

Blender グリースペンシル 解説

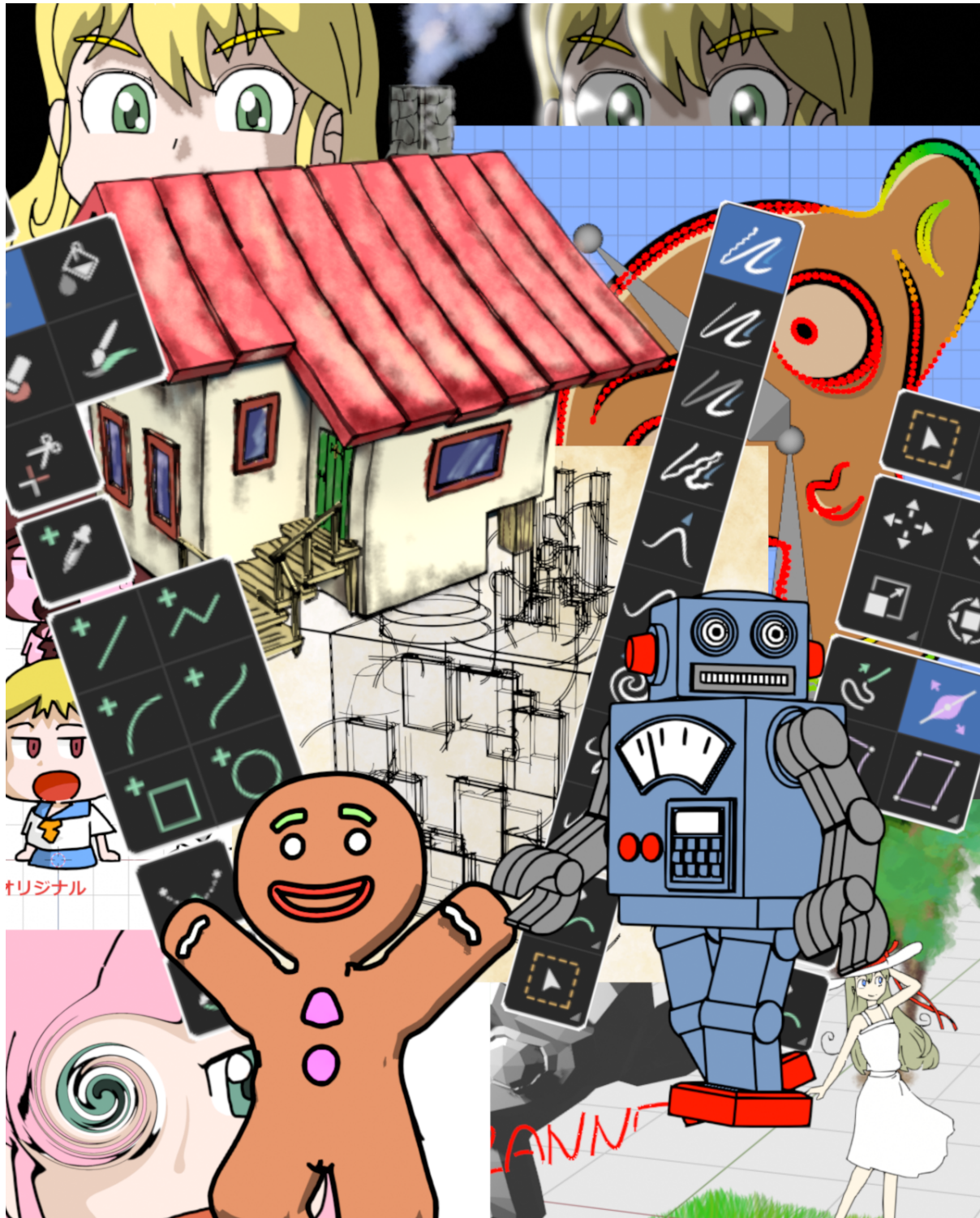
使いどころとモディファイアとビジュアルエフェクトの本

Version 1.6



Q@スタジオぼぶり
@popqjp

作者 Twitter アカウント
[Q@スタジオぼぶり](#)



目次

<p>初めに … 3</p> <p>グリースペンシル基本操作 … 5</p> <p>平面上のグリースペンシル … 5</p> <ul style="list-style-type: none">スザンヌとレイヤー構成 … 5光源とグリースペンシル … 6線と塗り (マテリアル) … 7 <p>表示色とカラーマネージメント … 9</p> <ul style="list-style-type: none">線のパラメーター … 9線を描く。レイヤー内の上下関係 … 10線をつなぐ。長い塗りつぶしの線 … 11簡単なお絵かき … 12 <p>塗りつぶし … 13</p> <ul style="list-style-type: none">バケツツールで塗りつぶし … 13中空の塗り。ホールアウトとマスク … 16 <p>テクスチャを使う … 18</p> <ul style="list-style-type: none">ストロークのテクスチャ座標編集 … 19点描とテクスチャマッピング … 20カラー属性 … 22スポイト(Eye Dropper)で色取得 … 23 <p>点や線の削除 … 24</p> <ul style="list-style-type: none">消しゴム(Eraser)ツール … 24ツール切り替え無しの削除 … 25カッター(Cutter)ツール … 25塗りの削除 … 26ゴミの削除 … 27 <p>オブジェクトやレイヤー操作の便利機能など … 28</p> <p>グリースペンシルの編集やスカルプト … 30</p> <ul style="list-style-type: none">スカルプト … 30カーブ編集 … 32 <p>3D形状としてのグリースペンシル … 34</p> <ul style="list-style-type: none">3次元オブジェクト上で描画 … 34線と線を結び描画 … 35オブジェクト上にカーブを描く … 35点描とテクスチャを駆使した、立体への塗り … 36 <p>キーフレームアニメーション … 37</p> <ul style="list-style-type: none">オニオンスキン … 38補間 … 40 <p>補間と閉じたストロークの始点、終点 … 43</p> <ul style="list-style-type: none">カーブ編集と補間 … 43 <p>モディファイアを使いこなす … 44</p> <ul style="list-style-type: none">Subdivide(細分化) … 44Array(配列) … 45Offset(オフセット) … 46Dot Dash(ドット点線) … 48Simplify(簡略化) … 49等間隔の破線 … 51点描のアニメーション … 51Smooth(スムーズ) … 52Mirror(ミラー) … 53Shrink Wrap(シュリンクラップ) … 53Noise(ノイズ) … 54 <p>Vertex Weight Proximity(頂点ウェイト(近傍)) … 56</p> <ul style="list-style-type: none">Vertex Weight Angle(頂点ウェイト(角度)) … 57Thickness(幅) … 57	<ul style="list-style-type: none">Multiple Strokes(マルチストローク) … 58マスクを使った網目パターン利用 … 59Outline(アウトライン) … 60アウトラインとカーブ編集(房のアニメーション) … 61Envelope(エンベロープ) … 61Texture Mapping(テクスチャマッピング) … 63 <p>Line Art … 64</p> <ul style="list-style-type: none">Line Art(ラインアート) … 64光源の設定(Light Reference) … 65Line Artパラメーター概略 … 66Length(長さ) … 71補助線付きの設計図スケッチ風味 … 72 <p>色に関連するモディファイア … 73</p> <ul style="list-style-type: none">Hue/Saturation(色相/彩度) … 73Opacity(不透明度) … 74Hardness(ハードネス)とぼかし … 74Tint(チント) … 75 <p>アニメーション制御やアニメーション的に動かせるモディファイア … 76</p> <ul style="list-style-type: none">Time Offset(タイムオフセット) … 76表情のコマアニメーション … 77Build(ビルド) … 78Lattice(ラティス) … 81メッシュによる変形 … 82Hook(フック) … 83原点以外の位置での Hook … 84Line Art Bake(ラインアートのベイク) … 84 <p>Armature(アーマチュア) … 86</p> <ul style="list-style-type: none">頂点ウェイトの設定を使って影響範囲を制御 … 86エンベロープを使って影響範囲の制御 … 88パーツ単体の制御 … 88パーツの一部を、コマアニメーションさせる … 89普通の歩行アニメーション … 90 <p>ビジュアルエフェクト … 93</p> <ul style="list-style-type: none">Blur(ブラー) … 93Colorize(カラーライズ) … 94Flip(反転) … 94Glow(グロー) … 95Pixelate(ピクセル化) … 96Rim(縁取り) … 96Shadow(影) … 97Swirl(渦巻) … 98Wave Distortion(波の歪み) … 99ビジュアルエフェクトと画面範囲 … 100 <p>その他のTIPS … 101</p> <ul style="list-style-type: none">ビューの回転 … 101Grease Pencil Tools … 101 <p>バケツ塗りツール使用時の描画、細かい設定 … 102</p> <ul style="list-style-type: none">Line Artの細かい挙動 … 103 <p>動画サンプルと.blend ファイル … 106</p> <ul style="list-style-type: none">終わりに … 106
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Blender グリースペンシル 解説

使いどころとモディファイアとビジュアルエフェクトの本 version 1.6



Q@スタジオぼぶり
@popqjp

作者 Twitter アカウント
[Q@スタジオぼぶり](#)

