析情報の表示、デバッグ用のプロパティ、物理演算用のジオメトリの視覚化、警告、ゲーム実行中のマウスカーソルの表示、指定解像度を持つウィンドウに合わせてゲームの画面枠の決め方を変えるオプションを指定できます。

Options



Render にある Display

Animation Frame Rate

ゲーム実行中の最大フレームレート。最小は **1**、最大は **120** です。

Debug Properties

有効にすると、ゲーム実行中にデバッグする設定にしていたプロパティの値が [フレームレートや分析情報](#) といっしょに表示されます。

Framerate and Profile

有効にすると、ゲーム実行中に行われる計算値と、デバッグする設定のプロパティをすべて表示します。各値については [フレームレートや分析情報](#) で説明されています。

Physics visualization

物理演算に使われる(殻や衝突形状のような)範囲や相互作用を視覚的に表示します。

Deprecation Warnings

非推奨の機能が使われると警告を表示します。非推奨の機能とは古すぎるか利用不能な OpenGL グラフィックカードの機能などが該当します。

Mouse Cursor

ゲーム実行中にマウスカーソルを表示するか非表示にするかを選びます。

Framing

Blender Game Engine の画面枠の決め方を *Letterbox*、*Extend* および *Scale* から選べます

Letterbox

必要なら縦横に余白を置いて、表示ウィンドウにゲームの描画領域全体を表示します。

Extend

縦または横方向に視界を広げることで、表示ウィンドウにゲームの描画領域全体を表示します。

Scale

表示ウィンドウに合わせて、ゲームの描画領域を伸縮します。

Color Bar

画面枠を **Letterbox** 形式にしたときの余白の色を指定できます。

Python API

このページは作成途中です。

完全な Python API の解説は [Python API](#) に記載されています。

さらなる情報:

- [Bullet physics](#)
- [Video Texture](#)

Bullet physics Python API

Bullet PhysicsはBlenderゲームエンジンにおいて衝突検出と剛体力学を提供します。以前の衝突検出システム(Sumo)と同じ設定項目についてはBlender上で設定できます。

しかし、新しい機能に関してはまだユーザーインターフェースがありません。よって今のところPythonを使って設定することになります。

機能:

- 乗り物シミュレーション。
- 剛体コンストレイント。ヒンジおよびpoint to point(ボールジョイント)
- 内部の物理設定へのアクセス。

Bullet physicsデモを見るのが最も使い方がわかりやすいでしょう。更なる情報は[こちら](#)。

Pythonスクリプトの例:

```
import PhysicsConstraints
print dir(PhysicsConstraints)
```

The VideoTexture module

VideoTextureモジュールを使うと、ゲームの実行中にテクスチャを操作できます。テクスチャに使えるソースはいくつかあります、ビデオファイル、イメージファイル、ビデオキャプチャ、メモリバッファ、カメラレンダー、またはこれらの混合です。ビデオファイルとイメージファイルは、ファイル名の代わりにURLを指定することでインターネットからロードできます。さらに、それらのイメージをGPUに送る前にフィルタを適用し、blue screen、color band、gray、normal mapなどの映像効果をかけることもできます。VideoTextureはイメージやビデオをロードするのにFFmpegを使用しており、FFmpegがサポートしているフォーマットはVideoTextureでもすべてサポートされています。以下はサポートしているフォーマットの一部です。

- AVI
- Ogg
- Xvid
- Theora
- dv1394 camera
- video4linux capture card (this includes many webcams)
- videoForWindows capture card (this includes many webcams)
- JPG

動作の仕組み

原理は簡単です。まず、すでに存在しているテクスチャをオブジェクトと名前によって指定し、次に動的コンテンツを持ったテクスチャを作成して、二つのテクスチャをGPU内で入れ替えます。ゲームエンジンはテクスチャが交換されたことには気づかず、通常通りにオブジェクトを表示します。それでいて、あなたがテクスチャをコントロールできるようになっているわけです。使用を終えると、新しいテクスチャは削除され、古いテクスチャが復帰します。

このページでは簡単な例を使ってVideoTextureモジュールを解説します。

ゲームの準備

VideoTextureモジュールを使う前に、適切なテクスチャをもったオブジェクトがなくてはなりません。

たとえば、ゲームの中に、本物の番組を流しているテレビを配置したいとします。テレビのオブジェクトを作成し、画面の部分には別個のテクスチャをUVマッピングで適用します。たとえば“tv.png”というテクスチャだとしましょう。このテクスチャの内容はここでは重要ではありませんが、おそらく電源が切れている画面のような、暗いグレーにすると良いでしょう。テレビの電源を入れなければならないとき、ビデオキャプチャカードからの映像を使った動的テクスチャが作成されるようにします。それをtv.pngと入れ替えれば、テレビは本物同様になります。

