

Blender 中間学習

～ チュートリアルの後に見るガイド ～

初版：May 9, 2013

更新：March 7, 2014

著者：Zaslon

Ver. 1.4

目次

この文書の目的	3
第 1 章：環境カスタマイズ	4
User Preferences 画面について	4
Addon の有効化	5
Dynamic Spacebar Menu	6
Material Utils	6
System の設定	8
視点周りの設定	8
Auto Perspective の無効化	8
View の設定	9
第 2 章：操作の基本	10
ショートカットキーは覚えやすい	10
操作対象を指定するもの	10
動作内容を指定するもの	11
対応関係のわからないもの	12
操作や機能を検索する時に注意すること	12
第 3 章：再利用性を高めるための心構え	13
流用できるものは流用する	13
名前を付ける	14
操作はマニュアルに殆ど載っている	15
困ったら検索とは言うけれど	15

第 4 章：Blender のデータ構造を理解する..... 16

Scene データの構成と Outliner.....	17
Object と Data の関係.....	19
複製の注意点	23
Data Block の関連付けと Fake User	24
Data と Material の関係.....	25
Material Slot の関連付け対象 ~チェスの例を交えて~	26
Material Slot と Material の関係 ~オセロの例を交えて~.....	29

第 5 章：モデリングのアプローチ 32

便利な基本操作	32
Cursor Snap メニュー	33
Pivot Point の変更と 3D Manipulator	33
3D Manipulator と Orientation(操作軸)の操作	35
Snap について.....	38
プリミティブから変形していく方法.....	39
チェスのポーンを作る	39
断面図を作って面を貼っていく方法.....	45
あまり説明されていないが便利な機能	48
Shear(せん断変形).....	48
Slide Mode	49

おわりに 50

更新履歴	50
------------	----

この文書の目的

この文書の目的は、**Blender** の学習と製作の効率化である。学習の効率化は初心者にも有効で、製作の効率化は中級者の方が有効かもしれない。初めは分からなくても、「そういうものもあるのか」と頭のどこかに残しておくことで、後の理解を助けるはずだ。

細かい機能を網羅することならチュートリアルを見たほうが良い。しかし、目的の形状を作るための操作を教えるだけのチュートリアルではカバーできない、面倒くさくて地味な割に影響の大きい機能がある。そういう部分がデータの再利用性や製作効率に影響する。

自分はこういった細かいセッティングについては使いながら身に付けていったが、その経緯は覚えていない。徐々にセッティングを変えながら煮詰めていったため、過去のデータには無駄な部分が多くてどうしようもなかったり、他のモデルと組み合わせようとして無駄に複雑になっていたりする。そういう煩雑さに後から泣くのを防ぐには、はじめの段階から再利用性を意識してカスタマイズした方が良い。

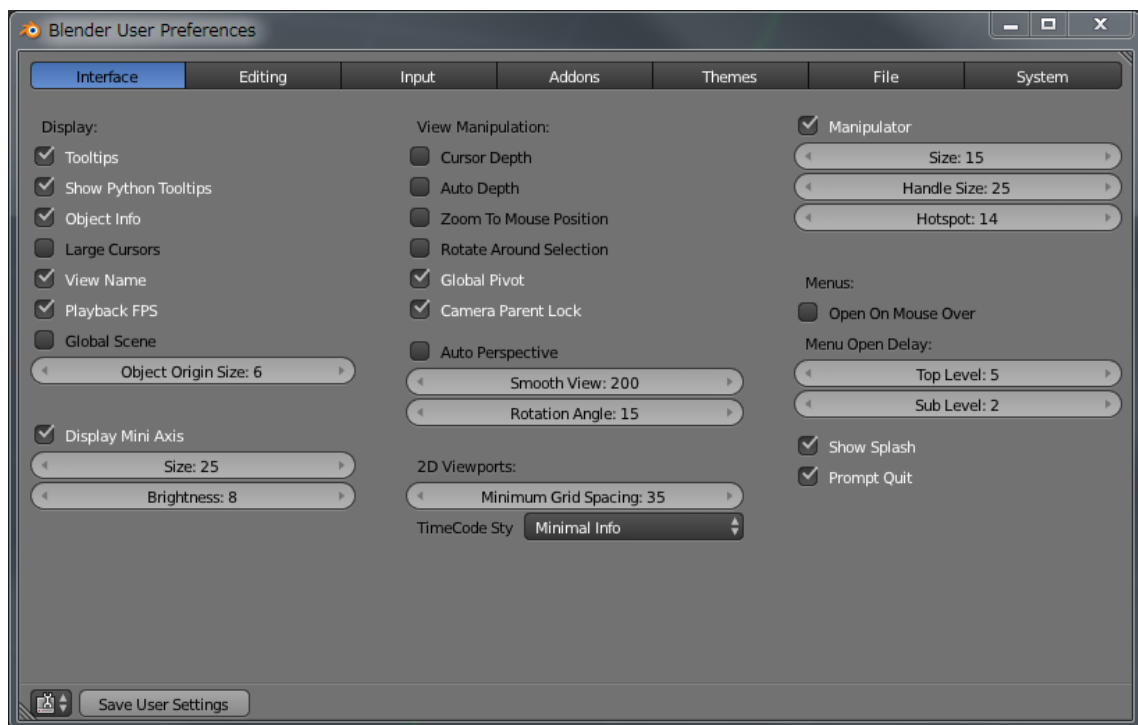
こういう機能は web、本ともに説明が極端に少ない。なので、自分が気をつけていることを書いてみた。学習と製作の助けになれば幸いである。

第 1 章：環境カスタマイズ

環境カスタマイズは操作のスピードを上げたり、あとから使いにくいファイルを生成したりしないために必要である。特に、新しいバージョンをインストールした後、前バージョンのセッティングを引き継がなかった場合や、新しい機能が増えた場合は見直すことになる。

User Preferences 画面について

- User Preferences の表示 `:Ctrl+Alt+U`
- 現在の設定を Startup file として保存 `:Ctrl+U`
 - Use Preferences 内の Save User Settings ボタンでも設定できる。



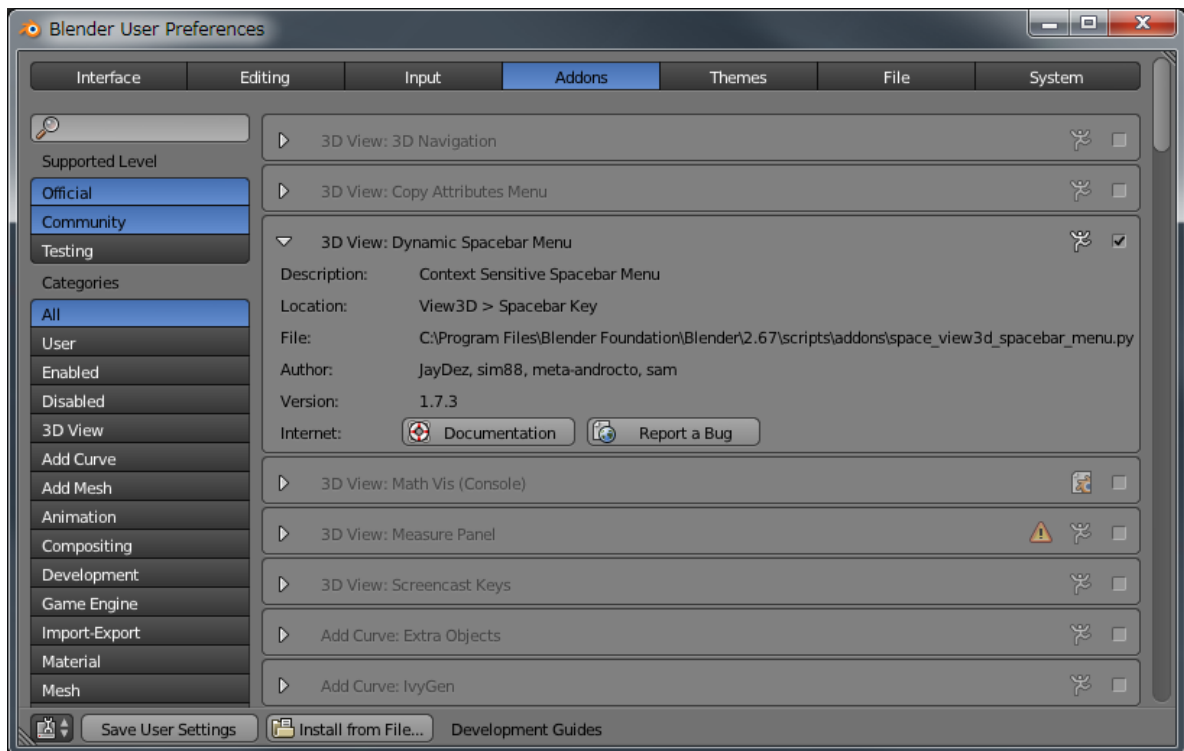
User Preferences 画面

大抵の設定はこの User Preferences 画面からアクセスできる。初期設定として重要なのは Interface, Addons, System くらいだ。気に入った設定を作ったら、Startup file として保存しておくといよい。Startup File とは、起動したときにその状態のファイルが読み込まれる、初期状態ファイルである。この Startup File ウィンドウ幅や構成、有効化した Addon リストなども保存される。

言語設定は、初期状態では英語表示だが日本語へ変更も可能だ。しかし、英語のままにしている。英語の方が資料が多いし、後述するようにショートカットキーの割り当ては英単語の頭文字になっていることが多いからだ。それに、「見慣れない英語」も「見慣れない日本語」も覚える労力は大して変わらない。

Addon の有効化

Addon は、結構な数のものが無効化された状態で同梱されている。

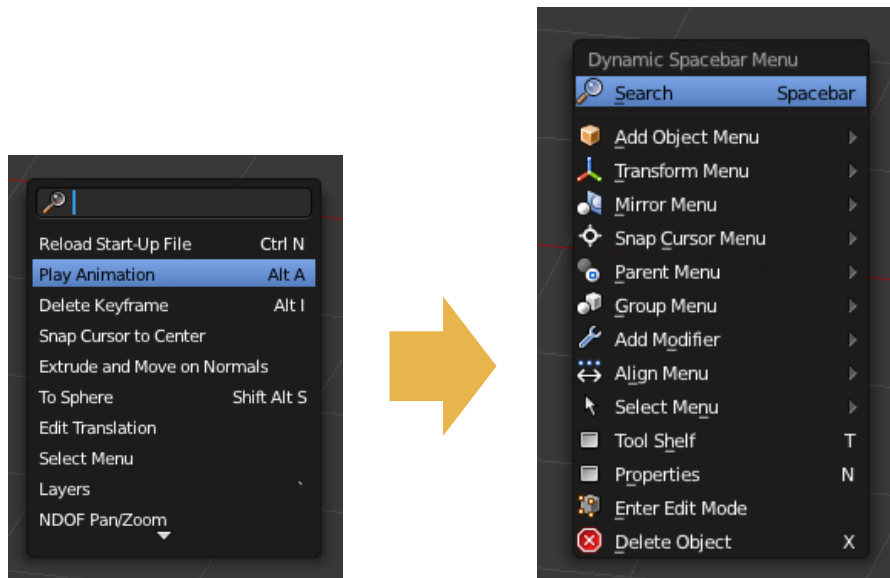


Addons 画面

いくつかの Addon は最初から有効化しておいた方がよい。特に、新たな要素や機能を追加せず、アドオンなしでも提供されている機能へのアクセス性を向上するアドオンを導入すべきである。その他のアドオンは、作るモデルに合わせて必要に応じて有効化した方がよい。こうしておくと、アドオンの提供が終了しても操作を覚えなおすだけで済み、データに影響がないので都合が良いのだ。アドオンによって追加される要素を用いたモデルが、アドオンなしでは動かなくなってしまうと不都合だし、完成したモデルをあとから調整するのは正直苦痛なのでやらないに越したことはない。

Dynamic Spacebar Menu

スペースを押したときに、通常は検索ウィンドウが現れるところを、様々なメニューの起点へと変更してくれる。その上、元のスペースバーの状態にもアクセスできるため、失う機能は何もない。



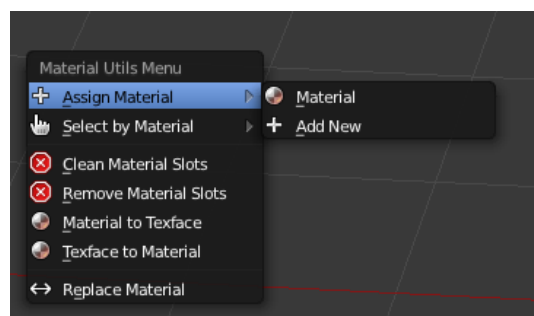
通常の Spacebar Menu

Dynamic Spacebar Menu

Material Utils

- Material Utils Menu :

というコマンドが追加され、下図のメニューが現れるようになる。



Material Utils Menu 画面

このメニューから、複数オブジェクトの材質を一括で変更したり、同じ材質のオブジェクトを纏めて選択したりできるようになる。オブジェクト数が増え

てくるといちいち変えるのは面倒だし、どこに使われているのかよくわからないマテリアルが増えてきたときに、調査するのに使える。

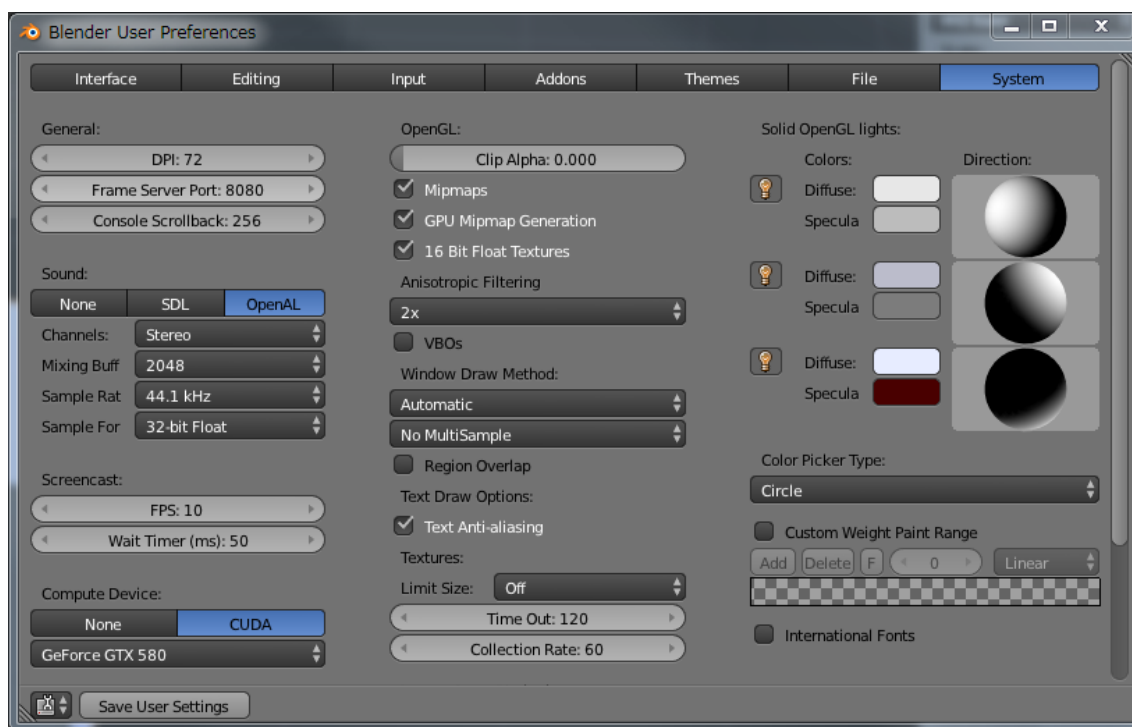


マテリアルパネルにゴミが残っている例

Blender では、使用オブジェクト数が 0 にならないとマテリアルが削除されない(これは第 4 章で解説している)。なので、ちゃんとマテリアルに名前を付けていかないと、上図のようなゴミ山状態になって対処できなくなってしまう。こういう時に Material Utils Menu の Select by Material を使えば、どこに使われているかが分かる。さらに、Replace Material を使えば Material を簡単に置換できるので便利である。