2022.05.10 ver 1.0

2022.05.18 ver 1.1 削除とTIPSのセクション、テキストチャ座標編集を追加

2022.06.03 ver 1.2 スポイトと、Line Art用 TIPS セクション追加

2022.06.13 ver 1.3 blender 3.2 対応

2022.09.02 ver 1.4 blender 3.3 対応

2022.12.03 ver 1.5 blender 3.4 対応

2023.08.03 ver 1.6 blender 3.6 対応

初めに

この本は、blender のグリースペンシルについての解説本になります。

グリースペンシルは、3DCGソフトである blender でシーンに配置するオブジェクトの一種で、

手描きの絵などをシーンに描き込む、といったことができるように作られている、一種独特なオブジェクトです。

2Dのパーツとして複数配置して2Dアニメーションを作成したり、3Dのシーン上に2Dの絵をはめ込んだハイブリッドなアニメーションを作成したり、3D空間上に描き込んでレタッチの様に使用したり、様々な使い道があります。

元々は、3Dシーンに手描きのメモや下描きを描く、ということを行う機能でした。

blender 2.79 から 2.80 に大きく更新される時に、大幅に機能の整備が行われて、2.80 の目玉機能の一つとなったものです。

旧来の「シーンに手描きのメモを追加する」という機能は、グリースペンシルの機能を流用した特別な形態として、アノテーション(Annotation)という名前で残っています。



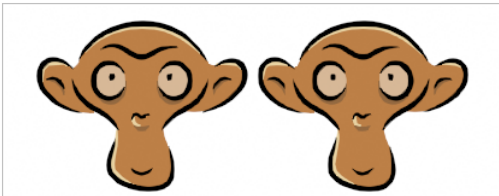
シーンにメモ書きをする機能として、
ペンのアイコンのツールで、シーンにAnnotationを描き込めます。
グリースペンシルも元々はこういう感じに、メモ書きをするツールが起源でした。

それが線と塗りを描画するオブジェクトに進化して、
図のイラストの様に シーンの中に絵を描き込むことができるようになったものがグリースペンシルです。

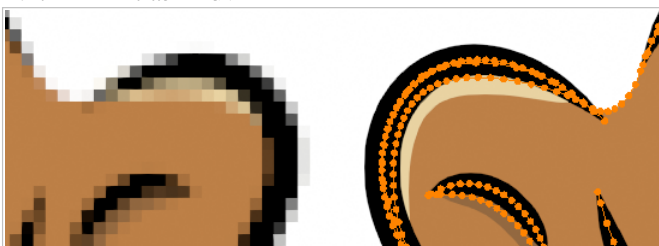
グリースペンシルはシーン内のオブジェクトの1つとしてレンダリングがされます。

アノテーションで絵などを描いても、プレビュー画面に表示されるだけでレンダリングはされない点に注意してください。

グリースペンシルのデータは、一般的な「画像」としてイメージされるラスタ画像ではなく、ベクトル画像に分類されるデータです。



一見同じような画像でも拡大すると…



ラスタ画像は、ピクセルの集合体でドットごとに色のデータを持っているような形式です。

普通の画像データは大きく拡大すると、左の図のようにカクカクが見えてきますね。
一方ベクトル画像は、主線や輪郭線といったものが、頂点をつないだものとして表現されます。

これは、3DCGモデルが、たくさんの頂点を結んだポリゴンで出来ているのに類似した考え方です。
頂点をたくさん使えば、ポリゴンでも滑らかな曲面でできた立体を表現できるように、充分頂点があれば滑らかな絵を表現できます。
blender を使っていれば、普段そうした3DCGモデルを多く扱っているでしょう。
そうした頂点の集合体という考えを、線や塗りを表現する方向に拡張したものだと考えれば、掴みやすいかと思います。

ベクトル画像には、ラスター画像には無い様々なメリット（拡大縮小に無制限に対応ができる、後から線の修正が容易である等）があります。
一方で、ラスター画像では分かりやすい表現や考え方（塗りつぶしの扱いや、グラデーションの表現等）で意外な扱いづらさが出る場合があります。

blender のメッシュオブジェクトには、モディファイアを使って様々な効果を与えることができました。
グリースペンスシルにも、描き込んだ後の線データに様々な効果を与えるモディファイアが用意されています。
この本は元々、モディファイア芸の本「モディファイア芸でいろいろな効果を作ってみよう」の姉妹編のつもりで、
グリースペンスシルのモディファイアについてまとめようとして書き始めたものです。

しかし、いきなりモディファイアを使っていろいろと効果を与えるのは敷居が高すぎ、
まずはグリースペンスシルで絵を作成するところの解説が無いと厳しいだろう…！
ということが分かり、グリースペンスシルで絵を描くための基本操作から入り、
その応用として、モディファイアなど様々なエフェクトをどのように利用していくかを解説する流れの本にしました。

Twitterでの普段の投稿から分かるように、作者は絵は上手ではありません。
グリースペンスシルを使って、上手な絵を描くための画力に関するノウハウに関しては、ほとんど参考にならないでしょう！
「これをするには、どう操作をするのか」
「この機能の使い道は何か？」
という、操作法や機能の説明についての内容になります。

最初にこの本が描かれているのは、3.1 がリリースされるころから、そろそろ3.2がやってくるぞ、というころの時期です。
初版での機能の紹介もそのころの内容に沿ったものですが、その後のグリースペンスシルの機能追加や変更に合わせて、バージョンを更新しました。
ver 1.3 として blender 3.2 で、ver 1.4 として blender 3.3 で増えた機能などについて追記をしました。
ver 1.5 では blender 3.4 で増えた機能に加えて、それまでの版で解説していなかった項目を幾つか追加しました。
ver 1.6 では、blender 3.6 までの変化に対応して、幾つかの解説を追加しました。

グリースペシル基本操作

グリースペシルのモディファイアをまとめて解説をしよう…というのが元々のコンセプトですが、グリースペシル自体が、比較的新しめの機能ですし Blender 使いなら必ず使う…！という機能ではないので、なじみの浅い人も多いのではないかと思います。そこで、グリースペシルの基本的な機能に関しての章を最初につけておくことにしました。

「まったくのblender初心者が、グリースペシルを使うための入門書」というには、だいぶ端折った内容になっています。blender の操作は大体分かっていてメッシュの編集とかも普通にできるよ。(でも、グリースペシルは、たまにしか使ってないから、機能を良く把握していないな…)というレベルの人がざっとおさらいをする程度のレベルになっています。

平面上のグリースペシル

グリースペシルの売りの一つは「3D空間上にもお絵かきができる」ことです。とはいえ、やはり絵を3D空間上に描くのは大変です。何の補助もなく3D空間上に自在に線を引くのは簡単ではないですし、3Dモデルの上に描き込みをするのも、曲がったキャンパスの上に絵を描くようなものです。

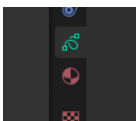
やはり基本的な使い方は、2D的な絵を作成してそれを3D空間に配置する、というものだと思います。そこで、まずは3Dであることは無視して、2Dの平面上でのお絵かきについての基本操作を見てみます。

スザンヌとレイヤー構成

サンプルのスザンヌの構造を見てみると、グリースペシルの仕組みがある程度見えてきます。



Add(追加) - Grease Pencil(グリースペシル) - Monkey(モンキー) でサンプルに用意されているグリースペシルのスザンヌを作成してみます。



グリースペシルの情報は(普通のメッシュと同じように)アウトライナー上で見ます。特に重要なのは、この2つのアイコン(Data と Material)のタブです。

データのタブを開いてレイヤーのパネルを見ると、2つのレイヤーから構成されていることがわかります。



目のアイコンでレイヤーごとの表示を切り替えて確認してみましょう。

Fills が塗り用のレイヤーになっていて、Lines が実線用のレイヤーになっていることがわかります。



デフォルト設定では、(3Dとしての線の位置に関わらず)上位に位置するレイヤーで下のレイヤーを上描きします。

平面上に塗りと線を描くような使い方であれば、一般的なペイントソフトなどと共通の使い勝手になる仕組みです。



パーツが立体的に組み合っているような複雑な場合は少し困ります。「オブジェクトを2つ並べる」のではなく「ひとつのオブジェクトの中で」スザンヌをコピーして前後に並べてみると、立体的な位置関係ではなく、レイヤーの上下関係によって上描きをされているのが確認できます。

コンソールを利用すれば、レイヤーのインデックスなど内部の情報に分かるのですが、「Fills」がレイヤー0、「Lines」がレイヤー1の番号になっています。

```
>>> bpy.data.objects['Suzanne'].data.layers[0].info
'Fills'
>>> bpy.data.objects['Suzanne'].data.layers[1].info
'Lines'
>>> |
```

インデックス順に、まず塗りつぶしの表示をして、その後で線を描画しているわけですね。

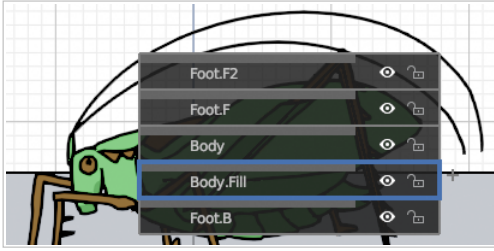
ところで、blenderの UI では、マテリアルのリストなどの普通のリストは、新しい要素は下に追加されていきます。

レイヤーに関しては、それでは一般のペイントソフト等と逆になってしまうので、合わせるためにレイヤーのリストは順番を逆に表示しているわけです。

サンプルのスザンヌはレイヤーを2枚使っていますが、レイヤーが1枚でも塗りと実線のどちらもあるような絵の表現はできます。

ただし、前後関係の管理などがややこしいこととなります。

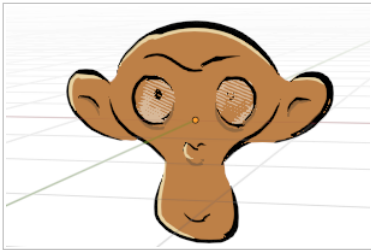
実線と塗りがあるような絵の表現には、スザンヌのサンプルのように2枚のレイヤーのペアを使うのが基本のようです。