VideoTextureが入れ替えられるテクスチャの作り方は二通りあります。

1. 単純なUVテクスチャ。
2. テクスチャチャンネルを持ったBlenderマテリアル。

VideoTextureはテクスチャのレベルで動作するので、ゲームエンジンがもっている凝ったテクスチャ機能も使えます。GLSL、multi-texture、custom shadersなど。

例

たとえば、ゲーム内のあるオブジェクトが、一つまたはそれ以上の面に対してマテリアル／イメージを適用されており、そこにビデオを表示したいとします。

最初のステップはTextureオブジェクトの作成です。これは一度だけ実行されるスクリプトで行うことにしましょう。ゲームがスタートした時点では、ビデオはテクスチャを更新しない限り表示されないかもしれませんが、これについてはあとで触れます。通常、スクリプトはビデオを表示させるオブジェクトに付加します。そうするとオブジェクトの参照を簡単に取得できます。

```
import VideoTexture

contr = GameLogic.getCurrentController()
obj = contr.owner

if not hasattr(GameLogic, 'video'):
```

“video”属性をチェックするのは、テクスチャの作成を一度だけにするためです。

マテリアルの取得

```
matID = VideoTexture.materialID(obj, 'IMvideo.png')
```

VideoTexture.materialID()は便利な関数で、これによりvideo.pngをテクスチャとして使っているマテリアルを取得します。この方法はBlenderマテリアルでもUVテクスチャでも使えます。UVテクスチャの場合は、そのテクスチャが適用されている面に付随する内部的なマテリアルを呼び出します。Blenderマテリアルの場合は、そのテクスチャを第一テクスチャとして使っているマテリアルを呼び出します。

“IM”の接頭辞は、探しているのがテクスチャであると示すためのものです。マテリアルの場合は“MA”を先頭につけます。たとえば、このオブジェクト上でVideoMatという名前のマテリアルを探す場合、コードは以下ようになります。

```
matID = VideoTexture.materialID(obj, 'MAVideoMat')
```

テクスチャの作成

VideoTexture.Textureクラスは、動的テクスチャをGPUにロードするTextureオブジェクトを作成します。コンストラクタは必須の引数1つと、オプション

ンの引数3つをとります。

gameObj

ゲームオブジェクト。

materialID

VideoTexture.materialID() で返されるのと同じマテリアルインデックス。0にすると最初のマテリアルになります。

textureID

複数のテクスチャが存在する場合のテクスチャインデックス。0にすると最初のチャンネルになります。UVテクスチャの場合はこの値は常に0にします。

textureObj

再使用したいテクスチャを持つTextureオブジェクト。

この引数を使う場合、そのテクスチャ上にはいかなるソースも作成するべきではありません。また、テクスチャを更新する必要もありません。得られるテクスチャはマテリアルにもテクスチャにも使用できます。

```
GameLogic.video = VideoTexture.Texture(obj, matID)
```

テクスチャを永続的にする

以前にオブジェクトに対してGameLogic “video”属性を割り当てましたが、その理由は、Textureはゲームの実行中に永続的でなければならないからです。ローカル変数はスクリプトの終わりで消去され、同時にGPUテクスチャも消去されます。GameLogicモジュールオブジェクトは永続的にしたいオブジェクトを保存するのに便利なのです。

ソースを作成する

Textureオブジェクトを得ることができましたが、テクスチャのソースがないので、これだけでは何もできません。VideoTextureで使用可能な種類の中から、ソースオブジェクトを作成する必要があります。

VideoFFmpeg

動画。

ビデオファイル、ビデオキャプチャ、ビデオストリーミング

ImageFFmpeg

静止画

イメージファイル、ウェブ上のイメージ

ImageBuff

メモリ上のイメージ

CGI、グラフィックアプリケーション

ImageViewport

ビューポートの全体または一部(つまりアクティブカメラによるレンダリングイメージ)

ImageRender

アクティブでないカメラによるレンダリングイメージ

ImageMix

上記のうちの二つ、またはそれ以上の混合

この例ではソースとしてシンプルなビデオファイルを使います。VideoFFmpegコンストラクタはファイル名を引数に取ります。ファイルの場所についての混乱を避けるため、GameLogic.expandPath()を使ってファイルのフルパスを取得します。ここでビデオファイルは.blendファイルと同じディレクトリにあるとします。

```
movie = GameLogic.expandPath('//trailer_400p.ogg')
GameLogic.video.source = VideoTexture.VideoFFmpeg(movie)
```

ビデオソースオブジェクトを作ったら、それをTextureオブジェクトのsource属性に割り当てることで、ソースのセットアップと永続化を行います。ソースオブジェクトそのものも永続的になります。

Textureのソースはいつでも変更できます。たとえばゲームの進行中に二つの映像を切り替えたい場合、以下のようにします。

```
GameLogic.mySources[0] = VideoTexture.VideoFFmpeg('movie1.avi')
GameLogic.mySources[1] = VideoTexture.VideoFFmpeg('movie2.avi')
```

そして実行中に再割り当てします。

```
GameLogic.video.source = GameLogic.mySources[movieSel]
```

ソースのセットアップ

VideoFFmpegソースはムービーの再生をコントロールする属性がいくつかあります。

range

[start,stop] (floats).