また、初心者向きではないメリットとして、昔作ったデータの Blender Render のマテリアルから Cycles Render のマテリアルへ変更するときに重宝するというものもある。

System の設定



User Preferences の System 画面

先ほどの User Preferences 画面の一番右のタブからこの画面を表示できる。左下の Compute Device のところから、CUDA とかを選んでおくと、もしかするとレンダリングが早くなるかもしれないが、私には正直よくわからない。このあたりの対応状況はバージョンアップに従ってどんどん変わるので、一応こういう項目もあるということを覚えておくといいかも。

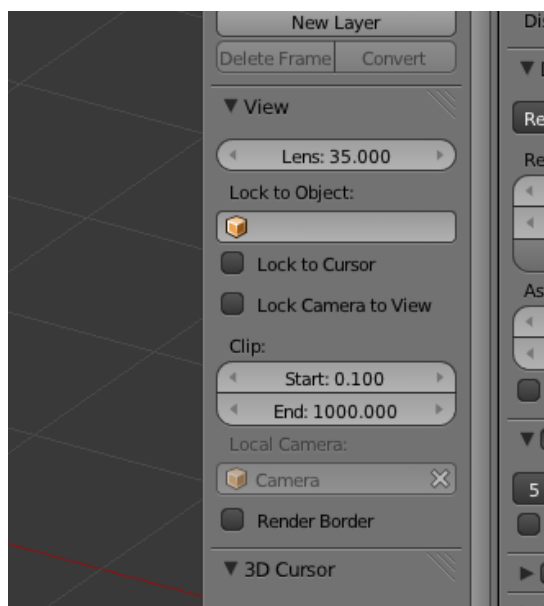
視点周りの設定

Auto Perspective の無効化

User Preferences 画面(p. 4 の画像参照)の中央あたりにある Auto Perspective は無効化しておいた方がよい。無効化した状態でも任意にパースを付けたまま画面回転は可能だし、有効化しておいても平行投影のままカメラを回せなくなるだけで、できることが減るだけだ。

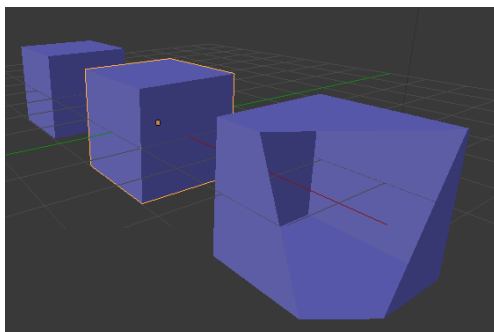
View の設定

View 設定は Properties shelf 内にある。Properties Shelf は 3D View 内で **N** を押すと表示される。Lens(焦点距離)は初期設定では 35 mm だが、遠近感が強すぎてモデリング中の確認には向かない。接写より望遠の方が歪みが少なくなるので、100 から 200 mm くらいにしておくともバランスが良い。

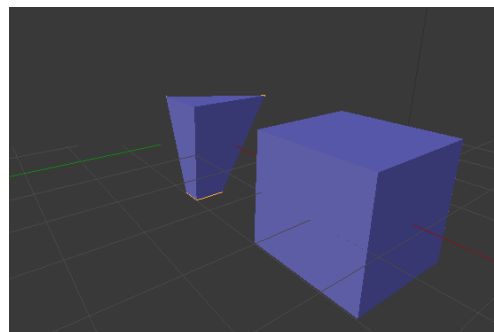


Properties Shelf 内の View 設定項目

また、同じく View 項目内にある Clip 項目も重要で、これはカメラからみて Start から End の距離までが描写されるようになっている。Start が大きすぎると手前側の面が切れて見え、End が小さすぎると遠くが見えなくなる。モデルが見えない時はここを見直すと良いかも。



Start が大きすぎる時



End が小さすぎる時

この画像はパース表示だが、平行投影でも容赦なく千切れるので注意。

第 2 章：操作の基本

ここで説明するのは基本操作というより、良く知らない機能がどの辺にありそうかを推測するための方法である。いきなりすべての操作を網羅しようとするから訳が分からなくなるのであって、基本はそんなに難しくない。

具体的なショートカットキー自体は web にチュートリアルやショートカットキー一覧があるので、それを印刷しておけばいい。そのうち覚えるはずだ。

ショートカットキーは覚えやすい

Blender がとっつきにくいと言われるのは、膨大な量のショートカットキーにあるのだろうか。とはいえ、Blender のショートカットキーが特段分かりにくい訳ではない。3DCG ソフトなら大抵こんなもんだろう。Blender は 3DCG ソフトの中だとわかりやすい方だと思う。

それでも分かりにくいと言われるのは、オープンソースなので、初めて触れる 3DCG ソフトになりやすい故の不幸だろう。自分は Shade で始めたので思うのだが、有料ソフトだと多少わかり難くても使って元を取ろうと言う気が起きるから、相対的に Blender は不利だ。

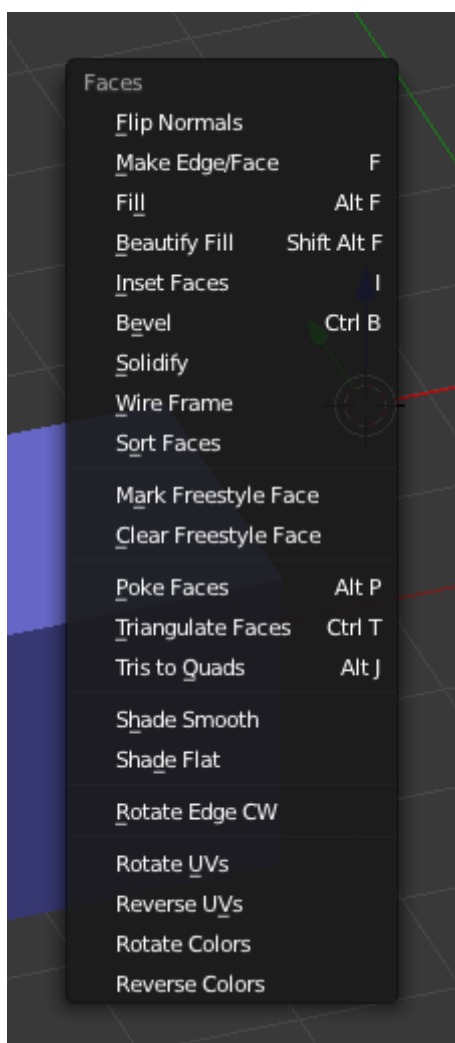
そういう訳で、ここでは Blender のショートカットキーそのものではなく、ショートカットキー割り当ての傾向について説明しようと思う。これを見れば、Blender のショートカットキーが無茶苦茶な設計をされているわけではないことが分かるはずだ。

操作対象を指定するもの

頂点(Vertex), 辺(Edge), 面(Face)の頭文字である **V**, **E**, **F** の 3 キー, 座標軸である **X**, **Y**, **Z** は操作対象の基本として度々登場する。

- 頂点/辺/面編集メニュー : **Ctrl**+**V**/**E**/**F**
- 特殊操作メニュー : **Ctrl**+**W**

メッシュに対する大抵の操作はこの 4 つのショートカットキーからアクセスできる。あとはたくさん出てくる機能の名前を調べればメッシュに対する殆どの操作はできるようになる。



Ctrl+F の面編集メニュー画面

ご丁寧に、ショートカットキーの割り当てがあるものはショートカットキーが書いてある(Fill は **Alt+F** といった具合に)。慣れない間はこの **Ctrl** メニューからアクセスして、徐々に覚えていけばいい。

動作内容を指定するもの

先ほどの操作対象指定型のショートカットキーよりも、こちらの動作の頭文字からとられたショートカットキーの方が多い。前の画像を見ても **Fill** の **F**, **Bevel** の **B**, **Poke** の **P** などが目に入るだろう。こういうタイプのショートカットキーは結構多い。

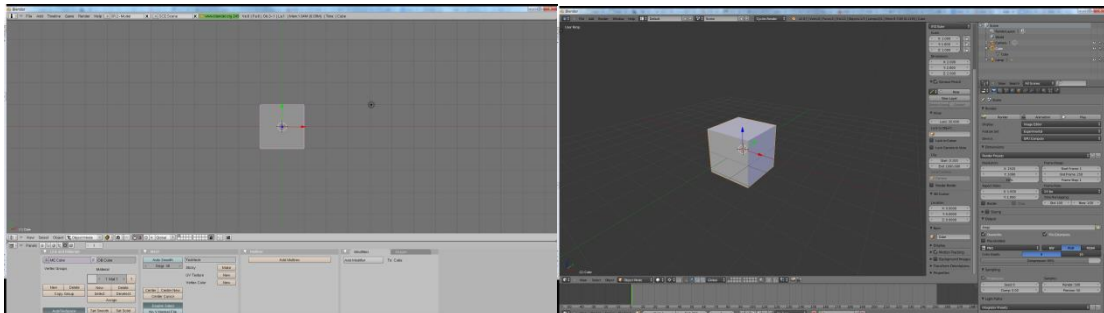
対応関係のわからないもの

じゃあ、**Tris to Quads** の **Alt+J** は何なんだ？ と思うかもしれない。これは多分、キーが足りなくなったから、空いていて押しやすいキーに操作を押し込んだだけだ。意味はないので、覚えるしかない。

とはいえ、こういうタイプのキーはそんなに多くないし、困ることもあまりない。こういう対応関係のわからないタイプの操作は、**操作の重要度や頻度が極端に低い**か、**極端に高い**かのどちらかだからだ。めったに使わないから変なキーでも気にならないか、ものすごく使うので、覚えやすさよりも押しやすさを重視したかのどちらかということだ。Be 動詞のような頻度の高い動詞ほど不規則変化になるのと同じだ(わかりにくいかな…)。

操作や機能を検索する時に注意すること

Blender は 2.5 以降とそれ以前で UI がかなり変わっていて、**2.5 以降を使う時に 2.4 以前の情報は役に立たないことがある**。特に、操作体系やパネルやタブの場所は全然違うので参考にならない。検索する時に「Blender face insert」とかで検索するよりは、「Blender 2.6 face insert」みたいにバージョンを記載したほうがヒットしやすくなる。



Blender 2.49

Blender 2.67

2.49 までは下部にあったツールボックスが右側に来たのが大きな変更点で、そのほか機能へのショートカットキーもいろいろと変わった。別のソフトに乗り換えるくらいの気持ちで臨むことになったことを覚えている。

第 3 章：再利用性を高めるための心構え

第 2 章で基本的な設定はおわり．とはいえ，何のためにあいつた設定をしたのかわからなければ意味がない．というわけで，基本的な心構えを述べたい．面倒くさい割に，はじめのうちは全然ありがたみのない内容なので，モデルが満足に作れるようになるまでは考えすぎなくても良い．自分の作りたいものを作る段階で見直してみしてほしい．

流用できるものは流用する

Blender に限ったことではないが，3DCG の便利なところは，データが流用しやすいことにある．再利用性をはじめから高めておけば，データが増えてきたときに圧倒的に楽になる．



流用の例

上の図は自分が製作したモデル 3 つだが，

- (1) キャタピラやハッチ，機関銃の **Object**
- (2) 迷彩やゴムの **Texture**
- (3) ライティングや背景などの **Scene**

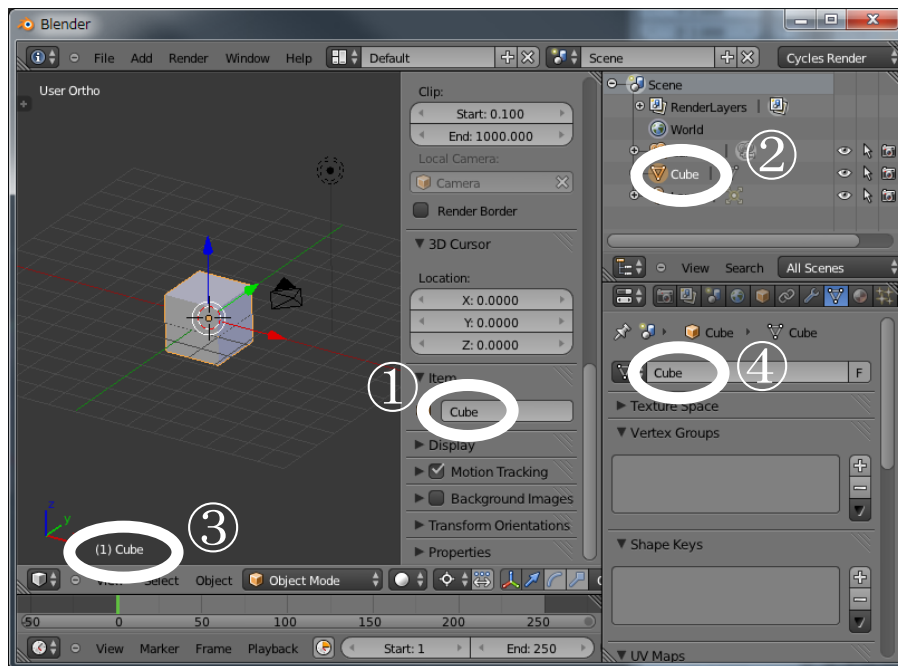
を流用している．オブジェクトやテクスチャを流用するのは使いたての頃でも思いつくものだが，シーンの流用は最近始めたことである．思っているより流用できる要素は多いのだ．

しかし、そのためには普段から気を付けておくべきことがいくつかある。ここでは操作というより、どういうことに気を付けて作れば後々操作に慣れてからが楽かを考えてみよう。

名前を付ける

Blender を始めたころには気にならなかったが、Object 名が Cube.o83g のように連なるようになって、流石に名前を付けようと思った。あとから名前を変えて回るのは骨が折れるので、新たな Object を作成するときに適宜名前を付ける方が良い。

しかしそこで困ったことが起きた。初期画面の Cube を見てみると、



Cube ひとつだけの初期画面

Cube というモノは画面の 4 ケ所もある。結論から言うと、①のところの名前を変えれば良いと覚えておけば取り敢えず問題ない。Object の複製を常に **Shift+D** でおこなっているうちはこれで事足りる(殆どの場合これで問題ない)。また、②の Outliner 上でも、**Ctrl+右クリック** によって名前を変更することができる。

しかし、リベットやナットのように同じような形のものを大量に複製したり、ひとつの Mesh を修正したら他の Mesh も修正された方が都合がいい時などは、

データ構造をある程度理解していた方が便利だ。これについては第 5 章で説明する。

操作はマニュアルに殆ど載っている

- Blender Wiki : <http://wiki.blender.org/index.php>

実はこの Learn の部分を見れば、大抵のことは載っている。むしろ情報量が多すぎてどう探していいか迷うレベルなので、普通に Google なんかで「Blender camera setting」とかで検索したほうが早いかもしれない。やりたいことや機能がはっきりしているならここから探す方がはやい。