もちろん、もっと描く要素が増えれば3レイヤー4レイヤーと増やしていくこともできます。このクツワムシを描いた例では、胴体と前の足、後ろの足などで5レイヤー使いました。構造が複雑な胴体部分は、実線用のレイヤーと塗り用のレイヤーのペアにしています。

レイヤー機能を多用するときには、アクティブレイヤー変更のショートカット「Y」を覚えておくと便利です。

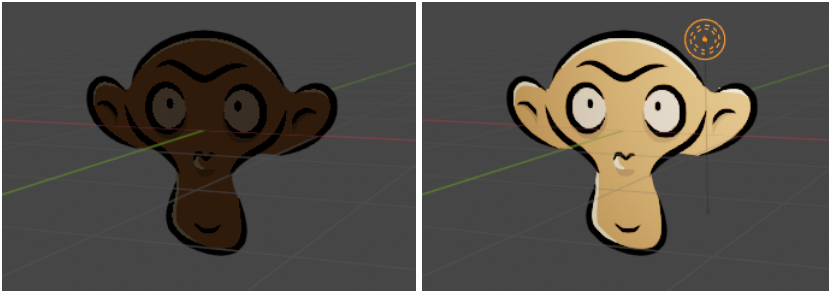
ストロークの Depth Order(配置方法)の設定で、3D位置を基準にすれば、レイヤー順ではなく位置で前後関係を決めて描画します。



サンプルのスザンヌのような場合は、同じ面に肌と白目部分があったり、塗りの面と線があったりするので、前後関係が判断できない、いわゆるz-fighting が起こってあまりうまくありません。

光源とグリースペシル

スザンヌを表示させた後で、プレビューのシェーディングの設定を「レンダリング」モードにして実際のレンダリングの具合を見ると、予想以上にスザンヌが暗く驚きます。(Blenderのデフォルトのセッティングのままだと…)

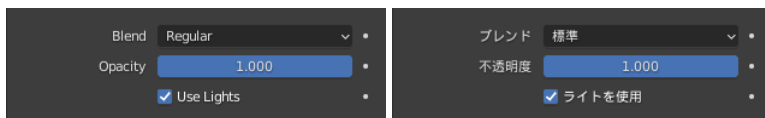


実際にレンダリングで表示される色は光源に反応すると、デフォルト状態だと光源がスザンヌの背後にあるからです。

光源を手前に移動させてみると、明るく照らされるのが分かります。

(ただし、影を落としたり、別の3Dオブジェクトの影が落ちたりすることはありません。単純に光源から照らされる明るさと光の当たる角度、そして表か裏かだけで色が決まります)

この光源に反応する挙動は、立体物の中にグリースペシルを配置してなじませるのには良いのですが、純粋な絵として使いたい時には、設定した色そのままに表示されて欲しいので困る場合があります。



光源に反応するかどうかの設定は2箇所設定ができます。レイヤーの設定に Use Lights(ライトを使用)があるので、レイヤーごとに光源への反応を指定できます。

ただし、多くのレイヤーを使う複雑なデータの場合、それぞれ設定をするのは少々手間がかかります。

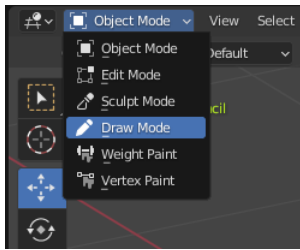


オブジェクト設定の Visibility(可視性)に、Use Light(ライトを使用)のオプションがあるので、これでオブジェクト単位でライトの使用の設定ができます。

オブジェクトがグリースペシルの時だけ表示されるオプションなので、ここに意外に気が付かなかつたりします。(著者はだいぶ長いこと気が付かず、レイヤー側のオプションだけを使っていました…)

線と塗り (マテリアル)

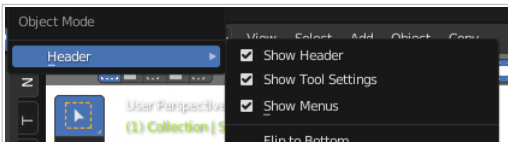
グリースペンをゼロから描きはじめる場合には、Add - Grease Pencil - Blank(ブランク)を選択して空のオブジェクトを作成します。GP_Layer という名前のレイヤーが1つだけあるグリースペシルオブジェクトが作成されます。



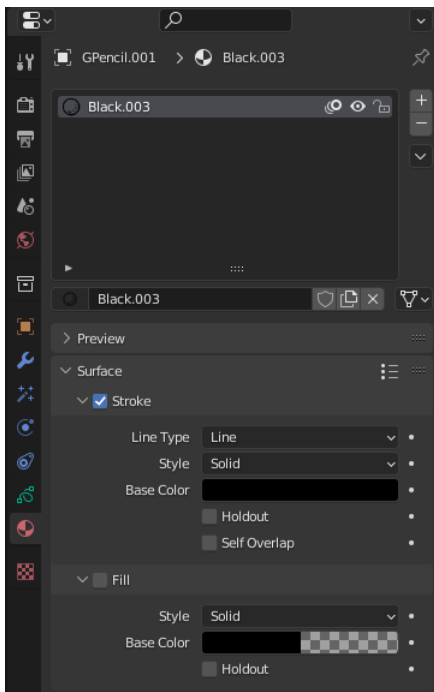
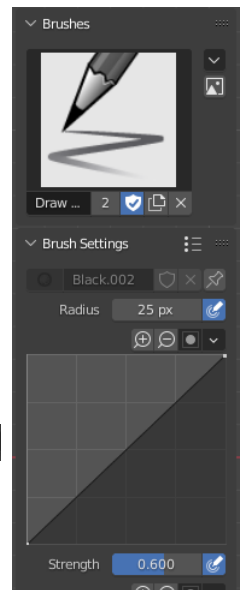
Draw Mode(ドローモード)にすることで描き込みができます。

ブラシの選択や設定(太さや濃さ)などは3Dビュー右側のパネル(Toolタブ)から編集ができます。筆圧のグラフが大きくて場所を結構食いますが、筆圧Offであればグラフは隠れます。

大きいモニターだったり、ビューを大きくとってあげれば良いのですが、画面右のパネルが手狭な場合には、ヘッダーにもツールの設定の表示をさせることができます。(ヘッダーの位置で右クリックメニューで表示/非表示を設定できます)



どちらを主として使うかは、環境や好み次第です。



マテリアルはあらかじめ作成して登録されたマテリアルから選択します。

このあたりの感覚は、メッシュにマテリアルを適用するときに、マテリアルスロットに登録済みのマテリアルから選ぶのと似ています。

新規のグリースペシルを作ると、Black(.*?) という黒の実線が新たに作成されていきます。

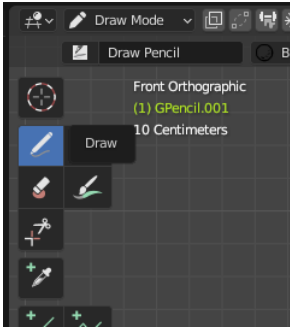
(マテリアルが増えすぎて困るときは、他のグリースペシルで使ったマテリアルを使いまわすなどして、こまめな整理が必要です)

マテリアル毎に、色などの設定を編集します。

Stroke(ストローク)とFill(フィル=塗り)の設定が別々にあるので、

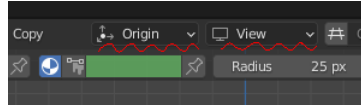
1. 実線用
2. 塗りつぶし専用
3. 実線+塗りつぶし用

のマテリアルを使い分けることになります。



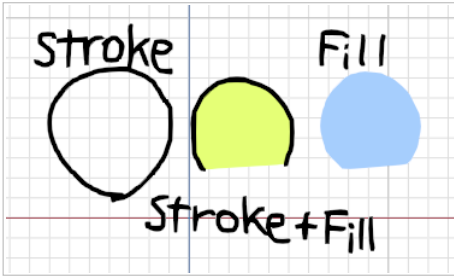
ドローツールで描き込みができます。

どこを基準にして、どの向きを向いた平面として絵を描くか？ という設定はこの2つの設定で行います。



3次元的なお絵かきができるのが売りなので、どのようにビューの向きを設定しても良いのですが…最初は、分かりやすいように X,Y,Zなどの軸に合わせた描画をするのが無難でしょう。

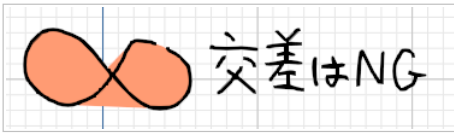
デフォルトのモードは、描画の基準はOrigin(原点)になっていて、ビュー方向に合わせた描画になっています。ビューを軸に合わせておけば、それに合わせた平面上に描画されることになります。