ビデオの再生範囲。始めからの秒数で指定します。デフォルトではビデオ全体です。

repeat

(*integer*).

リピート回数。-1で無制限にリピートします。

framerate

(*float*).

相対的なフレームレート。1.0より小さくすると遅くなり、大きくすると速くなります。

scale

(*bool*).

Trueにすると高速なニアレストネイバーアルゴリズムで拡大します。

テクスチャに使うビデオの幅と高さは2の乗数でなければいけません。そうでない場合、指定した方法でスケールされます。デフォルトでは正確ですが遅い`gluScaleImage()`関数を使います。一番良いのは最初からビデオを適切な大きさにしておくことです。そうすれば実行時にはスケールの必要がなくなります。

flip

(*bool*).

Trueにするとイメージは上下に反転されます。

FFmpegからのイメージは常に上限判定している状態なので、この属性はデフォルトでTrueになっています。

filter

ビデオをGPUに送る前にフィルタを適用します。

`VideoTexture`フィルタオブジェクトのうちのひとつを指定します。デフォルトではイメージは何も変更されずにGPUに送られます。ビデオにアルファチャンネルが存在している場合は、それは自動的にロードされてGPUに送られます。

ここでは`scale`属性をTrueにしましょう。なぜなら`gluScaleImage()`は本当に遅くて、リアルタイムビデオには向いていないからです。ビデオの縦横がもとと2の乗数になっている場合は、この設定は何の影響もありません。

```
GameLogic.video.source.scale = True
```

Play the video

これでビデオを再生する準備ができました。

```
GameLogic.video.source.play()
```

ビデオの再生はバックグラウンドプロセスではありません。テクスチャを更新して初めて再生されます。よって、毎フレーム実行されるスクリプトで`Texture`オブジェクトの`code>refresh()`メソッドを呼び出す必要があります。

```
if hasattr(GameLogic, 'video'):
    GameLogic.video.refresh(True)
```

ビデオソースが停止している場合は、`refresh()`は何の効果もありません。`refresh()`の引数は、テクスチャを再計算する必要があるかどうかを指定するフラグです。ビデオ再生の場合は、あきらかにTrueでしょう。

ビデオのステータスをチェックする

ビデオソースクラス(`VideoFFmpeg`など)は`status`という属性を持っています。ビデオが再生中の場合、この値は2です。停止している場合は3です。よって今回の例では、

```
if GameLogic.video.source.status == 3:
    #ビデオは止まっている
```

高度なワークフロー

`Texture.refresh()`メソッドでTrueを指定すると、一度GPUに送られたイメージは次のフレームでは無効化され、ソースから新しくロードされます。このやり方では、Pythonでイメージが使用できなくなるという副作用があります。よってソースの`refresh()`メソッドを手動で呼び出すという方法もあります。

高度なワークフローの例は以下ようになります。

- Pythonでイメージバッファを使う (テクスチャには影響しない):

```
GameLogic.video.refresh(False)
image = GameLogic.video.source.image
# imageはRGBAピクセルのバイナリ列
# ... imageを使う
# 次のフレームのため無効にする
GameLogic.video.source.refresh()
```

- Pythonで使うため、GPUにはダウンロードせずにイメージをソースからロードする:

```
# テクスチャのほうのrefreshを呼んでいないことに注意
# テクスチャオブジェクトを使わずにソースオブジェクトだけを作成することもできます

image = GameLogic.video.source.image
# ... imageを使う
GameLogic.video.source.refresh()
```

- もしメッシュに適用されているマテリアルが二つ以上あり、特定のマテリアルのテクスチャを修正したい場合、マテリアルのIDを取得します。

```
matID=VideoTexture.materialID(gameobj, "Mamat.001")
```

GLSL マテリアルは2チャンネル以上のテクスチャを持つことができます。テクスチャはテクスチャスロットの順番に応じて特定されます。ここでは2としましょう。

```
tex=VideoTexture.Texture(gameobj, matID, 2)
```

デモのダウンロード

[デモファイル](#)をダウンロードできます。

高度なデモ

[このデモ](#)は、同じテクスチャ上で二つのビデオを交互に使うデモンストレーションです。このデモを使うにはelephant dreamの予告編(teaser)のビデオファイルが必要です。他のファイルで置き換えることもできます。