困ったら検索とは言うけれど…

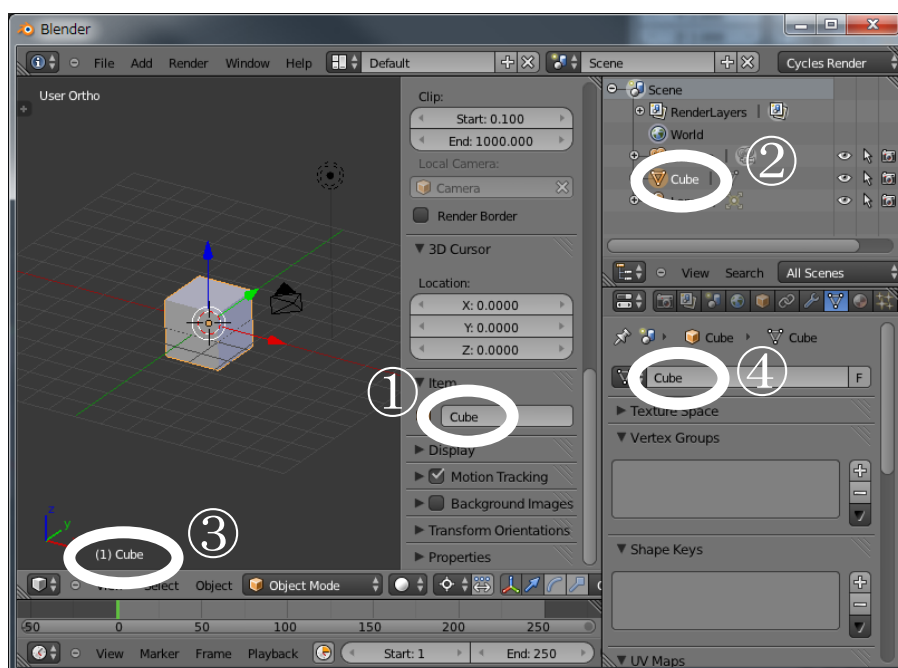
マニュアルに限らず、機能の名称や適応対象(キャタピラであるとか、人体の関節であるとか)が分かれば、Blender のチュートリアルや技術資料を探すのは容易だ。問題は、その名称や機能の存在すらわからない時である。一旦機能を見逃してしまうと、後から検索するのは難しかったりする。その機能が存在することは、前のバージョンからの更新が無い限り触れられなくなってしまうからだ。なので、リリースノートをこまめに確認することをお勧めする。Blender がバージョンアップされるたびに、どんな機能が追加されたかなどをまとめたリリースノートが公開される。リリースノートを確認すれば、とりあえずどういう方向の機能が搭載されたのか、どういう名称の機能があるのか等を押さえておくことができる。詳細は必要になった時に検索すれば良いのだ。

第 4 章：Blender のデータ構造を理解する

この章は、データ構造について説明している。始めたばかりの初心者はあまり意識する必要はないだろうが、チュートリアルで指示された操作の意味が良くわからなくなった時など、実際に内部で何がどう処理されているのかを知りたくなったら必要になる知識だろう。

Blender のデータ構造についての理解は、正直初心者には難しいかもしれない。これは使いづらいと言うよりは、現状の仕様が良くない。後から細かく説明するが、入れ子構造になっている Object と Data に全く同じ名前の物が生成されてしまうことが無用な混乱を生み出している。

具体的な例を見ながら、データ構造についてみていこう。これは前章でも出した図である。



Cube ひとつだけの初期画面

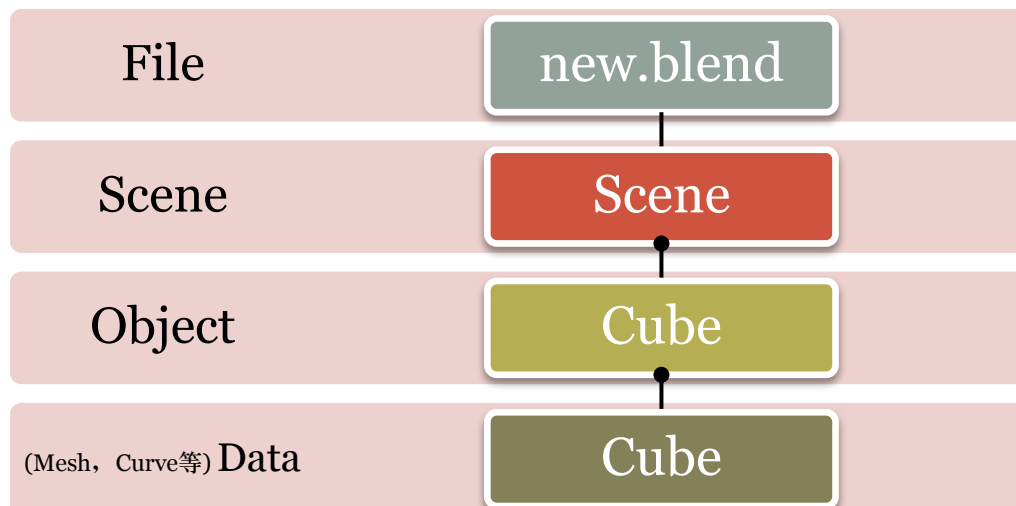
実は②の Cube の横の⊕を押すと、画面上の Cube が増える！ どうなってんだ！



Cube の中の Cube

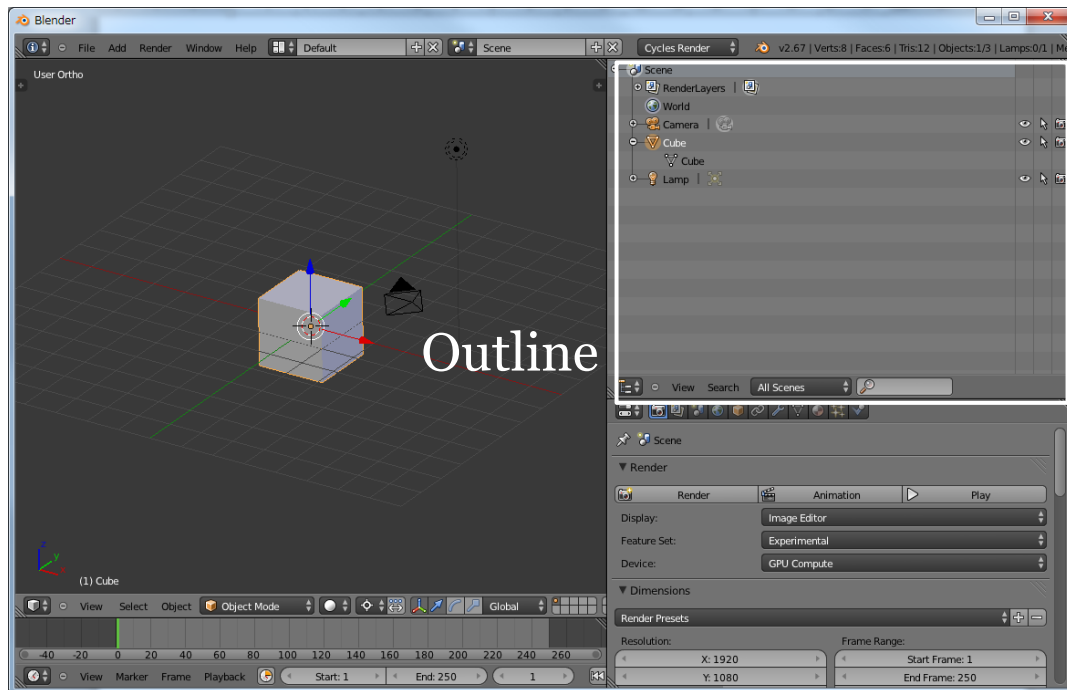
ここでの、①②④は **Object** としての **Cube** を指していて、③と⑤は **Mesh Data** としての **Cube** を指している。同じ名前だが全く別の物に対応しているのだ。これが分かり難さの諸悪の根源だろう(もちろん、元から分かりにくい要素ではある)。

Scene データの構成と Outliner



Blender のデータ構造

上図は初期設定を読み込んだ時の状態を模式図として表したものだ。先ほど述べた同名の別物というのは、**Object** としての **Cube** と、**Data** における **Cube** のことを言っている。名前は同じだがこの 2 つは別物だ！



Outliner

データ構造は Outliner で確認できる。表示を All Scenes にすると、上の図のデータ構造がそのまま表示される。他の画面でよく使うのは Groups だろうか？グループ機能の使い方は検索すれば十分な資料が見つかるので、割愛する。

Outliner 上では

- 名前変更 : **Ctrl** + **右クリック**

という操作が有効なので、覚えておくと作業が捗る。

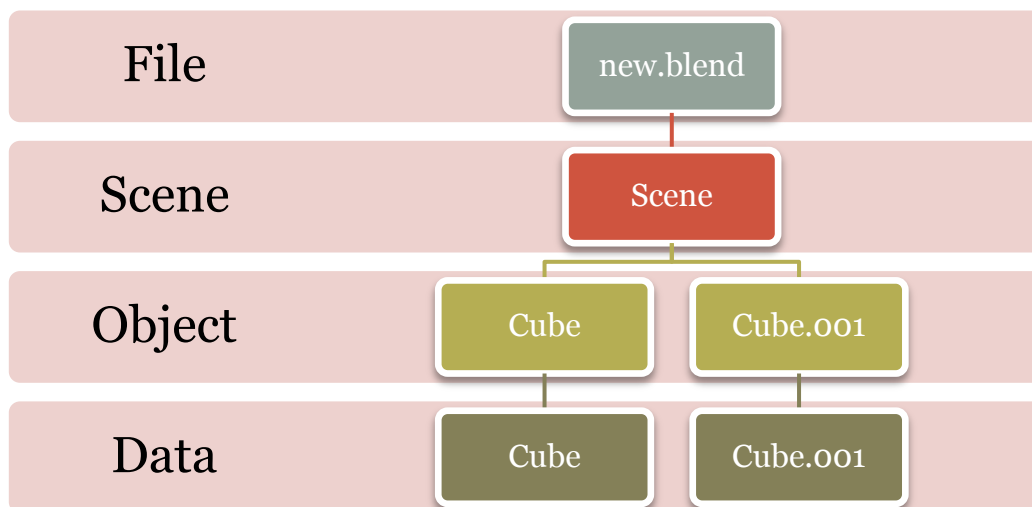
さて、とりあえず Material の説明は後回しにして、Object と Data の説明に移ろう。

Object と Data の関係

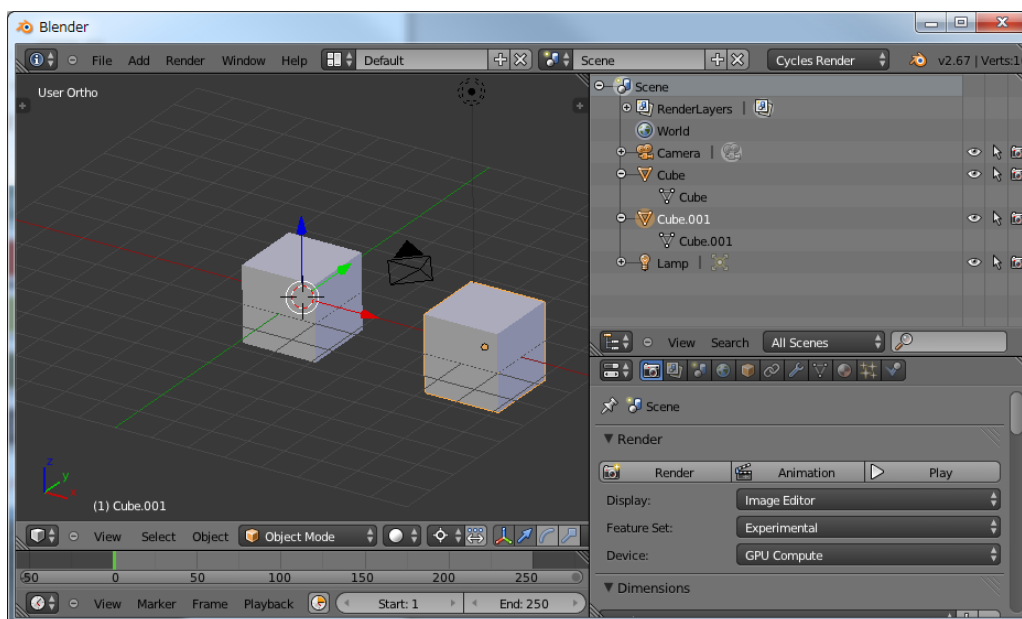
データ構造への理解を深めるために、

- Duplicate(複製) : **Shift+D**
- Data を共有し, Object のみ複製 : **Alt+D**

というコマンドをそれぞれ試してみよう. まず, **Shift+D** の複製を行うと, データ構造は下図の様に变化する.