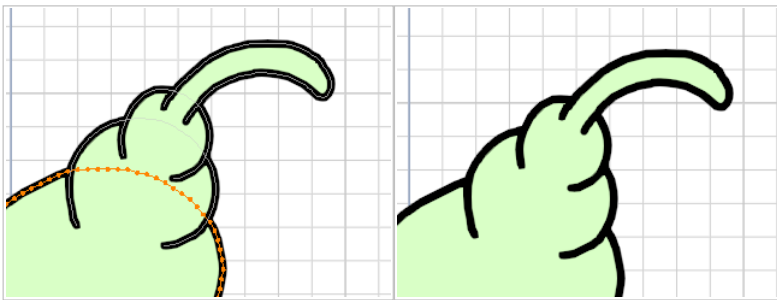
線と塗りを設定したマテリアルを3種類作って描画をしてみます。

当然ですが、図のように表示がされます。

閉じ切っていない線を描いて Fill 設定をすると、最初と最後の制御点が直線で結ばれた塗りつぶしになります。



また、塗りつぶしに関してはかなり複雑な形状にも追従するのですが、線が交差するような場合には対応できないことに注意が必要です。



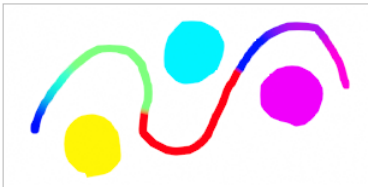
塗りと実線をどちらも設定したマテリアルによる描画は、このような一部が飛び出た形状などを描く時に便利なことがあります。最初に根元を描いたのちに、2段目を描く、3段目を描くというように順番に描くことで、下の段の線が自動的に隠れていきます。

(隠れる部分を描かずに、右パーツ左パーツとバラバラに描くと、左右の線の整合性が取りづらいですからね…)

マテリアルの色設定と、カラー属性

⚠ カラー属性は blender 3.1 まで頂点カラー(Vertex Color)と呼ばれていたパラメーターです。blender 3.2 での内部的な更新に伴ってカラー属性(Color Attribute)と表記されるようになりました。

いままで見たように、マテリアルごとに色が設定されている…ということは逆に言うと、使う色ごとにマテリアルを用意しないといけないのでしょうか？色数の少ないイラストはともかく、通常のお絵かきでは大変すぎます。メッシュのカラー属性と同様にグリースペンスルにもカラー属性が用意されています。カラー属性については、[後のセクション](#)で見てください。



マテリアルは単色の物を1つだけを用意して、カラー属性で色を付ける使い方も可能です。

逆に共通の色、例えば制服の色とか…はカラー属性に頼らずに、きちんとマテリアルを設定してその色を使っていれば、後から色の差し替えなどが簡単になります。このあたりは使い分けが重要ですね。

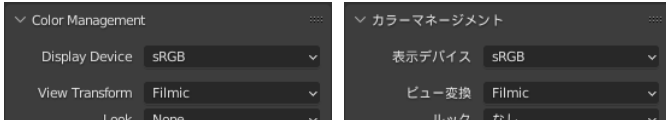
表示色とカラーマネージメント



光源に反応しないように設定して、つまり色がそのまま発色するはずの設定にして、原色などの分かりやすいマテリアルの色を付けて落書きをしてみます。

ところが、実際にレンダリングをしてみると、だいぶくすんだような色で表示されます。特に、背景を白色などに設定していると「白背景のはずなのにやけに灰色っぽい背景になるな？」と気づくはずで

す。これは Blender のデフォルトの表示の色空間の設定が「Filmic」という明るい部分をかなり暗くする設定になっているためです。フォトリアル調のCGで、白飛びをさせずに太陽光の様に強い(自然な)光源を扱うには重要な設定なのですが、イラストのように「この色で表示させたい」という色が決まっている時には裏目に出してしまいます。レンダリングの設定(レンダープロパティ)の Color Management(カラーマネージメント)の View Transform(ビュー変換)の項目を、必要に応じて Filmic から Standard(標準)に変更をします。



この色設定の変更忘れは、「フォトリアルではないスタイルの3DCG」を扱う場合にはすごく「あるある」な話なのですが、フォトリアル中心で、特にこのあたりの色設定の変更をしていない場合は、意外に盲点かもしれません。

直接比べてみると相当な違いがありますね。

Standard 設定では、Filmic でくすんだ雰囲気だったものが、元のマテリアルの強い色味に戻ることが分かります。

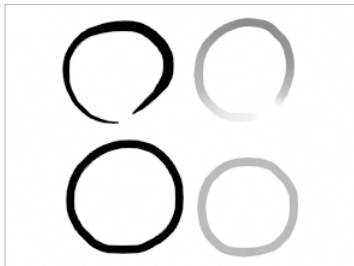


3DCGとの組み合わせ方によっては、Filmic の方が良くなじむ場合もあります。

「3DCGとどのように組み合わせるのか」のスタイルに合わせて、どちらの色設定を使うか最初に決めておきましょう。

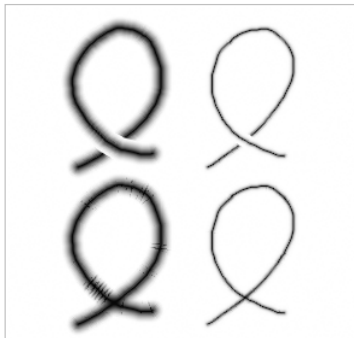
線のパラメーター

線を描くときの主なパラメーターは、Radius(半径)、Strength(強さ)、Hardness(ハードネス)等があります。



Radius(半径)は、線の場合は太さに対応します。
Strength(強さ)は、不透明度に対応します。

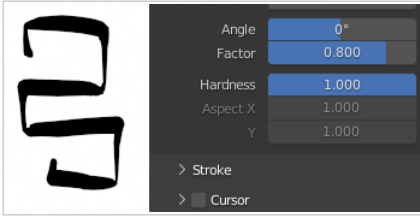
それぞれ、一定の値にしたり、筆圧設定で強弱をつけると左の図ようになります。
この2つは、「まあそうだろうな」というパラメーターで、非常に分かりやすいですね。



Hardness は、直訳で考えれば線の硬さ、になりますが、
1以下にすると、線の輪郭がぼやけ、その分細くなっていくような印象になります。
半透明が絡む為多少扱いが難しく、図のように線が交差する場所で背後の線を(薄い部分であっても)隠してしま

す。
それを防ぐために、マテリアルの線の設定で、Self Overlap(セルフオーバーラップ)を有効にすることができます。
ただし、線が重なる場所は重なって表示してしまうので、線の太さがある程度以上太いと、今度は太い線同士が重なり、ゴミが乗り始めます。

半透明がからんで線の描画がうまくいかない場合には、線で描画するのではなく、点描の設定にする方法を使う手があります。それについては、[後のセクション](#)で軽く述べます。



このほか、縦横を不均一にしてカリグラフィーペンのような効果を与えるパラメータもあります。

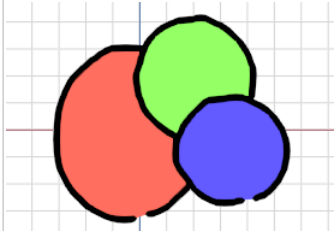
他にも手振り補正や、自動スムーズ化といったパラメータもあります。

全部説明するのは本書の範囲外だと思うので、気になる方は[マニュアル等](#)を確認してみてください。

線を描く。レイヤー内の上下関係

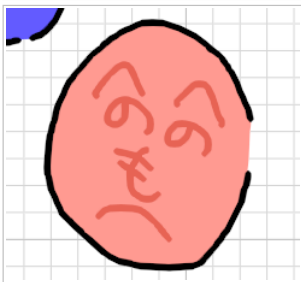
レイヤーごとの表示の上下関係は、すでに確認しました。

次に、同じレイヤーでの上下関係ではどうなるのかチェックしておきます。



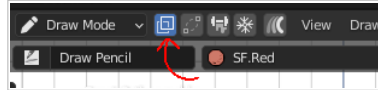
予想通りですが、後から描いたものほど上位に表示されます。

しかし、時には下に描き込みをしたい時があります。



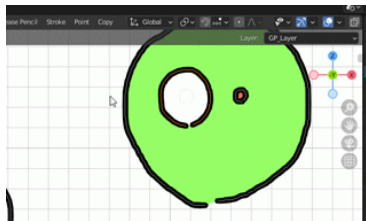
顔を描いてしまった後で、色付きの輪郭を付け足したい時などです。

最初に書いた顔の下に描き込みをする場合の設定はこのアイコンです。



新たな描画線が、既にある線の下側に回り込みます。

(動画 Order01.gif) *pdfでは最初の1コマが表示されています



ちなみに、一旦描き込んだ後のストロークの前後関係は、編集モードの Stroke(ストローク) - Arrange(整列) で変更できます。

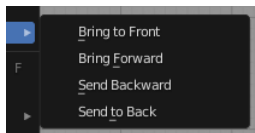
(動画 Order02.gif) *pdfでは最初の1コマが表示されています

「整列」の訳が多少しっくりこない(縦や横に並べる系の操作を連想してしまう) 点が注意ですね。

前後の移動は、一気に最前面最後に移動するものと、一段階ずつ移動するものがあります。

自分は基本UIは英語で使っているのですが…

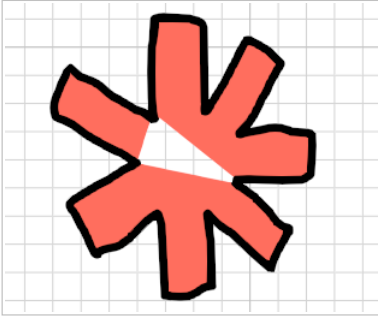
メニューの字面だけだと多少混乱するので、前から後ろに順番に並んでいる、と記憶してしまった方が良いでしょう。



Bring To Front	最前面に移動
Bring Forward	前に移動
Send Backward	後ろに移動
Send To Back	最後に移動

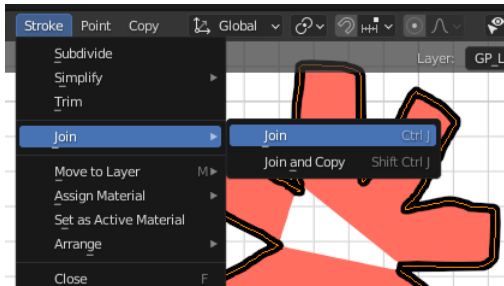
Send Backward と Send to Back がどっちがどっちなのか、字面では非ネイティブにはとっさに判断ができないので…