[このデモ](#)はImageMixソースを使ったデモンストレーションです。ImageMixは他のTextureソースを必要とします。たとえばVideoFFmpeg、ImageFFmpeg、ImageRenderなど。これらのソースをsetSource()でセットして、setWeight()でそれぞれの相対的な重みを設定します。重みは0から255で、すべてのソースの重みの合計が255にならなければいけません。ImageMixはすべてのソースを重みに応じて合成します。元になる複数のソースは、すべて同じサイズ(一番近い2の乗数にスケールされた後で)でなければいけません。そうでない場合、コンソールでPythonのエラーが出ます。

Standalone Player

スタンドアロンプレイヤーを使えば、Blender のシステムを読み込まずに Blende Game を実行できます。Blender の詳しい知識がない人にもゲームを配布できるようになります(また、無許可の編集もできなくします)。「Save as Game Engine Runtime」は便利なアドオンで、使用前に読み込んでおく必要があります。

次の手順で、動作しているゲームのスタンドアロン版を作れます。

- File » User Preferences » Addons » Game Engine » Save as Game Engine Runtime » Install Addon(ボタン)
(**Save as Default** ボタンを押して Blender の起動時にアドオンを毎回読み込むようにもできます)。
- File » Export » Save as Game Engine Runtime » (ディレクトリとファイル名を指定) » Save as Game Engine Runtime (ボタン)

これで、適切な .exe ファイルを実行するとゲームが起動します。必要なライブラリはアドオンが自動的に読み込むことに注意してください。

作成したゲームのライセンスに興味をお持ちなら、関係する議論について [Licensing \(eng\)](#)をお読みください。



Exporting...

ゲームを他のコンピューターに export するときは、ゲームの runtime や付随するライブラリなどのために新たなディレクトリを作ってください。それから、「ディレクトリをまるごと」、目的のコンピューターに転送します。

Licensing of Blender Games

(Blender ゲームのライセンス)

Blender はオープンソースソフトウェアであるため、Blender Game Engine とスタンドアロンプレイヤーで配布されるゲームのライセンスは複雑です。この記事ではその問題点と、考えられる解決策を挙げます。

Blender は GNU General Public License (GPL) の下、オープンソースソフトウェアとして配布され、所有されます。簡単に言えば、Blender システムそのものが誰にでも利用できる状態である限りは、あなたが Blender を使って作成したもの(スクリプト、テクスチャ、レンダリングした芸術作品等)をすべて所有できます。詳細は <http://www.blender.org/education-help/faq/gpl-for-artists/> をご覧ください。

Standalone Player License

(スタンドアロンプレイヤーのライセンス)

残念ながら、このルールは Blender のスタンドアロンプレイヤーで実行されるゲームや芸術作品には当てはまりません。ゲームを配布するには実行形式のファイル(run time)を作る必要があります。実際には Blender の .blend ファイルがスタンドアロンプレイヤーの「内部」に置かれます。このとき、Blender Game Engine に関連する機能だけを含む最小限の Blender が使われます。こうしてできた実行形式ファイルは元のプログラムの「派生品」(すなわちあなたのファイルとスタンドアロンプレイヤーの混成物)に分類されるため、GPL ライセンスである必要があります。

Distributing Games

(ゲームの配布)

あなたの作ったゲームを妥当なライセンス保護をもって配布する方法が、四つ考えられます。

- 1) あなたの作った Blender Game をライセンスで一切保護しません。ほんとうにライセンスが必要ですか？ 古いことわざを思い出してください:「真似されてこそ本物」
- 2) 何も問題がないふりをします。BGE ベースのゲームを配布した誰かを、Blender Foundation が告訴するようなことは非常に考えにくい話です。
- 3) Blender Game を実行するための 外部システムを使います。例えば BPPlayer や Gamekit があります(ただ、完全なテストはされていません)
- 4) 基本の .blend ゲームファイルを開始させる Game Actuator を使います。まず、別の .blend ファイルを作り、あなたの作ったゲームの内容全体を読み込んで実行するゲームアクチュエータ(Game Actuator)を作ります。次にこのアクチュエータを起動する常時センサ(Always Sensor)を作ります。これで問題を回避できます。あなたの作ったファイルはスタンドアロンプレイヤーの「外部」にあり、GPL で公開する必要はなく、「合法的に保護された」状態です。あなたのゲームはこの仕組みで完全には保護されていませんが、一般に配布されているゲームの大半は似たような保護レベルであると言えるでしょう。.blend ファイルにはアクセス可能ですが、あなたの希望するライセンスの対象外の目的に使われるとは限りません。

(謝辞:このページは Dalai Felinto のブログ記事の内容を基にしています)

Page status ([reviewing guidelines](#))

Text This sub-section needs some review & formatting...

Proposed fixes: none

- [Android 用ゲーム開発ガイド\(eng\)](#)
- [Android 用 Blender のビルド\(eng\)](#)
- [GLEES を使う Blender のビルド\(eng\)](#)