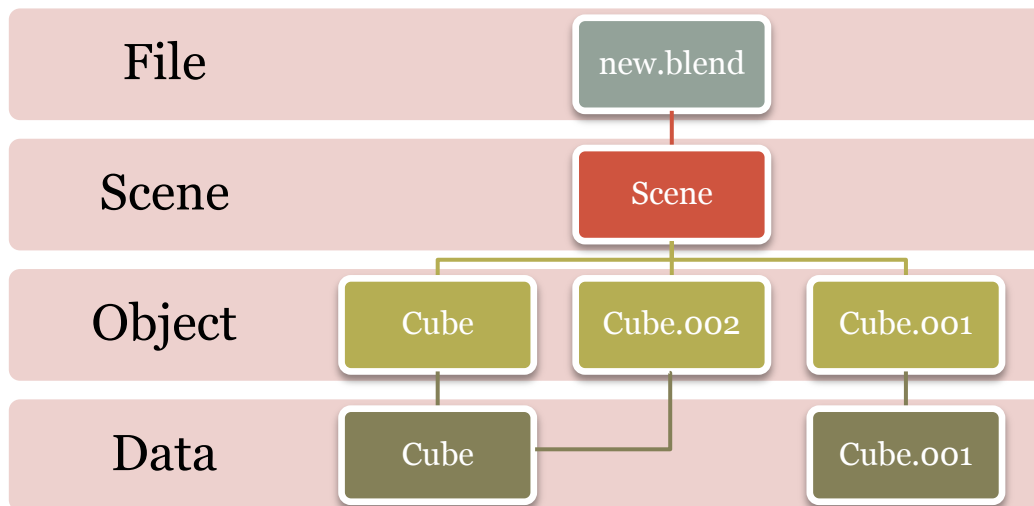
Cube 複製後のデータ構造



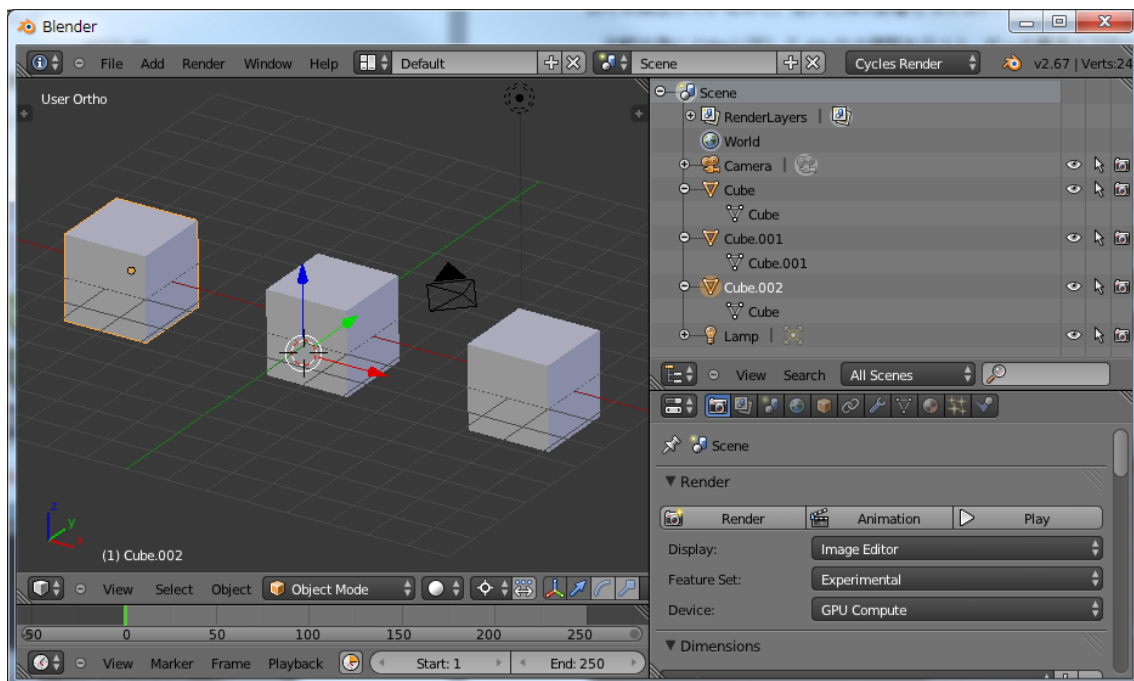
Cube.001 複製後の画面

Outliner 上でも、同様の変化が確認できる。この 2 つの Object と Data はそれぞれ独立しているので、互いに何の影響も与えない。

比較の為に Cube に対して **Alt+D** の複製を行うと、データ構造は下図のように変化する。

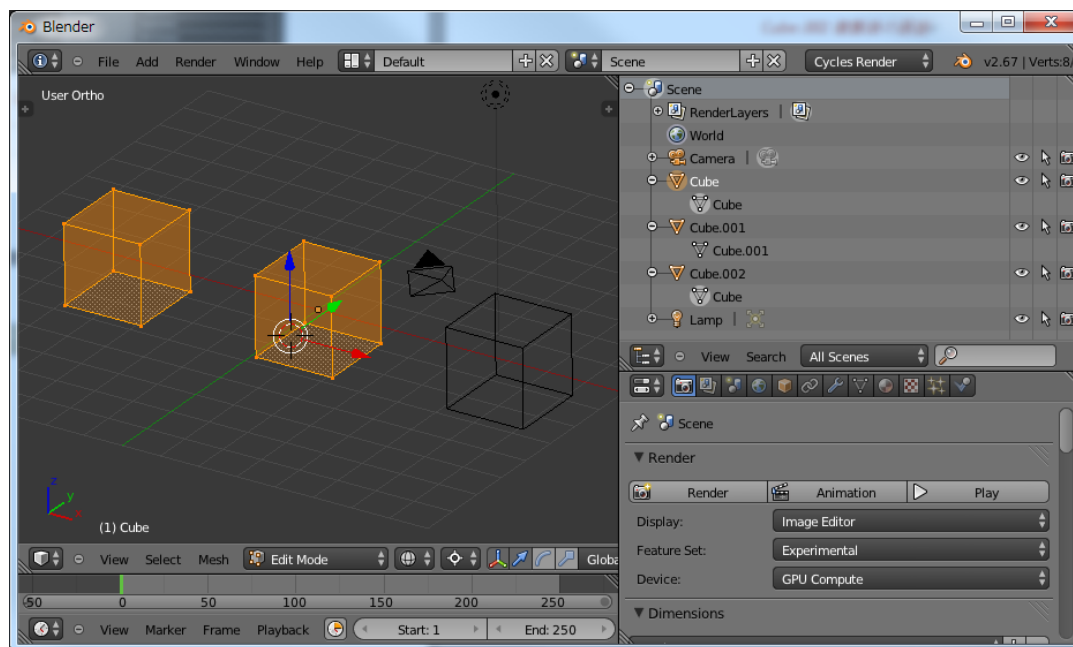


Cube.002 複製後のデータ構造



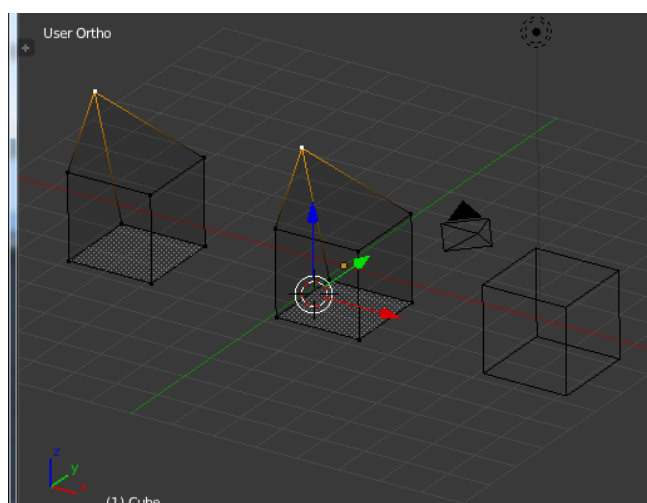
Cube.002 複製後の画面

今度は、Object のみが複製されて Data は複製されていない。[Tab] キーで Edit Mode にしてみると、違いが顕著に分かる。



Cube の Edit Mode

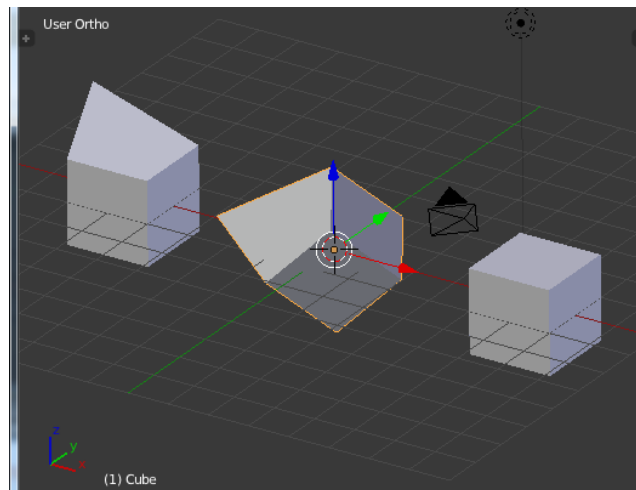
Cube の Edit Mode なのに、Cube.002 も同時に Edit Mode に入る。これは、Cube と Cube.002 が Data を共有しているからだ。頂点の位置であるとか、面の貼り方のような形状データは **Data** に保存されているのだ。Object はそのコンテナを務めているに過ぎない。自分で弄ってみると良くわかるが、この状態でメッシュを弄ると Cube と Cube.002 の双方に影響が及ぶ。



Cube のメッシュを弄った後

Cube.001 だけは Data を共有していないので、影響がない。当然 Cube.001 のメッシュを弄ると、他には影響が及ばない。

ここで「じゃあ Object って何のためにあるの？」と思うだろう。それは Object Mode で移動や変形をすると良くわかる。



Cube を Object Mode で回転させた後

この時、Cube.002 は回転しない。回転しているのは **Data の Cube** ではなく、**Object の Cube** だからだ。Object Mode という名称は、Object を対象にする操作をするから、そういう名前になっているのだ。

つまり、Blender では、メッシュ等の形状データは Data に格納されており、Object は物体ごとのローカル座標系を指定する、という分担になっている。Object によって指定されたローカル空間に、Data が読み込まれるのだ。どのくらい全体座標からずらすかを Object が指定している、と言っても良いかもしれない。

この辺りは言葉では説明し辛いですが、いくつか **Shift+D** と **Alt+D** で複製したモデルを弄っていると感覚が掴めてくるので、是非やってみてほしい。このあたりのことを理解すると、Modifier や Object Constraint が何に対してどう作用しているのかが掴みやすくなり、学習も製作も捗るだろう。

複製の注意点

Modifier の Array や Mirror を使えば大抵の同形状複製はできるので、**Alt+D** を使う機会はそんなにないかもしれない。自分は方向が異なるが形状は全く同じ部品が必要になった時、Array と組み合わせて使うことが多い。もっとも、この作品もかなり反省するところはあるのだが…



複数の複製方法

Alt+D で複製する機能は使い慣れると楽に形状を作れるが、絶対に同じ形状の部品であり、かつ、単純形状である場合に限って使った方が良い。一定以上複雑な形状の組み合わせなら Group の方が便利だし、一カ所不都合があっても誤魔化せない。まったく同形状であることを強いられるので、融通が利かないのだ。

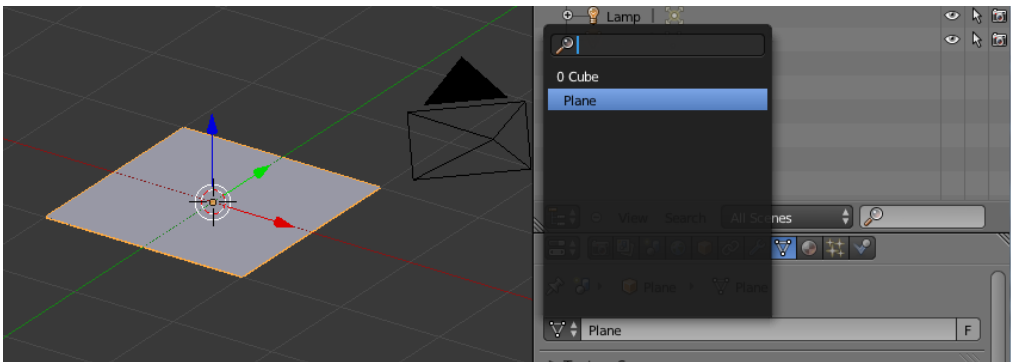
また、Data の共有はぱっと見で分かりにくいので、データの見通しを悪くしてしまい、管理が面倒になりがちだ。

Data Block の関連付けと Fake User

Data と Material の関係を説明する前に、説明しておくべきことがある。今までの説明では、Object にたいして Data を「関連づける」とか「接続する」といった表現を用いてきた。Outliner では階層構造が表示されているのだから、親子関係ではないのか？ と思った人もいるだろう。Blender では、すべての Data Block は(Parent 設定をしない限りは)親子関係ではなく、独立した Block として扱われている。なので、階層構造に見えても、データに上下関係は基本的にない。

これを端的に示しているのが、前節で説明した Object と Data の関係だろう。Data はあくまで Object に関連付けられているだけで、Object からはデータとしては独立している。なので、複数 Object から単一の Data を参照するような、単純な階層構造ではありえない現象が起こり得る。

内部的には Material だろうが Mesh Data だろうが Object だろうが、同一の扱いを受けている。Material の項で少しふれたが、関連付けられた Data Block の数が 0 になると、ファイルを閉じる際に自動的にデータが削除される。これは全ての Data Block に対してなされる動作で、Mesh Data も関連付けられた Object が無くなると 0 マークが付き、削除される。



Cube Object を削除して、Plane を作った時の例

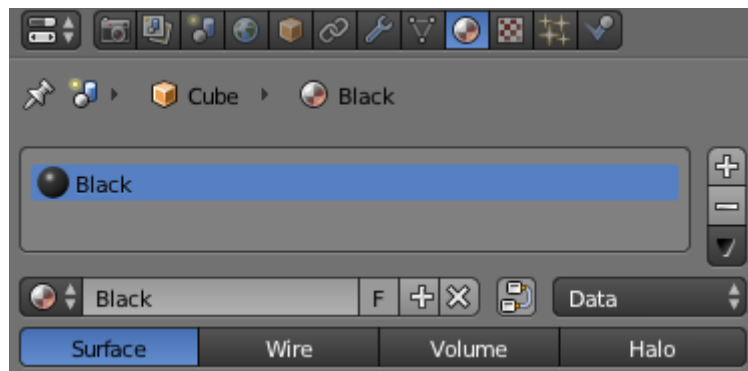
上の画像では、最初に用意されている Cube を ☒ で削除し、新たに Plane Object(と Mesh Data)を生成した後、Plane Object の mesh 選択欄を開いた画面だ。Cube の横に 0 マークがついているのがわかる。この状態でファイルを保存して再び開くと、Mesh Data の Cube は削除される。状況によっては、関連付け

る Object なしの Mesh Data をファイル内に残しておきたい場合があるだろう (Bone の Custom Shape を使う場合など)。そういう時は、選択欄の右端にある F のボタンを押す。F は **Fake User** を追加するボタンで、**User** 数が 0 になったときに自動でデータが削除されるのを防ぐため、架空の User を関連付けるためのボタンだ。

Object だけは義務的に Data を所持しなければならない(空の Object は作れない)ので、空の Object を作る用に Empty が用意されているのだ。

Data と Material の関係

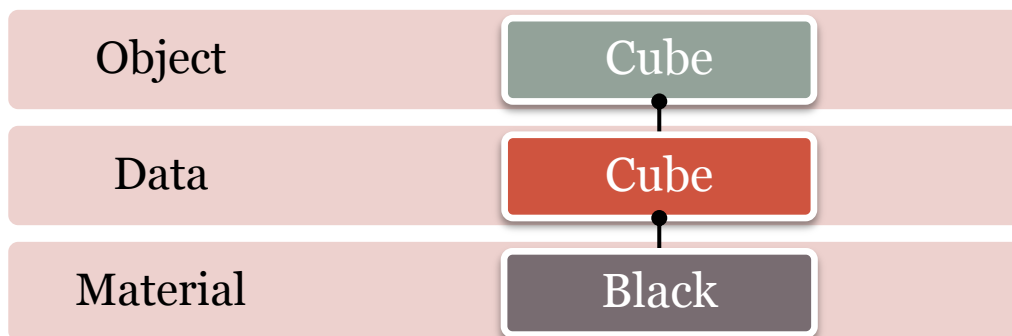
Material のパネルは、実は結構混乱しやすいインターフェースだ。プラスとかマイナスとか書いてあるボタンが、それぞれ何を増やして何を減らすのかが分かりにくい。



Material のパネル

Material Slot の右のボタンは、Slot を増減させるボタンだ。黄色い枠で囲まれているプラスとバツボタンは、Material Slot に装填する Material を選択したり、新規作成したり、削除したりするボタンだ。F のボタンは Fake User を追加するボタンで、前節で説明したものと効果は同じだ。**Mesh Data** も **Material** も、内部的には扱いは一緒なのだ。

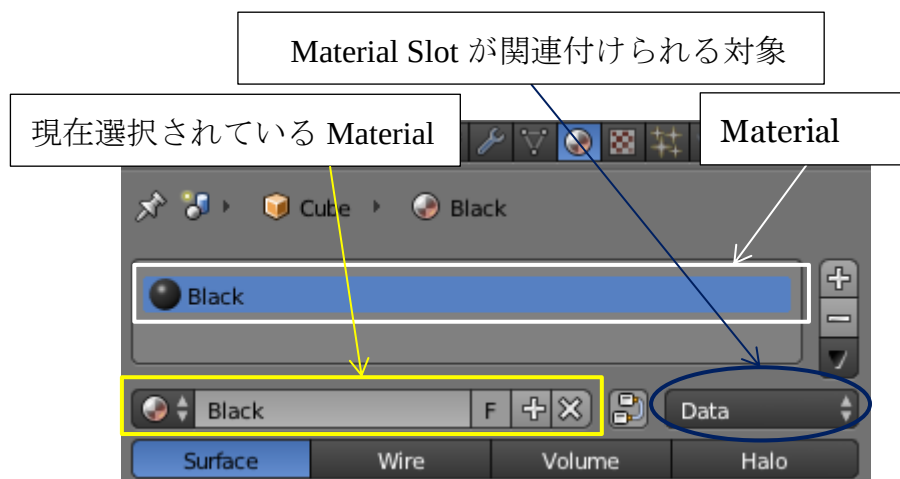
さて、上図の状況を図示してみよう。以下のようになる。



上の画像のデータ構造

Material パネルには「一番上の Material Slot は Cube の Mesh Data に関連付けられており，そこには Black という材質が選択されている」という状況を示している。