線をつなぐ。長い塗りつぶしの線



長い線や複雑な線で塗りつぶしをしたい場合があります。
線だけの表示をする場合には、多少繋がってなくても見た目につながっていればO.K.なのですが…

塗りつぶしのマテリアルで処理をしたい場合には、実際に線が繋がっていないといけません。
長い線を描きながらいるような人でなければ、手描きで長いストロークを一度に作るのはなかなか難易度が高かったりします。

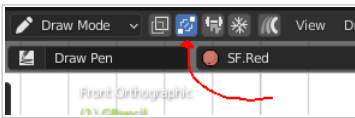


編集モードで後からつなぐ場合には、Stroke から Join(統合)を使用します。

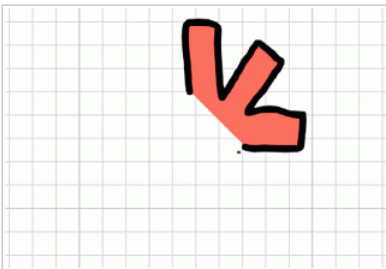
この操作は線の向きに依存するので、筆順が途中で逆になっていたりすると、線の頭ではなくお尻に繋がってしまったりします。

そのため、まずSwitch Direction(向きを反転)で筆順を揃えてから…となるので結構処理が面倒です。

描画中に、休み休み筆を止めながら描いた線を、連続な線にするモードがあります。



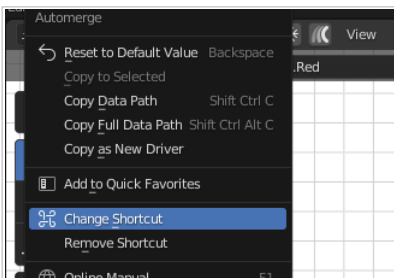
このボタン(Automerge, 自動マージ)です。



終点近くから描画を開始すると、その線とつないで線の描画を開始します。

(動画 Connect01.gif) *pdfでは最初の1コマが表示されています

これで、長い線を一気に描くのが苦手な場合でも、だいぶ楽になります。
ただ、込み合った場所ではあっちこっちの線にくっついてしまう点だけが注意が必要でしょう。



実は、こうした設定ON/OFFのトグルスイッチも、右クリックメニューからの操作でショートカットの設定をすることができます。

あからさまにショートカットを設定できそうな実行ボタンや、メニュー (ショートカットが表示される)の項目と違って、盲点ではあります。

(自分はだいぶ後からこれに気が付きました…)

これらの設定を何度も切り替えて使うようでしたら、ショートカットに登録することを考えてみるのも良いかと思います。

簡単なお絵かき

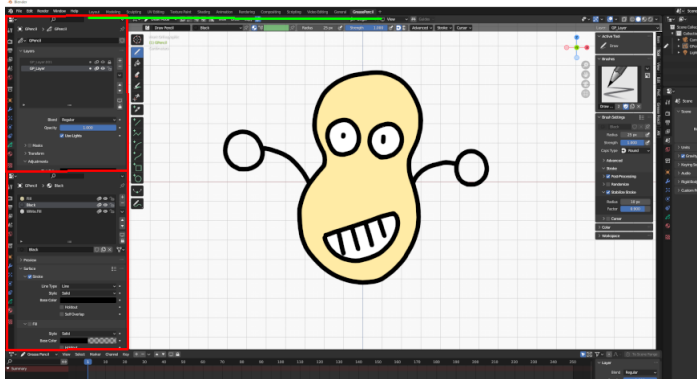
ここで一息入れて、いままで見てきた機能で簡単なお絵かきをしてみます。

ここまでの内容でレイヤー分けが重要なこと、色分けにはマテリアルを複数使うことが分かりました。

そのため、お絵かき用にはアウトライナーを2つ開いて、データ表示タブとマテリアル表示タブが同時に画面内にあるのが望ましいです。

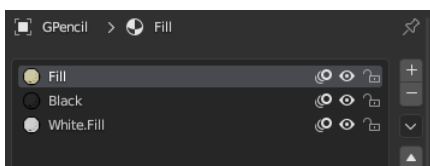
レイアウトは完全に個人の好み次第なのですが、アウトライナーを2つ使いやすい場所に開き、

データタブと、マテリアルタブを開いてそのレイアウトを登録しておくのがお勧めです。

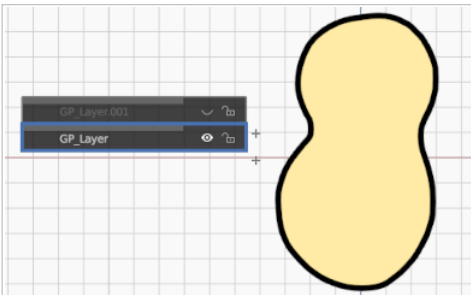


画面最上部のワークスペース用のタブ(緑の線)を使い、グリースペシル用のレイアウトを設定して、スタートアップファイルとして保存しておくのがお勧めです。(もちろん既存の2Dアニメーション等の設定を使っても良いわけです)

オブジェクト選択用にもアウトライナーや、プロパティのウィンドウも開いておきたいところですので、取捨選択を慎重に考えて、使いやすいレイアウトにします



上の瓢箪みたいなキャラクター（そこはかたなく見覚えのあるような…）のために、線だけのマテリアル、線と塗りつぶしのあるマテリアルを合わせて3つ用意しました。

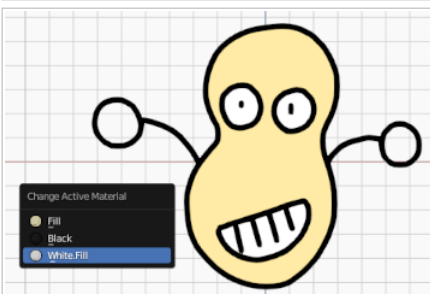


レイヤーは2枚用意をしました。

下のレイヤーには、基本の瓢箪型だけ書いて、顔や手は上のレイヤーに表示するようにしてみます。

滑らかな線になるように、スムーズ化などの設定も自分の使いやすいパラメーターなどを確認しながら描画します。

今回のようなトゥーンめいた線であれば、線を閉じるところで、Automergeの機能のON/OFFをうまく使って綺麗に線を閉じたいところです。



上のレイヤーに顔や手を書きこみます。

白目や口の部分は、まず「実線+白で塗りつぶし」のマテリアルを使って描き、その上に黒の実線で黒目や歯を書きこみました。

マテリアルを選ぶには、ショートカットの「U」も使えます。

簡単なお絵かきでグリースペシルの描画に慣れてきたら、先に進みます。

レイヤーとマテリアルの切り替えは特に重要なので、ショートカットを覚えてしまうのが良いと思います。

Y	レイヤーの切り替え
U	マテリアルの切り替え

塗りつぶし

バケツツールで塗りつぶし

バケツツール、正式名称ではないのですが、アイコンがバケツであることが多いので、よく使われる言葉です。塗りつぶしツールともよく呼ばれます。

Blender でのツール名は Fill(フィル)ツールです。

先ほど、[レイヤー内の上下関係](#)の確認をしましたが、1レイヤーでもバケツツールを駆使して線画の塗りつぶしをすることができます。

しかし、多くの場合は、調整が大変になってしまうので、実線用のレイヤーと、塗りつぶし用のレイヤーを分けるのが無難だと思います。



バケツツールにも設定が数多くあります。基本の設定は、ここでは省略しています。

詳細設定の中で特に重要そうな物だけ確認をしておきます。

Layers(レイヤー) で、境界判定に使うレイヤーを指定します。

見えているレイヤーを(全て)使ったり、今選択している(Activeな)レイヤーだけを使ったりすることができます。

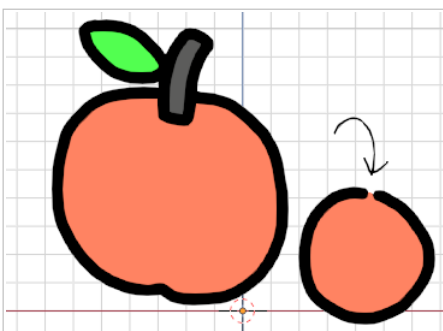
Layer Above(上のレイヤー)などの設定もあるので、塗りレイヤーの上に実線レイヤーがあるような場合には、これを使っても良いかもしれません。

また、バケツ塗りに重要な特性としては、塗りが漏れるかどうかがあります。多少の穴が開いていても漏れずに塗りつぶしてくれるかどうかはとても重要です。

塗り漏れ防止機能の設定は以前は Leak Size(リークサイズ) という名前でしたが、Blender 3.4 以降アルゴリズムが変更されています。

現在は **Gap Closure** の **Size(サイズ)** というパラメーターになっています。

この値が大きいほど、塗りが漏れなくなりますが、当然細かな線が入り組んでいるような場合には、塗り残しが増えるようになります。

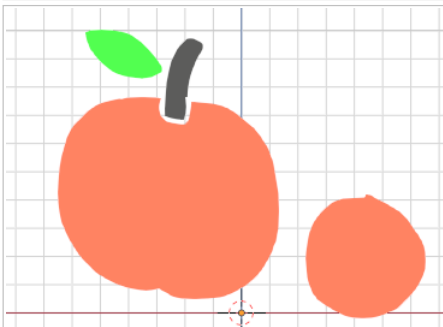


線画で描かれた絵をバケツツールで塗りつぶします。

適切に Gap Closure Size を指定することにより、多少閉じていない線でも漏れずに塗りつぶすことができます。

また、線の隙間から漏れてしまった場合、塗りつぶしが行われません。

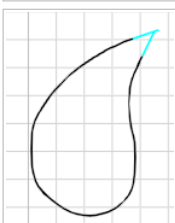
「画像の端」という概念がないので、画面端まで塗りつぶすということにならないのです。これは、Limit to Viewport(ビューポートに制限)を有効にしても同様です。



線画を隠すと、どのように塗られているかが分かります。

この例では実線を太く使っていますが、塗りの間に隙間が空いていて、実線の太さを考慮した塗りつぶしになっていることがわかります。

これは、**Boundary(境界)**の設定で変えることができ、Edit Lines(編集ライン)の設定にすると、線の太さとは関係のない、編集用の線の位置そのもので塗りつぶしの判定が行われます。(その分漏れが発生しやすくなる点は注意です。)

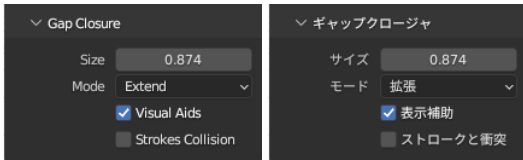


塗りがはみ出る現象を防ぐのが、Gap Closure です。

Visual Aids(表示補助)の設定を有効にすると、どのようにはみ出しを防いでいるのかが見ることができます。

水色の線が延長線として表示され、そこが境界として扱われます。

尖った毛先が抜けているような、そういう線でも漏れずに塗りをするようになります。



この補助線は、単純に延長線を伸ばす Extend (拡張) と、周囲を探索して近くの線端とを結ぶ Radius(半径)モードがあります。



単純な図形でモードを比べてみると、このように補助線の引き方に違いが出ます。経験上どちらかというと Radius の方が塗り漏れが少ないですが、その分塗り残しが増える印象です。

Visual Aids(表示補助)が有効な状態だと、最初のクリックで補助線が表示され、次のクリックで塗りが実行されるようになります。



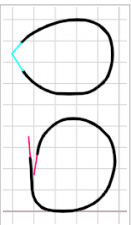
この機能は線が途切れている場所が多数あるような領域を塗るときに、大きな力を発揮します。Radius モードだと、探索半径に線端が見つからない場合は赤い丸が表示されて、パラメーターの目安にすることが出来ます。Extend モードでは、Stroke Collision(ストロークと衝突)を有効にすると、延長線が何にもぶつからない場合に赤く表示されます。

Size パラメーターは、ホイール操作でGUIから調整することもできます。

(動画 GapClosure01.gif) *pdfでは最初の1コマが表示されています

この例では、Size のパラメータを適切に設定することで、線の途切れの大半は対応できました。

※)本来、塗り漏れが起きると塗りは実行されませんが、分かりづらいのでここでは画面外に四角い枠を配置し、塗りがその範囲に収まるようにして実行しています。



Extend モードで **Stroke Collision(ストロークと衝突)** を有効にすると、補助線と線 (または補助線同士) がぶつからないような場合に線が赤く表示されるようになります。

線が互い違いにずれるなるなどして、上手く塗れない場所を特定するような場合に役立ちます

境界ストローク

閉じていない線の内側を、漏れずにうまく塗りつぶすのはなかなか大変で、自動では無理な場合もあります。

そういう時に、手動で「一時的な」境界線を引いてそれを使って塗りつぶす方法が用意されています。

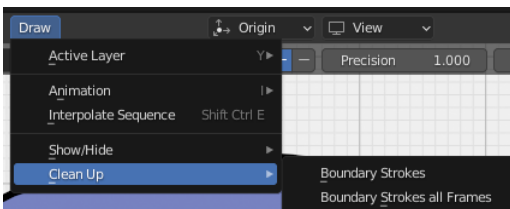
(この線は、Boundary Stroke(境界ストローク)と名前が付いていて、後から一括で削除ができるようになっています。)



境界ストロークは、ツールを変えずにパケツツールのままで線が引けるというのがメリットです。パケツツールのまま Alt + マウス左ドラッグで線を引き、そのまま線を利用して塗りつぶせます。漏れを防ぐために使うだけではなく、色トレス線の代わりに実線の無い輪郭線を描くときに使っても良いでしょう。

(動画 BoundaryStroke01.gif) *pdfでは最初の1コマが表示されています

(レンダリング時には表示されません。プレビューでレンダリング設定を使っている時は注意です)

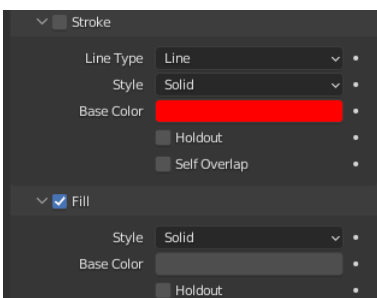


境界ストロークは、必要がなくなったらメニューから一度に削除できます。

モードでメニュー位置が変わるので多少注意です。

編集モード : Grease Pencil - Clean Up

ドローモード : Draw - Clean Up



境界ストロークの表示色は、ストローク用のカラーです。この例では、赤を使って分かりやすく区別ができるようにしています。(カラー属性は、塗りのみに適応されるようにして、赤の色が残る工夫をしました)

特別に境界ストロークを区別する色などは無いようなので、ストローク用のマテリアルを使って描いてしまうと、見た目でも普通の線との区別がつかなくなってしまうので、やや注意点です。

(境界ストロークは、レンダリングでは表示されないの、モード切り替えで区別する手はありますが…)



Alt + 左ドラッグなので、Emulate 3 Button Mouse モードでの操作をしていると、操作が衝突してしまってこの機能が使えません。

もし必要であれば、設定を Alt との併用から変えて使用します。

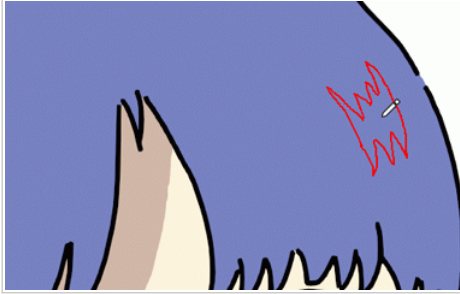
Preference (プリファレンス) 設定のキーマップから変更ができます。

Grease Pencil - Grease Pencil Stroke Paint(Fill)-Grease Pencil Draw で (Fill)の項目の中の4つ目にあります。

このあたりは名前が似ていたり、同名のオプション違いの操作が並んでいるので、少し探すのは大変です。

バケツ塗りではない塗り

線描用の材料ではない塗りつぶし用の材料でも、描画ツールをドローツールに変更すれば、普通に描きこみができます。ですから、ツールを切り替えればバケツ塗り、普通の手書きでの塗り…と切り替えられますが…少し面倒です。



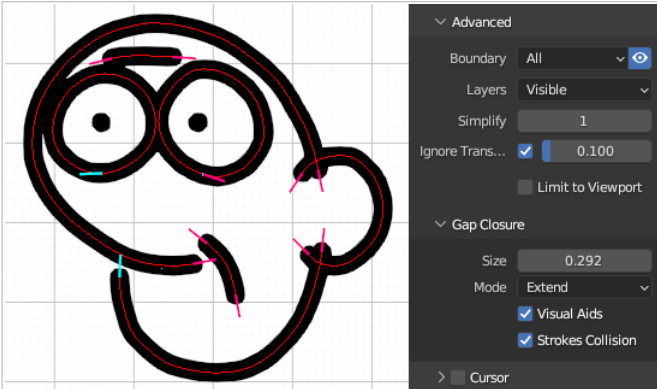
描画ツールの変更が面倒な場合に、バケツツールの状態のままで同じことができます。「境界ストロークで色トレス用の線を描いて塗りつぶす」という手順も考えられますが、もう一つ、Shift キーを押しながら左マウスボタンで、直接の描きこみができます。バケツ塗りの作業の途中で、同じ色で単純な描きこみもしたい時に役立ちます。

こうしたバケツツール使用時の直接の描画やBoundary Stroke に関する補足説明を [最後のTIPSの章に](#) 追記しました。

境界の種類と表示

先ほどリンゴの絵で、塗りつぶしの境界の設定の違いを確認しました。

その後、塗り漏れを防ぐ方法をいくつか確認したので、もう少しだけ境界の設定について詳しく見てみます。

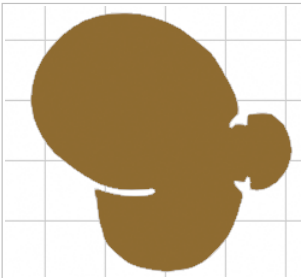


Boundary(境界)の設定のところにもよく見ると目のマークがついています。これを有効にすると、塗りつぶしの際に Edit Lines(編集線)、つまり線の太さに関係なく、線の真ん中のラインが赤で表示されます。

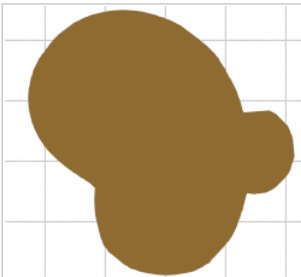
Gap Closure の Visual Aids (表示補助) と Stroke Collision(ストロークと衝突)の表示も有効にしておく、赤と水色で実際に塗り分けに使われるラインが表示されることになります。