Material Slot の関連付け対象 ~チェスの例を交えて~

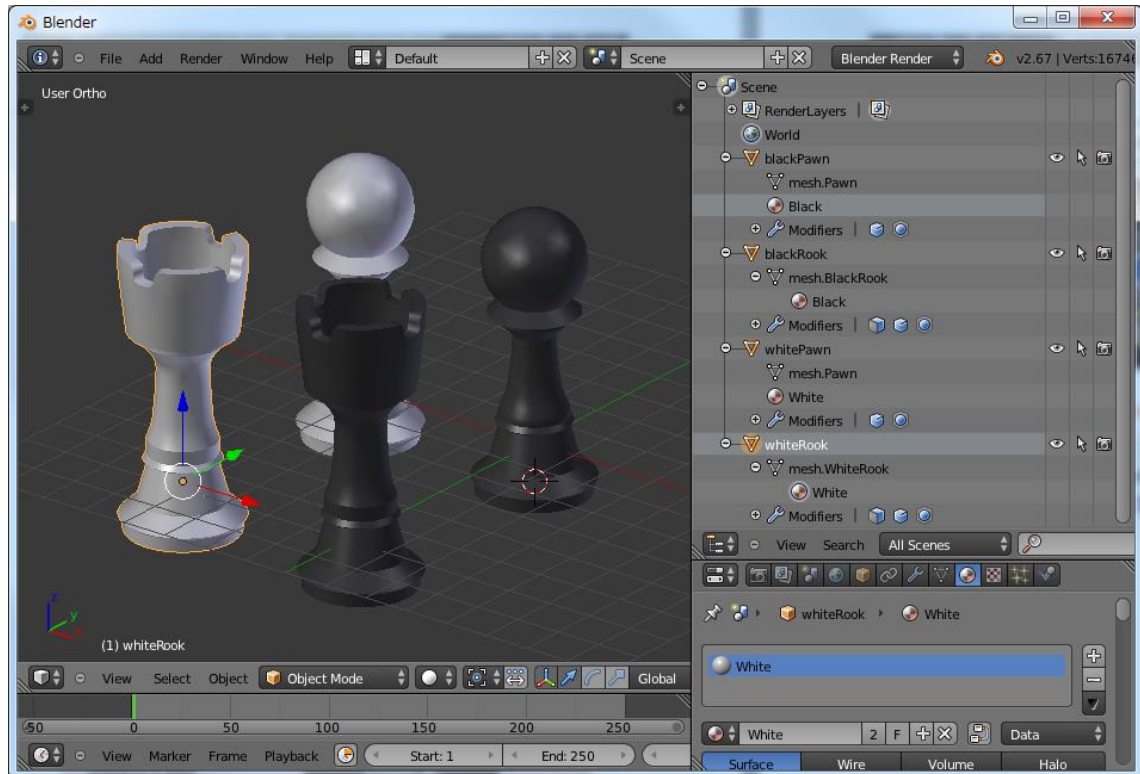


Material パネル周りの説明

先ほどの画面に説明を付けると，上図のようになる。

Material は通常 Data と結び付けられているが，これを Object に変えることも出来る。まったく同形状で別の色の物を作るときは，Object に変えれば良いということだ。チェスの駒はいい例だろう。例を作ってみた。読むだけでは意味が分かりづらいだろうが，Cube でもいいので，自分で Scene を作ってみると

分かりやすい。[Shift]+[D]と[Alt]+[D]で複製したモデルに対して、Material を設定して弄ってみれば、より理解が深まるはずだ。



ポーンとルーク

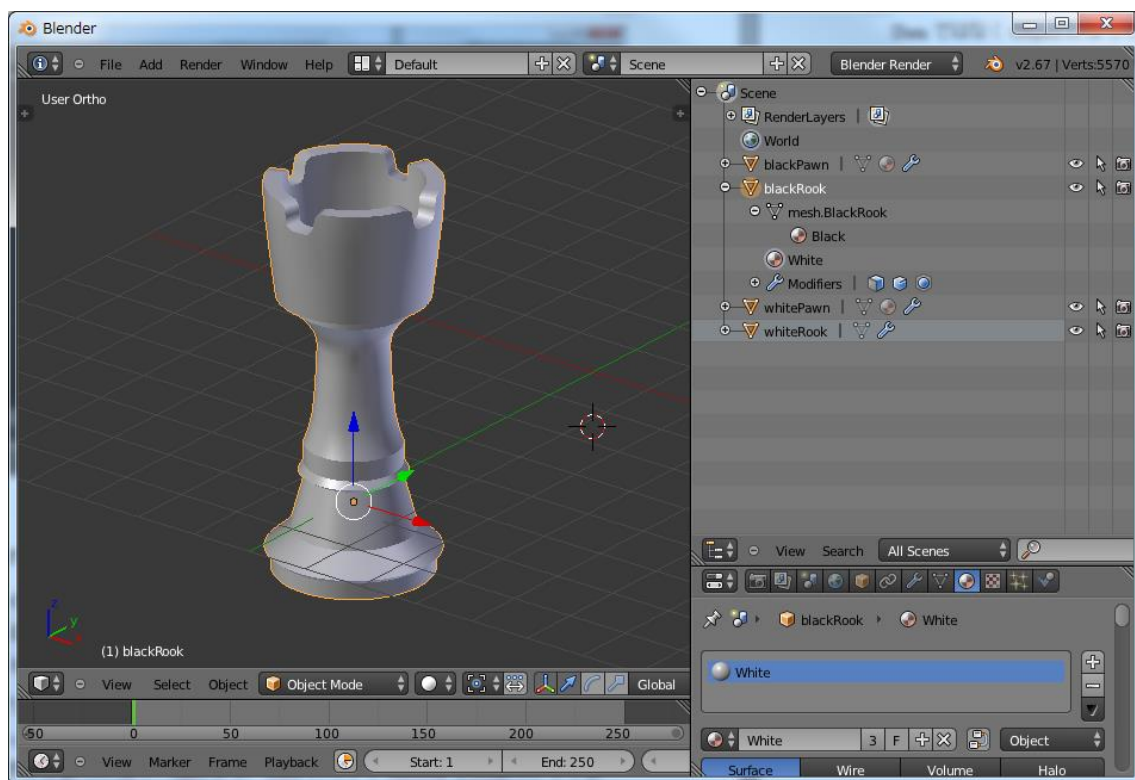
このポーンとルークの Scene は、

- (1) Object は 4 つある (blackPawn, whitePawn, blackRook, whiteRook)
- (2) Data は 3 つある (mesh.Pawn, mesh.blackRook, mesh.whiteRook)
- (3) Material は 2 つある (Black, White)

という構成になっている。データ構造は Outliner を見ればわかるが、以下のようになっている。この Scene では、ポーンは **Mesh Data** を共有している。なので、形状は自動的に全く一緒になる。Data に対して Material を関連付けてしまうと色に差がつけられなくなってしまうので、ポーンの Material はそれぞれの Object に接続されている。Outliner 上でも、mesh.~~の下ではなく、Object 直下に接続されているのが分かるだろう。

一方で、ルークの方は別々の **Mesh Data** を持っている。なので、黒と白のルークにはデータ上は何の関連性もない。そのため、ポーンの場合と異なり **Object** が **Material** に関連付けられている。補足しておく、別にポーンと同じように Object に関連付けても結果は同じだ。ただ、デフォルトで Material の接続対象が Data ではなく Object になっているので、特に必要が無ければ **Object** に接続しておいた方が良い。そのほうがデータの見通しも良くなる。あくまで Data は形状のデータのみを持つようにしておいた方が、扱いやすいのだ。

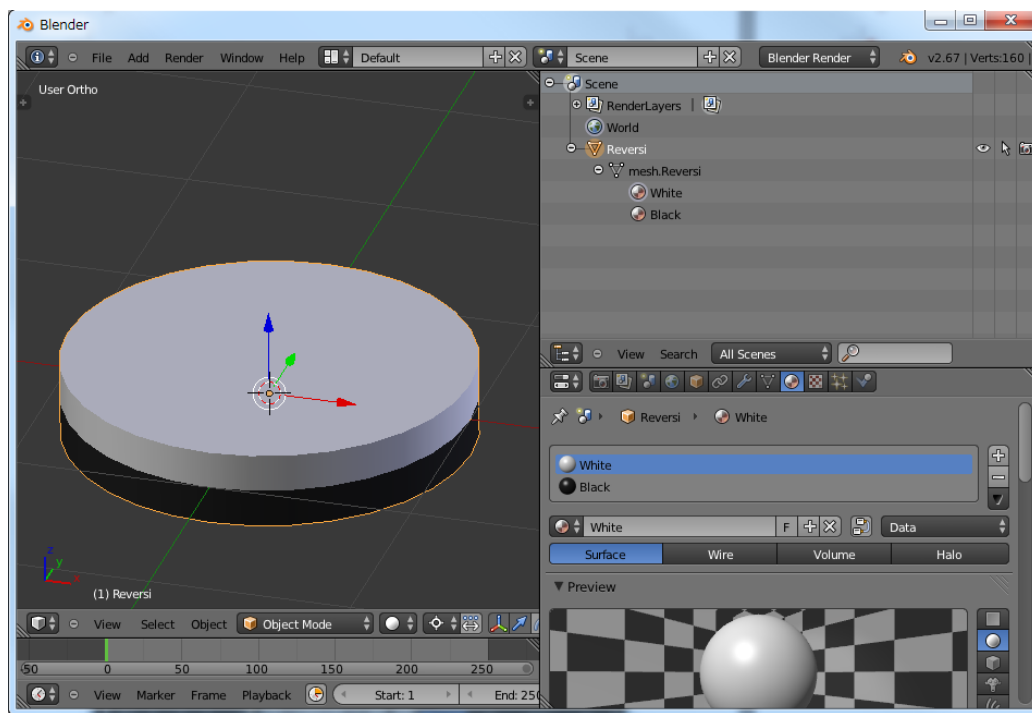
ちなみに、Data と Object 両方に Material Slot をひとつずつ設置して、それぞれに別の Material を設定すると、Object に対する設定が優先される。これは、Outliner 上で上階層に来る設定の方が後から適用されるためだ。この適用順序は、覚えておくと Modifier や Object Constraint の理解に役立つ。



BlackRook の Data に Black を、Object に White を設定した例

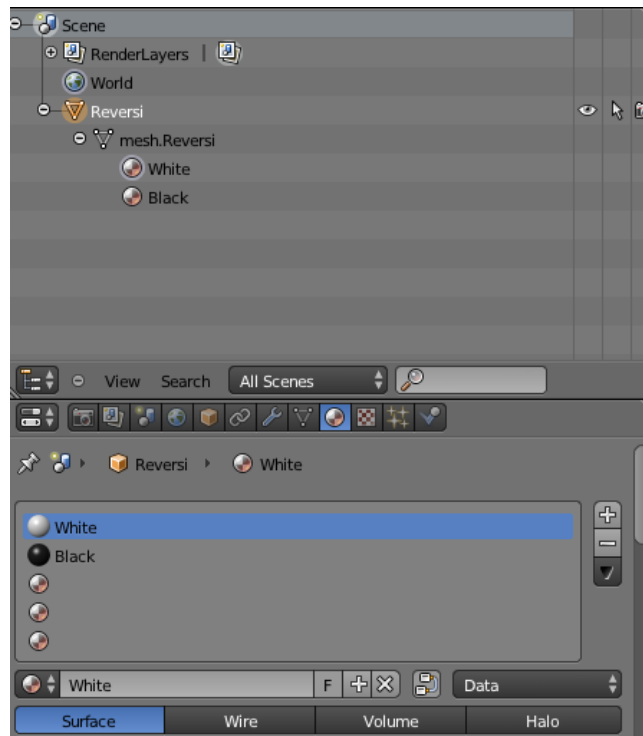
Material Slot と Material の関係 ~オセロの例を交えて~

今までは同じ形状で別の色が存在する Scene を例に、Material Slot の接続先について説明してきた。しかし、現実には Material Slot は複数用意することができる。その場合どうなるかを、オセロの駒を用意した Scene で説明しよう。



オセロの駒

今回は、White と Black の双方の Material は mesh.Reversi という Data に関連付けられている。

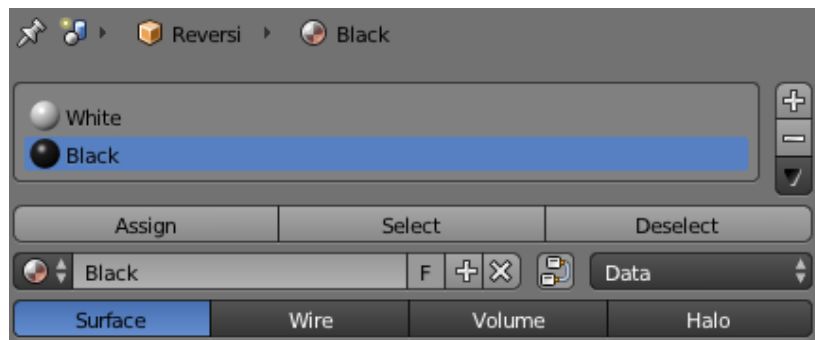


未装填の Material Slot を 3 つ作った例

Outliner には未装填の Material Slot は表示されないが、Material slot 自体はいくつでも作ることができる。新たに作成された Material Slot は

- (1) Material Slot がひとつもなかった場合、すべての面が割り当てられる
- (2) Material Slot がひとつでも存在した場合、割り当てなしで作成される

という動作をする。Material Slot を用意してから Edit Mode に入ると、下図のような画面が現れるのだが、そこには Assign(今の Material Slot に Face を割り当てる), Select(割り当てられている面を選択する), Deselect(割り当てられている面の選択を解除する)の 3 つのボタンしかなく、Material Slot への割り当てをなくすボタンはない。つまり、**Material Slot がひとつでも存在するなら、材質割り当てのない面は存在できないのだ。**



Edit Mode 中の Material パネル

覚えておいてほしいのは、**Material と Object(あるいは Data)**は、直接関連付けられているわけではないということだ。Material Slot というインターフェースを介して接続されている。なので、**Material Slot を新規作成しても、Material は生成されない**。一方で、Material を新規作成しても、関連付けられた Material Slot に Face が Assign されていなければ、Material はどこにも表れない。

実用上のことを言うと、あまりひとつの Object に Material Slot をたくさん設置すると、後々管理が面倒だ。とはいえ、Object が大量に表れるのもそれはそれで面倒なので、そのあたりは好き好きだろう。ただ、少なくとも **Data が複数 Object に関連付けられているときは、沢山の Material を関連付けると混乱するのでやめた方がよい**。

第 5 章：モデリングのアプローチ

ここでは、細かいモデリング操作については説明しない。そういう説明は、他のチュートリアルを参照してほしい。ここでは、モデリングの最初の段階をどうはじめて良いか分からない！ という場合のアプローチ方法をいくつか例を挙げてする。

どの方法をとるにせよ、実は最も時間をかけるべきは、写真をはじめとした参考資料をとにかく集めることだ。そして、開始前に完成形を軽くスケッチしておいた方が良い。そうしないと、どこを作りこんでいいのかわからなくなるし、作りこみやすい部分を 1 カ所作りこんでしまうと、他の作りこみにくい部分が情報不足に見えてしまったりする。こういうコンセプトの失敗は後からカバーしにくいのだ。

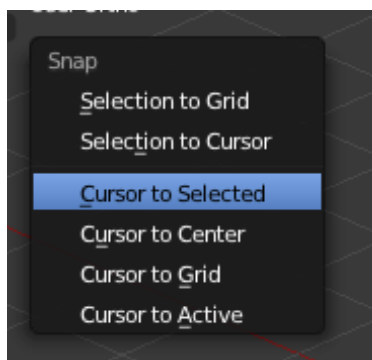
便利な基本操作

移動や回転などは、ひとつでもチュートリアルをやったことがある人ならわかると思うので割愛する。本当に良く使う割に説明がさらっと流されている機能をいくつか挙げておこう。

- Cursor Snap メニュー : `Shift+S`
- Pivot Point を 3D Cursor にする : `[(カンマキー)]`
- Pivot Point を Bounding Box Center にする : `[(ピリオドキー)]`
- Mesh Select Mode を変える : `Ctrl+Tab`
- Snap のオンオフ : `Shift+Tab`

これらは本当によく使う。どの機能も画面からアクセスできるが、これらのキーを覚えておいた方が良い。Mesh Select Mode の変更はやってみればすぐわかるので細かい説明は割愛するとして、残りのショートカットキーの動作を細かく説明していこう。

Cursor Snap メニュー

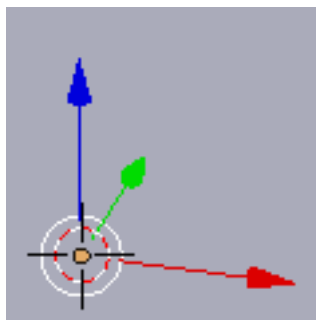


Snap メニュー

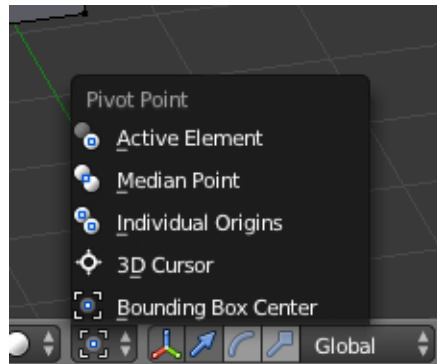
[Shift]+[S]で出せる Snap メニューは上図のようになっている。これは 3D Cursor を任意の点に移動させるコマンドだ。良く使うのは、Selection to Cursor と Cursor to Selected だろうか。この 2 つは「選択している頂点、辺、面をカーソルの位置に持ってくる」と「選択している頂点、辺、面の位置にカーソルを持っていく」という動作をする。**[S]**での拡大縮小の起点を決める使い方が多いだろう。

Pivot Point の変更と 3D Manipulator

Pivot Point というのは、操作の起点となる点だ。表示上は 3 色の矢印が表示されている起点の白丸だ。白丸の部分が **Pivot Point** で、矢印そのものは **3D Manipulator** という名前がついている。3D Manipulator については次項で説明する。



Pivot Point と 3D Manipulator

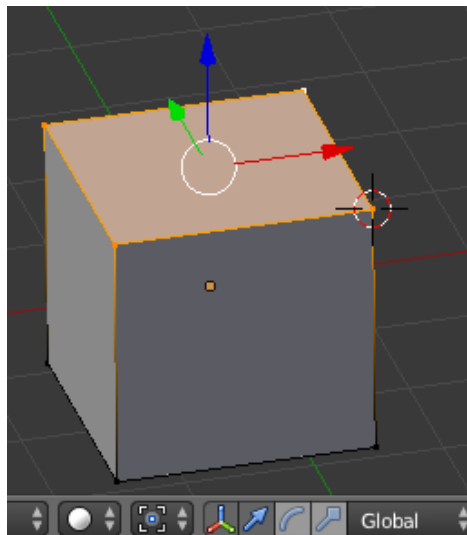


Pivot Point メニュー

これが Pivot Point メニューだ。画面下側のメニューからマウスで選択することが出来る。良く使うのはショートカットキーが割り当てられている 3D Cursor のモードと、Bounding Box Center の 2 種類だろう。

3D Cursor モードでは、文字通り 3D Cursor を起点に操作することができる。前項で説明した Snap 機能と組み合わせることで、頂点を集める位置を決めたりするのに使う。

一方、Bounding Box Center モードは選択している頂点のちょうど真ん中を自動的に操作の起点とするモードだ。下の画像で比較してみよう。



Bounding Box Center モードで4つの頂点を選択している状態

このとき、3D Cursor は右上手前側の頂点のところにあるが、操作の起点である **Pivot Point** は 4 つの頂点のちょうど中間点に来ている。これが **Bounding Box Center** モードの動作だ。

次節で説明しているチェスのポーン作成では Bounding Box Center モードを多用している。頂点を一周選択して拡大縮小するような操作では、いちいち 3D Cursor を Snap するよりも、このモードの方が素早く成形できる。

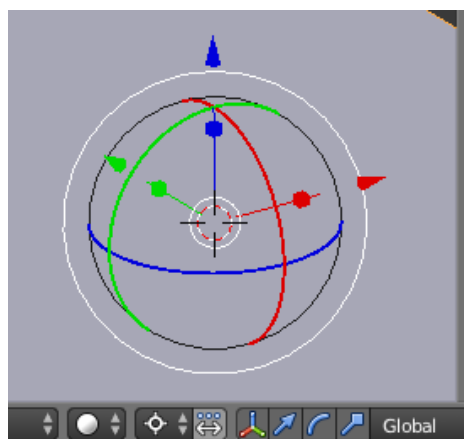
3D Manipulator と Orientation(操作軸)の操作

先ほどの Pivot Point メニューの隣に、3D Manipulator に関する設定がある。これは初期設定ではショートカットキーが無いが、たまに操作する部分だ。3D Manipulator の形状と機能は、3D Manipulator メニューで変更できる。



3D Manipulator メニュー

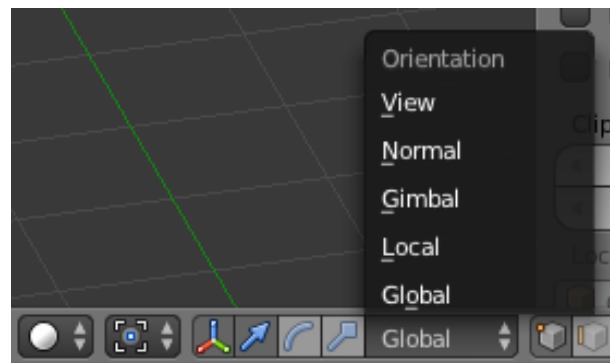
初期状態では 3D Manipulator の表示はオン、モードは Translate Manipulator になっている。Translate Manipulator は、矢印をドラッグすることで平行移動させる Manipulator だ。他の Manipulator には Rotate Manipulator, Scale Manipulator がある。これらの Manipulator は **[Shift]** を押しながら選択すると、複数選択ができる。



全ての Manipulator を表示した場合

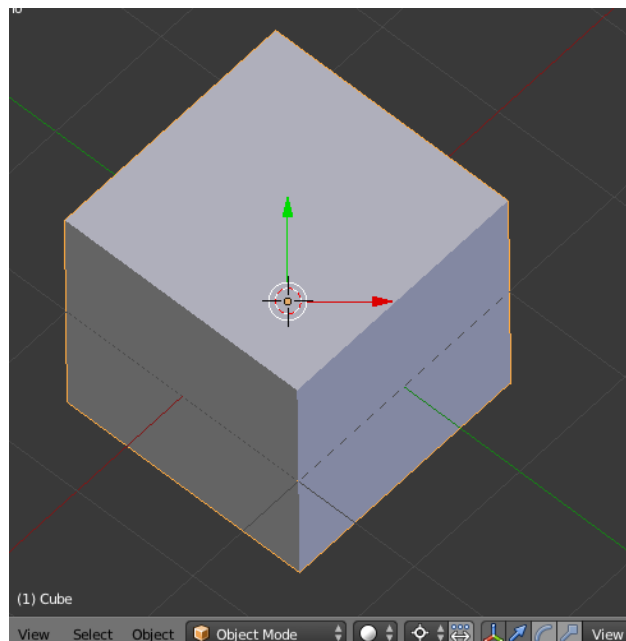
正直, Translate Manipulator 以外はあまり表示したことが無い. Armature によるポージングをするときに使う, というのは想定できるかもしれないが, 自分自身が動画作成に詳しくないので確証はない.

さて, 3D Manipulator メニューの隣には, Orientation メニューがある.



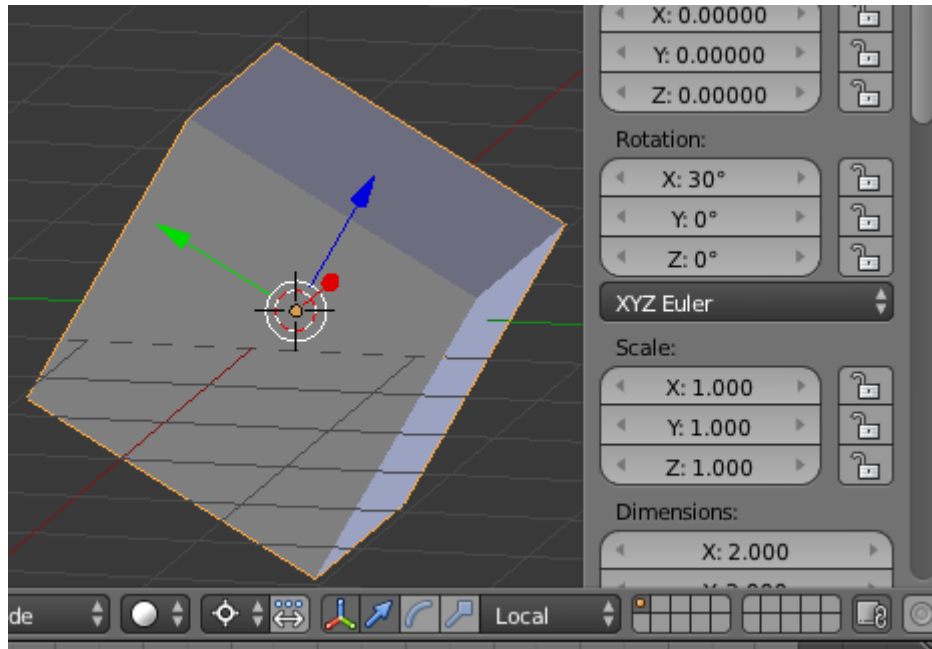
Orientation メニュー

このメニューは, 3D Manipulator がどんな軸に動くのかを規定することができる. 初期状態では Global が選択されており, World 座標系に一致した方向に動くようになっている. これを View にすると, カメラに対して平行・垂直方向へ動くようになる.



Orientation が View の場合

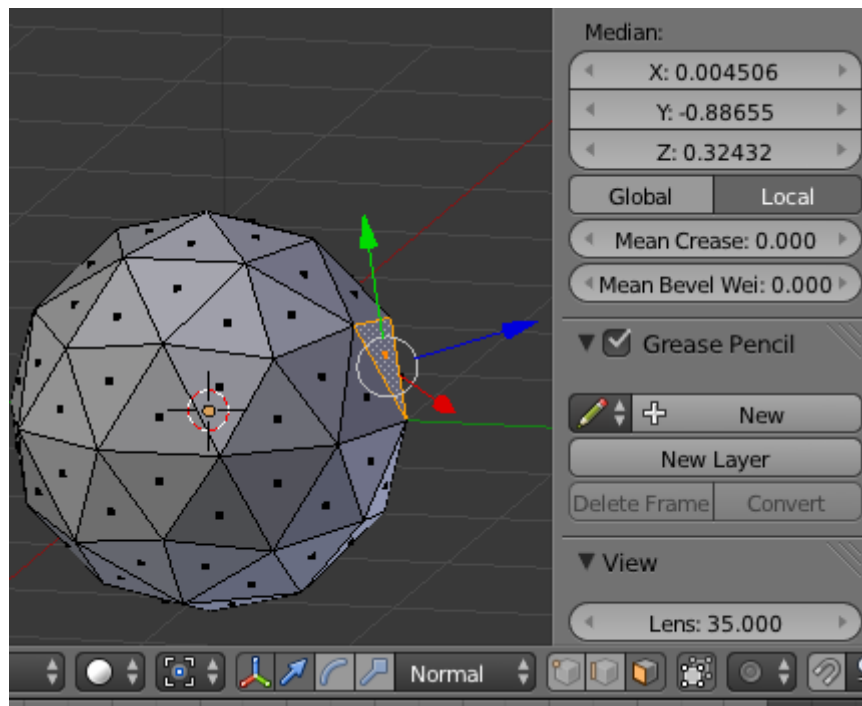
良く使うのは **Global**, **Normal** と **Local** のモードだろう。Local モードの場合は、Object の軸に対して平行・垂直にカーソルが動く。第 4 章で「Object は物体ごとのローカル座標系を指定する」といったのは、ここで生きてくるのだ。



Object の Rotation が 30 度のときの Local モード Manipulator

このモードでは、Object での Rotation に合わせて Manipulator の方向が変わる。上手く生かせば、傾いた物体の形状編集はとても楽になる。無暗に **Object** を統合するより、こういう利便性を考慮して統合したほうが良い。

一方で Normal モードは、辺や面の法線方向を基準に Manipulator が動く。この Cube では分かりにくいですが、Isosphere を例にとると動作が分かりやすいだろう。



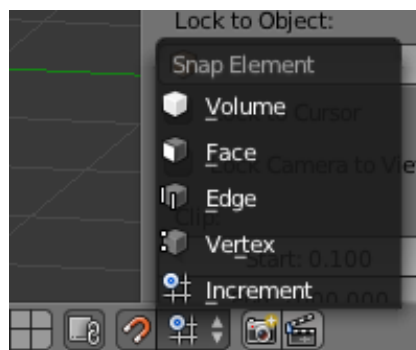
Normal モードの Manipulator

選択されている面に対して垂直な方向に青い Z 軸が伸びているのが分かる。このモードは形状が複雑になってきたときに重宝するだろう。

いずれのモードでも、**G**, **R**, **S** を押した後に座標軸のキーを **2** 回押すことで、現在の Orientation に沿った方向へ操作を制限することができる。

Snap について

ここでいう **Snap** は、**Vertex**, **Edge**, **Face** を別の対象に沿わせるための機能だ。前の **Cursor Snap** 機能とは関係がないので注意が必要だ。



Snap Element メニュー

上図が Snap Element メニューだ。添わせる対象を選択することができるが、良く使うのは **Increment** 以外の 4 つだ。Snap 対象選択はマウスでなければできないが、機能のオンオフはマウスでもショートカットキーでも可能である。