Boundary(境界)の設定は、左の様に太い線で描かれたイラスト的な絵の塗りつぶしで、意外に重要な場合があります。

上のおじいさん(?)の図はよく見ると、下唇のところで実線の途切れがあります。わざと塗りのはみ出しが起りやすくしています。



Boundary(境界)の設定をストロークにして、つまり線の太さの効果も考慮した塗りつぶしを実行します。実線を非表示にすると、口のラインや、耳の線が入り込んでいる部分が塗られていないことが分かります。



Boundary(境界)の設定を Edit Line(編集線)にして、線の太さに関係なく塗り潰しを実行した場合です。先ほどの図の、赤い線と水色の線が塗りの境界線になります。

線を非表示にすると、口の部分の塗り残しがありません。塗りつぶし領域の中に入り込んだ口や耳の部分の線は(太さが無いので)無視されていることが分かります。



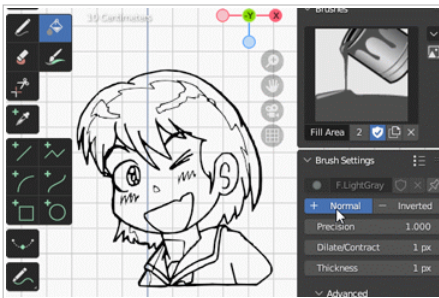
実線を表示してしまえば、どちらのモードでも塗りつぶしても見た目に差は無いのですが、特に太い線のイラストで、実線が塗りの領域に入り込むような場合には、塗りの結果や、塗りのはみ出しを防ぐ設定の効き方が違ってくるので、多少気にした方がよいかもかもしれません。

(このおじいさんの形は、口の部分で違いがはっきりと分かるような線の形になるように描いた図形です。くると巻いて内側に入り込むような線が、要注意ということですね。)

反転フィル

簡単な線画を塗りつぶすのであれば、順番に塗ればよいだけで簡単ですが、複雑な絵をグリスペンシルで描くようになると、細かなパーツを順に塗っていくのが手間になってきます。

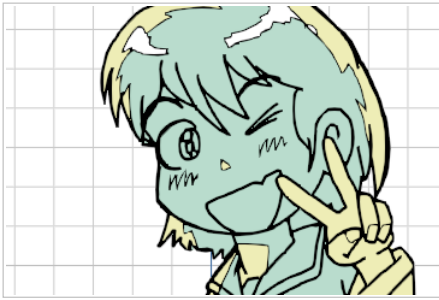
単一色の塗りなどの用途に使える、面白い機能があります。



塗りの設定をInverted(反転)にして、閉じた線の「外部」で塗りを実行すると、逆転して閉じた領域全てをひとまとめに塗りつぶします。

形状は細かいけれども、ほぼ単色の塗りが必要な場合に非常に有効です。

(動画 InvertedFilling01.gif) *pdfでは最初の1コマが表示されています

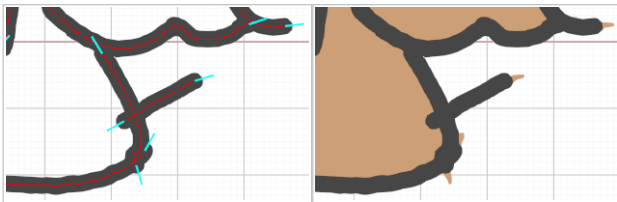


ショートカット操作の場合は、Ctrl + クリックで、反転フィルになります

反転フィルは、領域が全部単色で塗られるので、複数の色を使う場合は、基本色を塗った上にさらに通常の塗りを重ねて表現することになります。

ほぼ単色で、多少の色乗っている、というような時に使いやすい機能です。

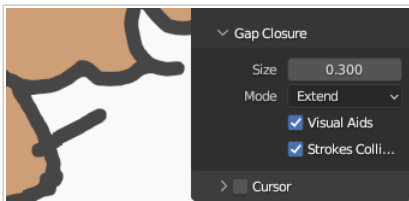
また「小さなパーツの領域が塗り漏れで透明になる」という現象を防ぐ下塗りとして使う手もあるかと思えます。



Inverted Fill は、3.6 時点では設定によっては Gap Closure の影響が強くなります。

その場合はゴミが乗ってしまうので、使うのが多少難しいかもしれません。

Gap Closure Size を 0 にすれば影響しませんが、線をきちんと閉じないと塗れなくなってしまいます。

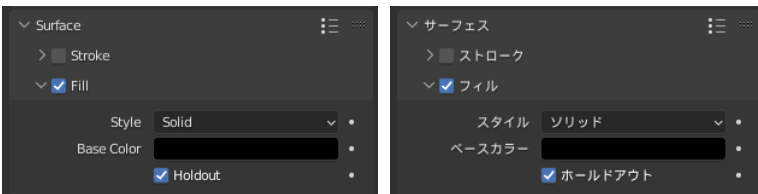


私の確認したかぎり (3.6 現時点では) Extend モードで Stroke Collision を有効にした場合に外部のゴミが発生しないようです。

これは延長線の有効/無効を判定して色分けをする機能ですが、同時に無効な延長線を処理時に使わないようにしているのかもしれない。

中空の塗り。ホールドアウトとマスク

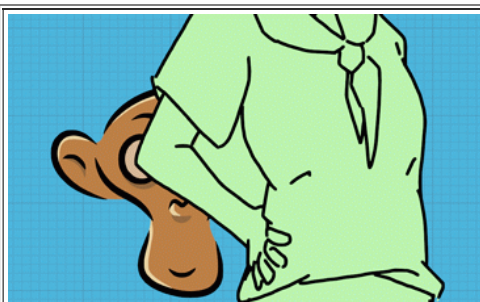
塗りつぶしの処理が、ある閉じた線の内側を塗りつぶす、ということというのは、ドーナツのように中空の形状を塗ることができないことになります。それでは色々困りますから、塗りの中に空洞を開ける方法が2つほど用意されています。



ひとつは、**Holdout(ホールドアウト)**というオプションを有効にしたマテリアルを使って、中空部分を上書きすることです。このオプションを有効にしたマテリアルは、その下のマテリアルを透明化する機能が付きまます。

反転フィルを使って、輪郭線の内部を単色で塗りつぶすと、下の図の脇の下のような中空の部分まで塗られてしまいます。

ホールドアウトで透明にしてみます。



線のレイヤーと、塗りのレイヤーの間にもう一つレイヤーを作成し、そこに中空部分を描き込んでみました。

ホールドアウトを有効にしたマテリアルで、穴を開けたい部分にストロークを配置します。分かりやすくするために塗りを使わずに、手描きで線をなぞってストロークを作成しています。通常はバケツ塗りを使うことになるでしょう。

(動画 HoldoutFilling01.gif) *pdfでは最初の1コマが表示されています

レイヤー分けは必ず必要なわけではなく、描画順がホールドアウトのマテリアルで上書きをするようになっていれば、そこが透明になります。

ホールドアウト用のマテリアルは、普通は色は黒にしておきます。

黒以外の色がついていると、完全に透明にはならず、その色味が付きまます。

もう一つの方法は、**マスク機能**を使うことです。

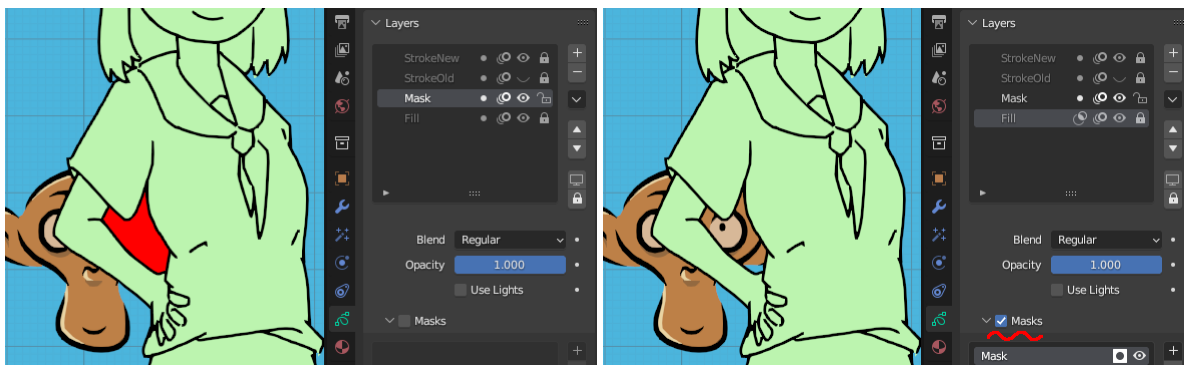
マスクとして設定したレイヤーを（または、その反転を）マスクとして隠すようになります。

抜きたい中空部分をマスク用レイヤー(Mask)として、赤く塗りつぶしました。

Fillのレイヤーのマスク設定を有効にして Maskレイヤーを設定します。

デフォルトでは、Mask レイヤーで塗られている部分だけが表示されます。

(この場合は、空白にしたい部分を塗ってあるので、反転設定をします)



ただし、それだけだと Mask レイヤーそのものも表示されるのでやっぱり赤い領域が残ってしまいます。

Mask レイヤーの Opacity(不透明度) 0 に設定することで、透明な抜きをつくるマスクになります。

(設定忘れが多そうな部分なので強調です)