この機能の動作は弄るとすぐに分かるので、是非やってみてほしい。Snap をオンにした状態で、頂点や辺を移動させてみればわかる。

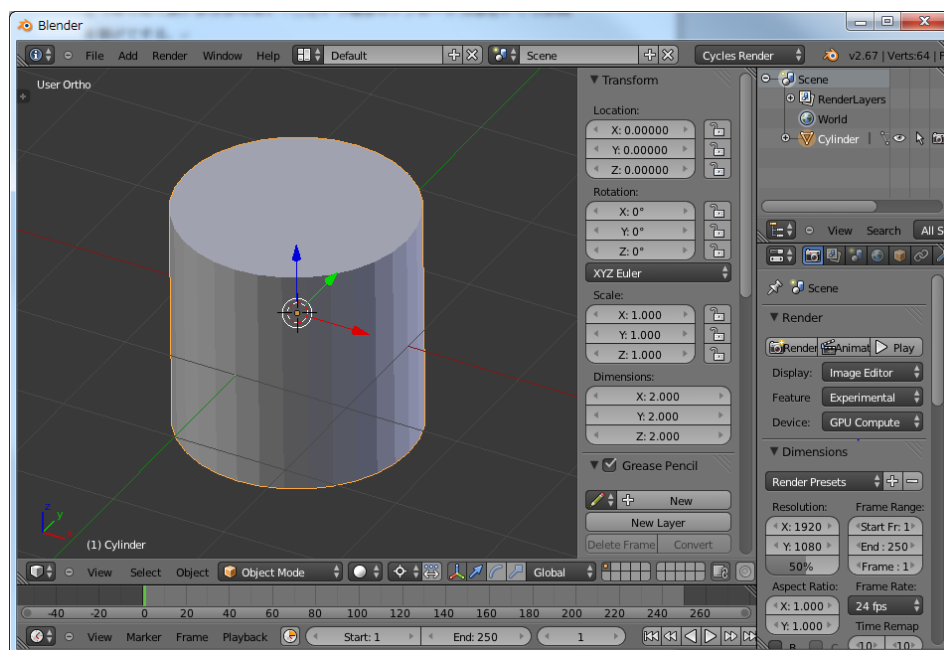
プリミティブから変形していく方法

目的の形状を目指して立方体、球、円柱などから変形していく方法。形が画一的な機械や、ハイポリのために 4 角形の面にこだわった場合の動物などに用いる。

チェスのポーンを作る

第 4 章の例で出したポーンはこの方法で作った。

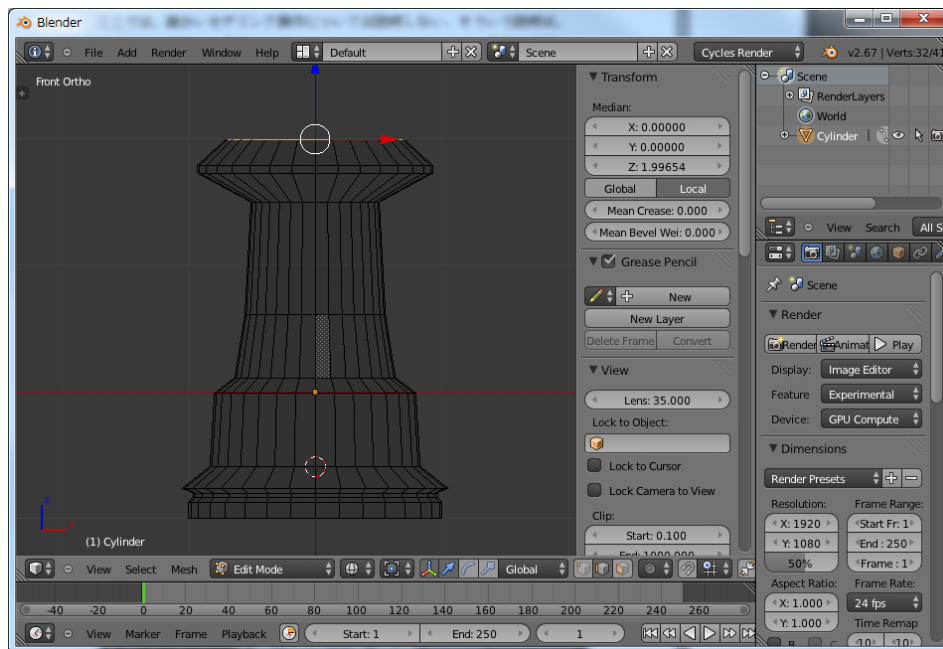
まず、最もチェスの駒を作りやすいであろうプリミティブを考える。ポーンの場合、てっぺんの球体を除けば円断面なので、円か円筒だろう。



Cylinder を新規作成

今回は Subsurf をかけるつもりなので、円筒を何角形にするかは、それを考慮して決めた方が良く。元から十分な細かさを持たせると、分割した時に無駄に細くなってしまうからだ。

さて、次の段階は断面形状を整えていくことだ。天辺の面を削除して、断面形状を作っていく。

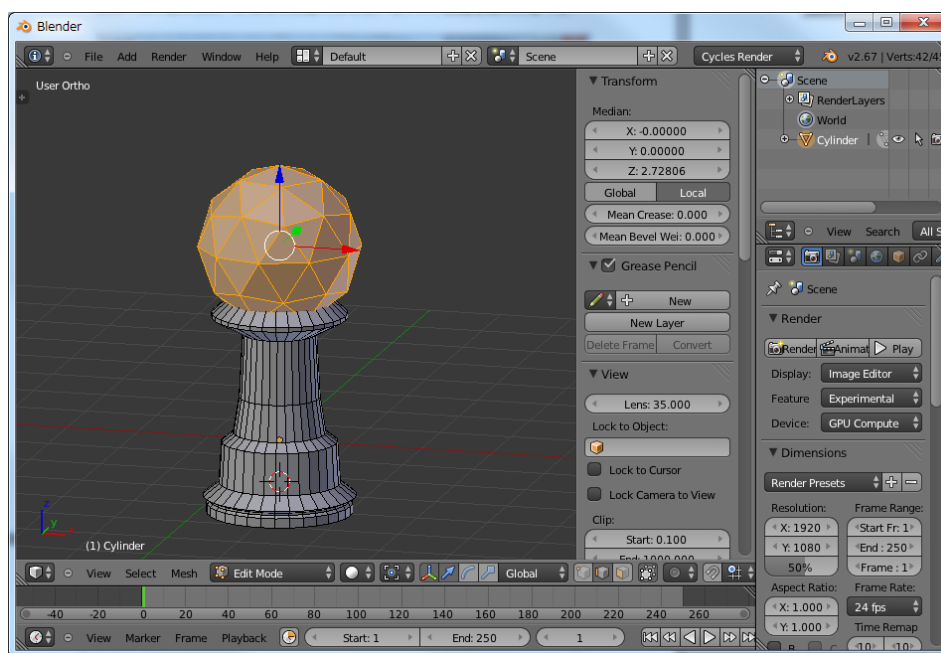


断面成形

- Loop Cut : **:Ctrl+R**
- Extrude(押し出し) : **:E**
- Scaling(拡大縮小) : **:S**

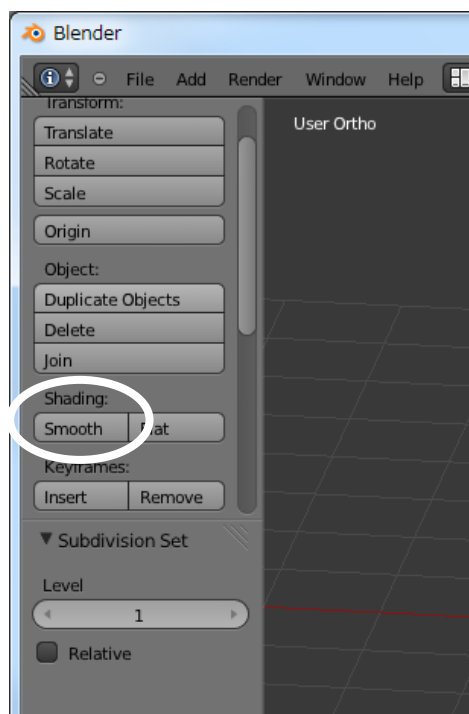
あたりの機能を使いながら、形状を決めていく。

これが終わると、今度は天辺に球を追加する。今回は球に特にテクスチャを貼るわけでもなく、きれいな分割と少ないポリゴン数が求められるだけなので、Icosphere を追加する。Edit Mode 中で Mesh を追加すると、同じ Data 内部に形状を作ることができる。Edit Mode を抜けてから Isosphere を追加すると、別 Object となる。今回は別 Object に分ける必要が特にないので、Edit Mode 中で追加した。



形状の完成

これで形状は完成だ。今のままだとカクカクなので、Smooth をかけておこう。
Smooth ボタンは **T** で出てくる Tool Shelf に格納されている。



Tool Shelf の Smooth ボタン

ここで、本当はキッチリとラインが出てほしいところまで滑らかになってしまふ。それを防ぐために EdgeSplit Modifier を追加する。

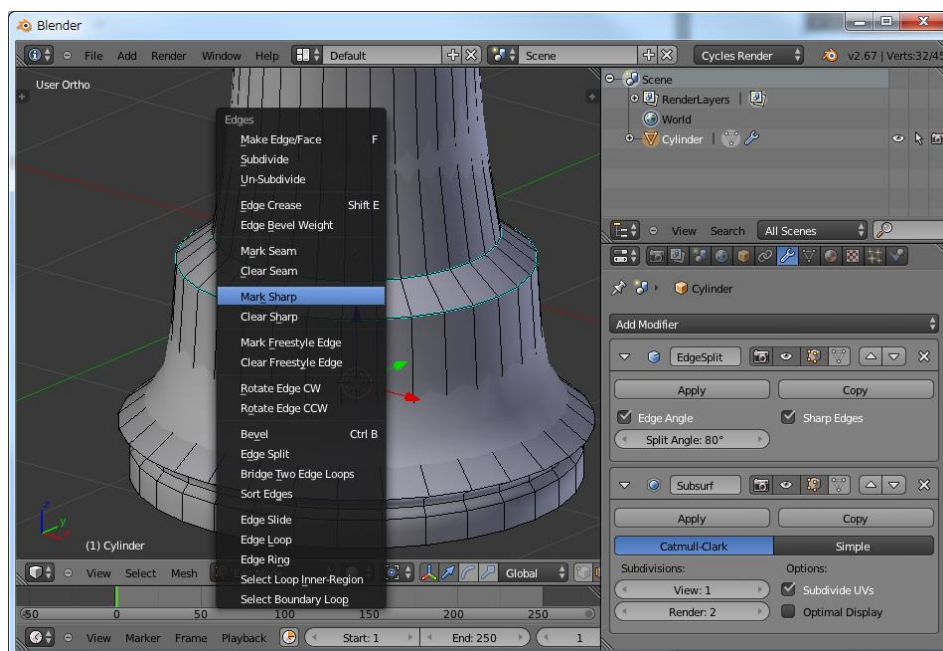
次に Modifier の設定をおこなう。

- Subsurf を n 段階かける : **Ctrl**+**n**(数字キー)

この操作によって、簡単に Subsurf Modifier をかけられる。これはモデリング中にかなり多用する操作なので、覚えておくといい。今回は 1 段階で十分なので、**Ctrl**+**1**を押す。すると、すべての面が 1 回分割されて滑らかになる。とがってほしいところは **Ctrl**+**E**で出てくる Edge 編集メニューから Mark Sharp を選択すると、その時選択されている Edge が EdgeSplit の効果対象となり、角張った感じになる。選択時には

- 辺一周選択 : **Alt**+**右クリック**

の操作が便利だ。

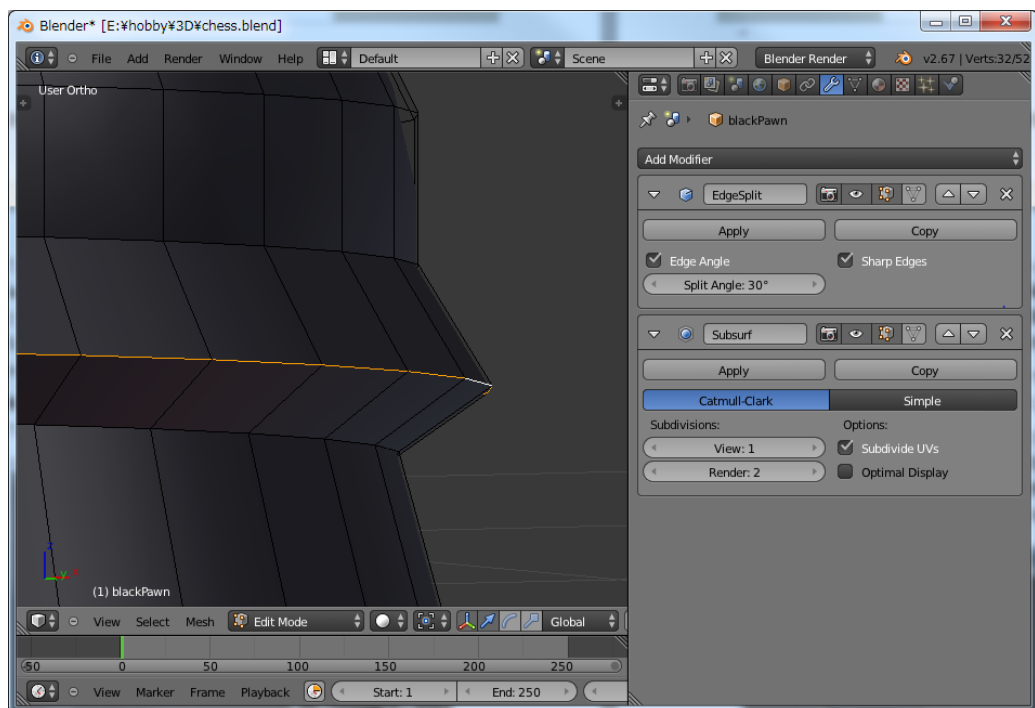


Edge を EdgeSplit の対象にして角を出す

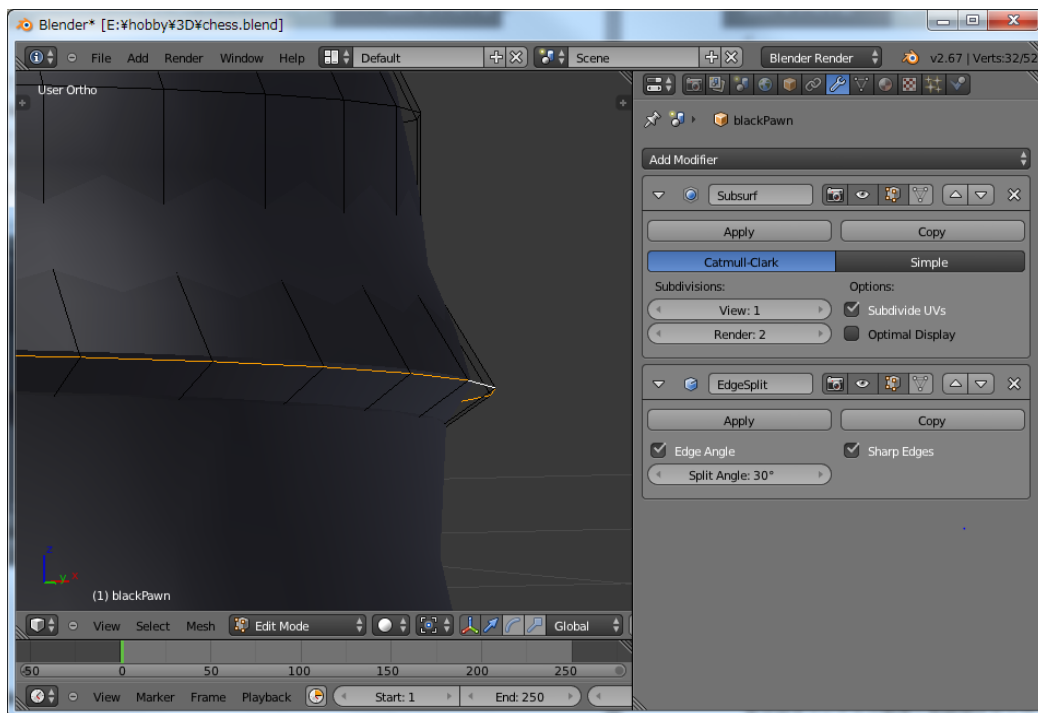
ここで注意すべきは、**Modifier** をかける順序だ。**Modifier** は上から順に適用される。なので、先に EdgeSplit をかけてから Subsurf をかけるのと、先に Subsurf をかけてから EdgeSplit をかけるのとでは、結果が異なる。先に

EdgeSplit をかけると，Mark Sharp で設定した辺が切断されてから Subsurf がかかるので，尖った角が辺とピッタリ一致する．逆順でかけると，Subsurf で滑らかになってから切断されるため，見えている辺とずれたところがシャープになる．

この辺は自分で確かめてみたほうが良いだろう．たいていの場合は EdgeSplit が先の方が所望の結果が得られる．

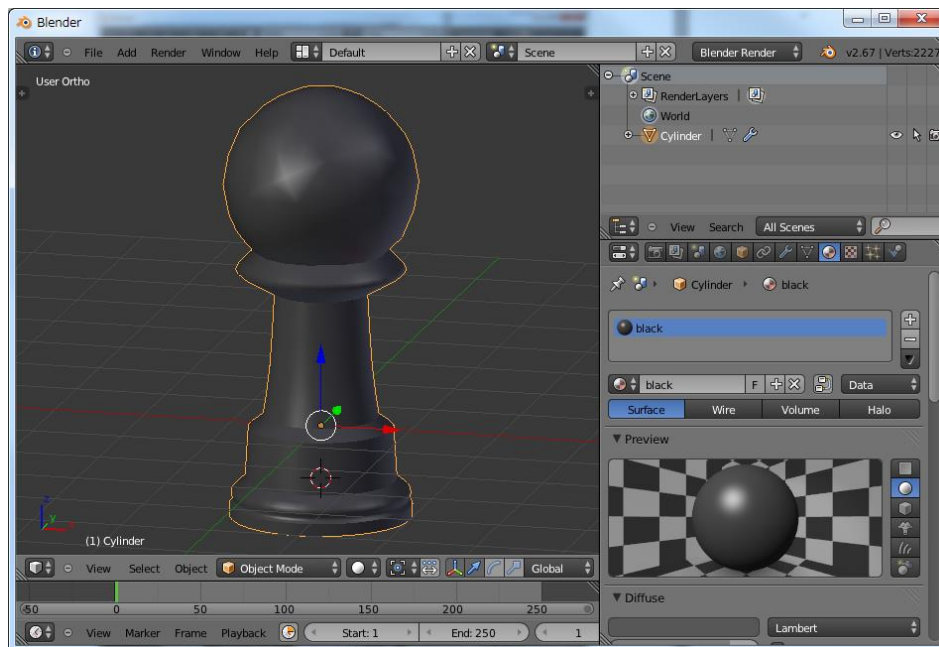


先に EdgeSplit をかけた場合



先に Subsurf をかけた場合

この操作が終わったら、最後に Material を設定しておしまいだ。Material 設定については検索すれば色々と情報が出てくるので、ここでは割愛する。



Material 設定を行う

殆どの 3DCG は、以上の操作をひたすら繰り返すだけだ。どんなに複雑に見える形状も、大抵はプリミティブからの変形で作れる。特に人工物ならその傾向が顕著だ。自然物であっても、木なんかは特徴を捉えれば円柱の変形でいけるだろうし、石も球からの変形でいけるかもしれない。どの方法が正解という事はなく、その辺は好みや習熟度に合わせて変えるべきだ。

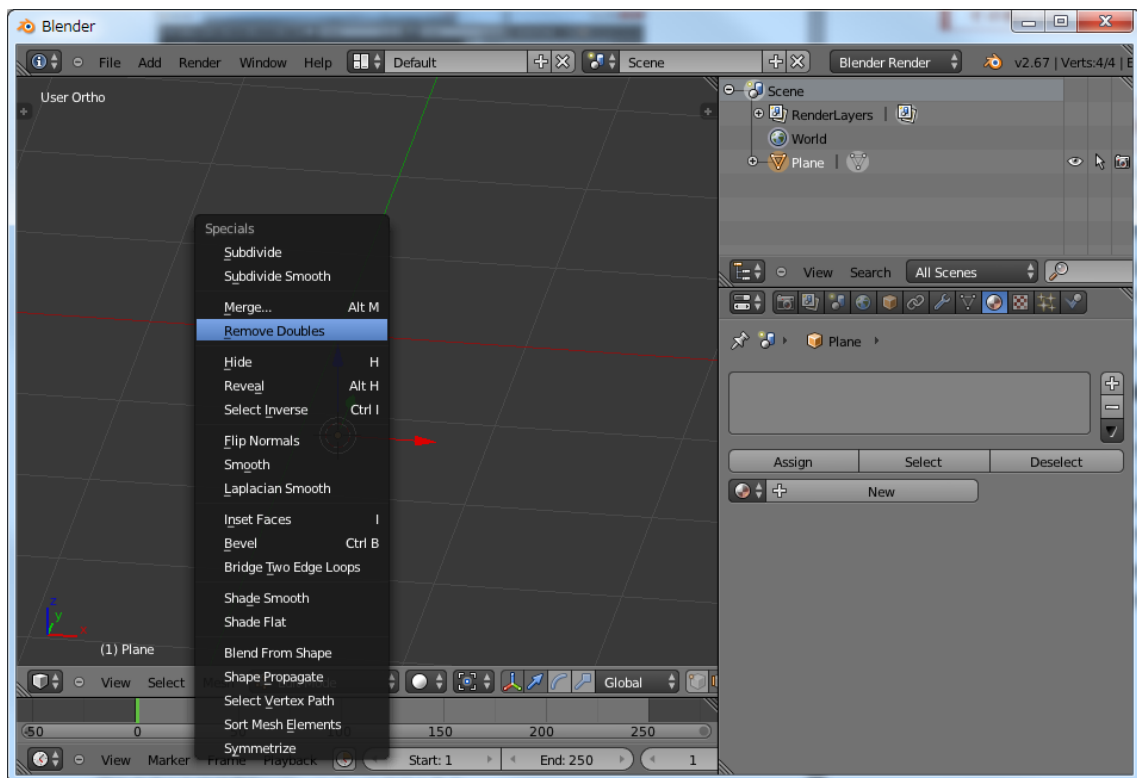
断面図を作って面を貼っていく方法

3 面図が用意できる場合や、形状が複雑に捻じれていたり、微妙な形状であったりする場合や、ローポリのために 3 角形ポリゴンを最高効率で貼っていくことを目指す時に用いる。

自分はオリジナルの戦車や戦闘機ばかり作っているんで、あまり断面図からモデリングすることはない。戦闘機ですら、箱の胴体と箱の翼、円柱の機首から作っている。なので、この方法にはあまり習熟していない。なので、適切な例を出すことが出来ないが、こういう方法もある。

大抵のチュートリアルでは、まず頂点を **E** で押し出して辺を作り、一周の形状を成形していく。その後、そこから面を押し出して成形していくようだ。

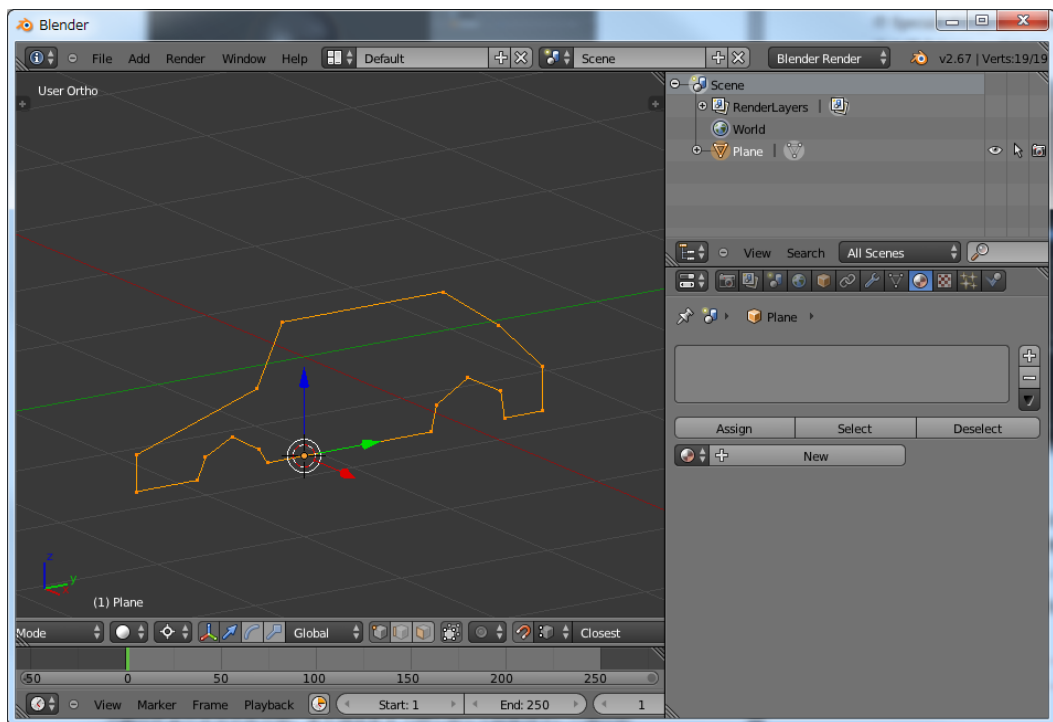
初めに、Plane を作り、**Tab** で Edit Mode に入る。**S**→**O** の順でキーを押すと、**o** 倍の拡大操作がなされるので、すべての頂点が一点に集まる。そこで **Ctrl**+**W** の Specials メニューから Remove Doubles を選択すると、同位置に存在する選択中の頂点が Marge されて、頂点がひとつになる。



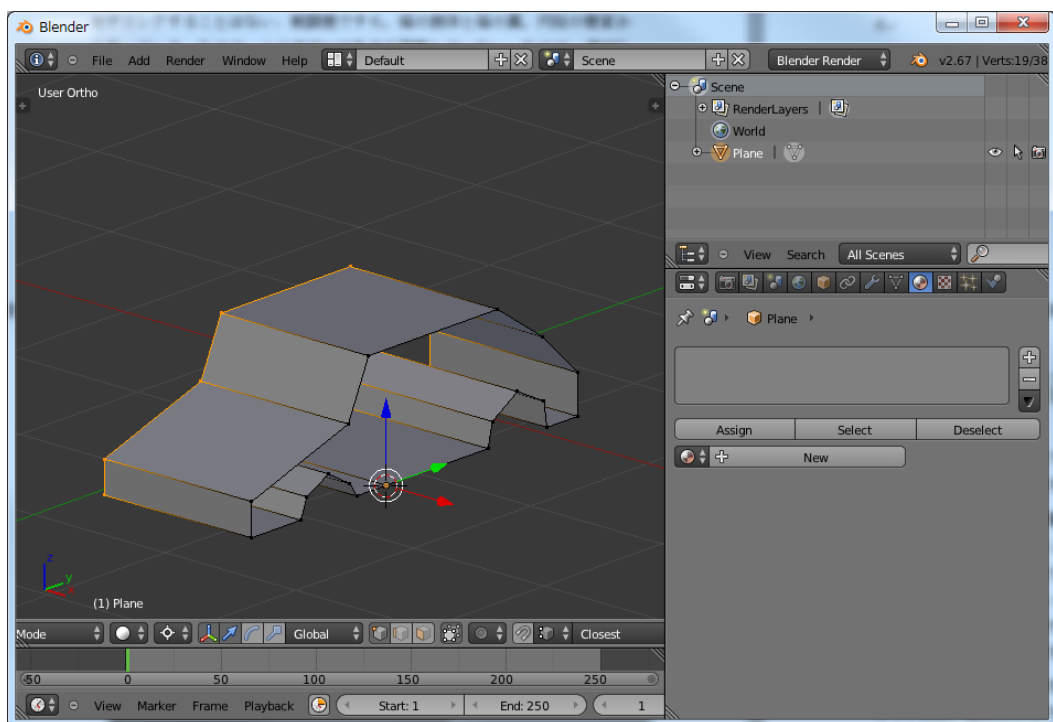
頂点をまとめる

実のところ、この操作は **Alt**+**M** の Marge 機能で可能な事ではあるが、**o** 倍の拡大縮小という操作は汎用性が高いので、覚えておいて損はない。がたがたの頂点の X 座標をそろえたいときは、**S**→**X**→**O** と押せばよいのである。

さて、頂点を纏めたらあとは **E** でどんどん辺を作っていけばいい。一周したら、全体の辺を **E** で押し出して基本形状を作る。あとは先ほどのポーンと同じように作りこんでいくだけだ。



断面形状を作り終わったところ



E で押し出し

あまり説明されていないが便利な機能

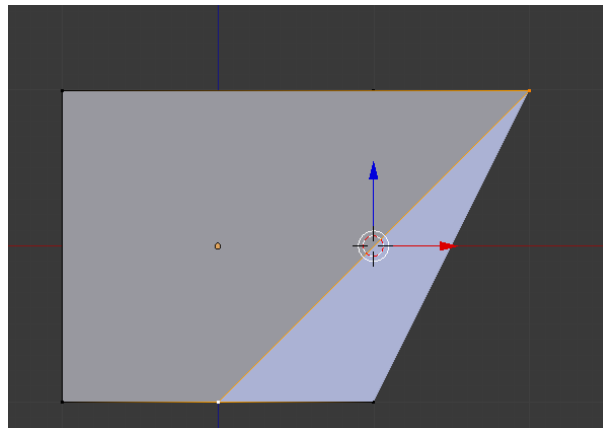
これは筆者の主観だが、あまり説明されていない割にとっても便利な機能がいくつかあるので、紹介しておく。

- Shear(せん断変形) :**Ctrl**+**Alt**+**Shift**+**S**
- Slide Mode :**G**→**G** (G キーを 2 度押す)

以上の 2 つは覚えておくと便利だ。

Shear(せん断変形)

この機能は、長方形を平行四辺形にするような変形を司る。下の図を見たほうが分かりやすいだろう。

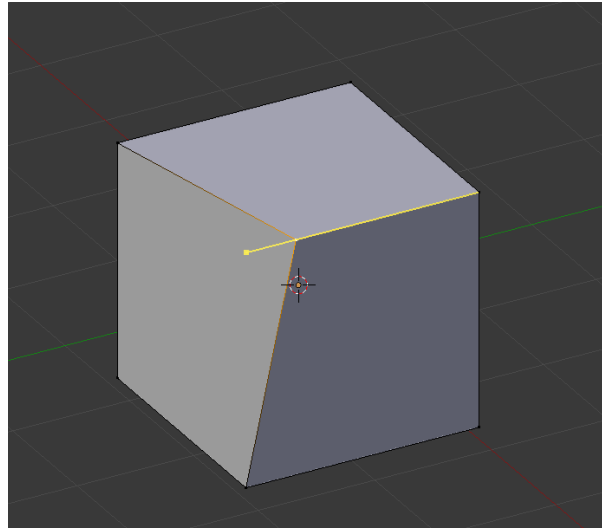


Shear 変形後の Cube

辺に対して Shear をかけると、Pivot Point の場所は動かず、Pivot Point から離れるほど移動量が大きくなるような変形がかかる。マウスで操作すると物凄い移動量が必要になるので、移動軸の制限や数値入力と組み合わせて使おう。

Slide Mode

[G]を2回押すと、Slide Mode が選択される。選択されているのが Vertex か Edge か Face かで自動的に動作が変わるので、非常に便利だ。



Slide Mode 中の Vertex

特に Subsurf をかけているときに、現在の形状を維持しながら丸みのかかり具合を調整したいときなどに重宝する。

おわりに

この文書では、あえてあまりモデリングそのものには触れていない。これは、モデリング以外にも 3DCG の要素は沢山あるということを学んでもらい、良くある勘違いを防ぐためである。

良く勘違いされるが、3DCG とはいっても殆どの場合、**最終出力物は 2 次元のイラスト**である。なので、形状を作る**モデリングは全体の工程の一部**に過ぎず、案の作成からライティングや構図を含めた全ての行程が「3DCG の製作」なのだが、そのことを忘れて形状の再現にひたすら邁進してしまう人が多い。

モデリングに対する手抜きは悪ではない！そんなこと言っていると、テクスチャに色々な情報を付与してモデル自体を単純にしていくゲーム用モデルなどは、手抜きの産物でダメという事になってしまう。ゲーム用モデルはゲームで動かなければ意味がないので、闇雲にモデルを精密化するわけにはいかない。目的に照らし合わせて考えれば、モデリングに手をかけすぎるわけにはいかないのだ。モデリングの精密化とは別のテクニックが沢山必要になるだけで、むしろ何も考えずに精密化していくより大変だろう。

どんどん手を抜いて、労力をかけずに目的の出力物を得られるようになろう。そうすれば、結果的には作りたいものをどんどん作れるし、スキルアップにもつながるはずだ。

更新履歴

ver.1.0,2013/05/09：初版

ver.1.1,2013/05/12：章立てのミス，文体等修正

ver.1.2,2013/05/15：文章修正

ver.1.4,2014/03/07：p.19-20 の誤植修正