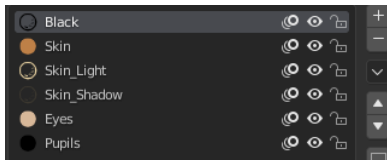
Holdout とマスク機能、同じことをしているように思えますが、

使い分けが必要な場合はあるのでしょうか？

	<p>実は、Holdout は、影響が下のレイヤーまで及びます。 左の例では文字盤の上に文字レイヤーを作り、文字の穴の部分 Holdout を使って穴をあけ…ようにしましたが、こうなっていました。 Holdout の効果で下のレイヤーの文字盤までくりぬいてしまっています。 このような場合には、レイヤー構造が増えて多少ややこしくなってしまいますが、Mask 用のレイヤーを駆使しないといけないことになります。</p> <p>(逆に「レイヤーを組んだ絵をまとめて大穴を開けたい」というようなときが Hold out を使うべき時ですね)</p>
	<p>Mask機能は自分自身の下にあるレイヤーを使うこともできます。 汚れ用のレイヤーを上置いて、はみ出しO.K.で気楽に汚れを塗って、その後下のレイヤーをマスクに使うことで、はみだしを無くす、という使い方もできます。 (動画 Mask01.gif) *pdfでは最初の1コマが表示されています (この汚れのレイヤーのように、カチツとした実線や塗りつぶしではなく、ブラシで塗ったような効果については後述します)</p> <p>こうして見ると、Maskのほうがきめの細かい制御ができる代わりに、レイヤー数が増えたり設定が煩雑であると言えそうです。 Holdoutで済むような場合に、Holdoutを使うというのがよさそうです。</p>

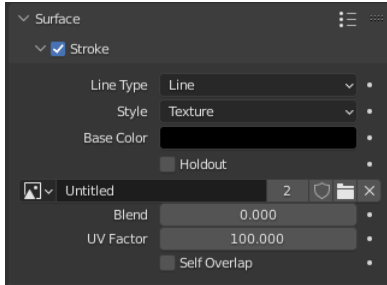
テクスチャを使う

グリースペンスリルは、単色の表示だけではなくて、画像テクスチャを使った貼り込み表示を行うことができます。テクスチャを貼るときの特性を確認してみましょう。

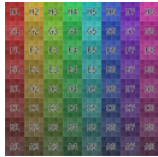


ここでもサンプルのスザンヌを使って見ます。

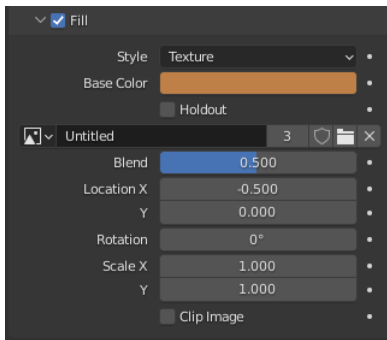
実線を表示してる Black と 塗りつぶし用の Skin のマテリアルで、表示タイプを変更します。



Strokeのスタイルを Texture にして使用するテクスチャを設定します。新規に Color Grid の画像を作成してそれを使用しました。

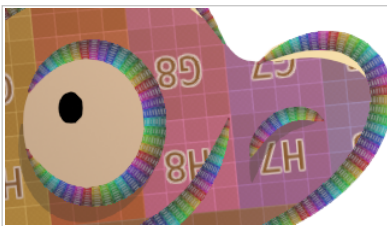


(名前は特に設定しなかったため、デフォルトのUntitledです)



同様に Skin の Fill(フィル)設定も Texture 使用にして同じくColor Gridのテクスチャを使用します。Blend の設定などで、基本色との混ぜ具合などを調整します。

(光源の向き次第で見づらくならないように、Layers(レイヤー)側の Use Lights(ライトを使用)設定もOffにして調整した方が、画像を使った確認には良さそうです)



線や塗りつぶし部分に UV が貼られていることが分かります。線の部分は線に沿っていますが、塗りの部分は妙な向きを向いています。

UVの貼られ方のルールを把握しておかないと自在に使うのは難しそうです。

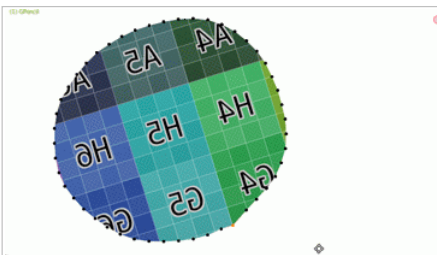


線を拡大してみると、線の幅に応じてV方向0-1になるように貼られているようです。U方向のスケールは、UV Factor の設定を変えると変化します。このあたりは、直感的にも理解しやすい部分です。

ただ、塗りつぶしのテクスチャの向きは、なぜ最初に斜めにひっくり返っているのか、理由がいまいち分かりません。

手描きの円などを描いて、点の間隔が適度に開いているようにして編集してみると理由が見えてきます。

Stroke - Simplify(簡略化) - Sample(サンプル) で点の数を減らして、個々の点を分かりやすくします。また、Select - Select First (最初を選択) を使うことで、ストロークの最初の点の位置が分かります。



この最初の点を動かしてみると...

UVが、最初の点と次の点の位置関係で決まっていることが分かります。

スザンヌの本体も、ストロークの最初の位置関係に応じて、テクスチャが斜めに張られているのでした。

(このUVは、グラデーション設定の時にも、向きを決めるのに使われます)

(動画 UV01.gif) *pdfでは最初の1コマが表示されています

多少厄介なのは、テクスチャの向きを揃えたい、というような場合やアニメーションなどで最初の2点の位置関係が大きく変わるような変化をさせたいような場合です。

そのようなときは、UVの向きが大きく動いてしまうので、テクスチャの向きを揃えるのが難しくなってしまいます。

最初の2点を縦や横に揃えて、強さ(濃さ)0にして見えなす、といった調整をしたり、

「あまり動かない場所」が最初の2点になるように、描きだす前に少し考える必要があるかもしれません。

ストロークのテクスチャ座標編集

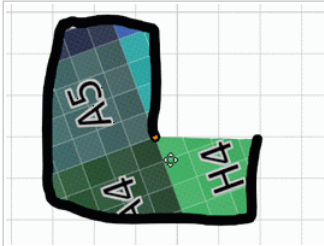
最初の2点で位置関係でUVの向きが決まるようなのですが、先ほどの例だと裏返しになっていました。



通常の間を描くような感じでの描き方（下側から描きはじめ）の場合、反時計回りに描くと通常向きのテクスチャになります。

(動画 TexCoord01.gif) *pdfでは最初の1コマが表示されています

考えてみれば、最初の2点の位置情報だけでは表か裏かの情報は無いので、表裏が逆になる代わりに、180度ひっくり返った表示になってもおさそうです。そこで、もう少し詳しく挙動を調べてみると…



先頭の頂点の位置を動かしていくと、やはりテクスチャが裏返る角度がありました。この図では90度や270度を挟むと、そこでテクスチャの向きが反転するようですね。

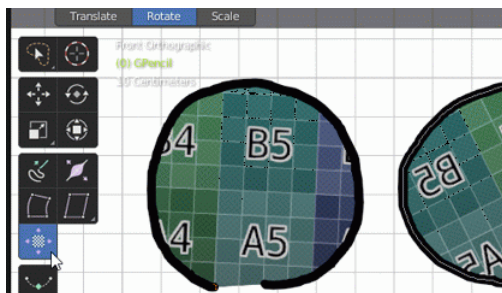
(動画 TexCoord02.gif) *pdfでは最初の1コマが表示されています

※)形状によっては、45度や135度などで裏返ります。裏返る角度の条件は完全には把握できませんでした。



裏返る位置関係のストローク用には、スケールを反転させたマテリアルを使います。

さて、癖があって多少分かりにくいUV座標なのですが、頂点位置と数値入力で調整するのはかなり厳しいので、GUI上で操作するツールは用意されています。



エディットモードでのTransform Fill(フィルトランスフォーム)で、Translate, Rotate, Scale の3つの要素についてGUI操作ができます。（回転の最初の起点はやはり最初の頂点の位置になっていますね）

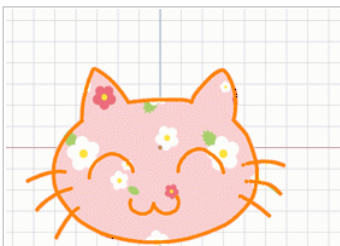
(動画 TexCoord03.gif) *pdfでは最初の1コマが表示されています

この操作自体が、ビューから見た座標ではなく、デフォルトのUV座標上での変形になっているので、操作はしにくい（移動の向きが見た目と合わない）のですが、それでもだいぶ直感的な操作になっています。

こうしてみると、質感として使うのでテクスチャの向きがあまり影響しない、というような場合は良いのですが、模様として使いたいの向きが重要、というような場合にはちょっと使いづらい仕様になっているのが分かります。

きっちり固定したテクスチャを使いたい場合は、まっすぐ配置するために四角の上などに貼ったテクスチャのレイヤーを作り、[マスク機能](#)を使って表示するのが良いかもしれません。

メインのレイヤーの上に、パターン用のレイヤーを作って重ねて表示をします。



花柄模様のテクスチャを表示するレイヤーを作り、その下にネコちゃんを描いたレイヤーを作成しました。ネコの形で花柄がマスクされるようにしています。

(動画 TexCoord04.gif) *pdfでは最初の1コマが表示されています

ただ、当然花柄のレイヤーを上置いて、マスクを設定しただけだと、猫の形の花柄模様だけになってしまいます…



ネコちゃんは白黒（白い塗りつぶしと黒い線）で描いておいて、花柄のレイヤーを乗算で上から重ねる構造にすることで、右のような輪郭付きで花柄模様の猫として表示されました。



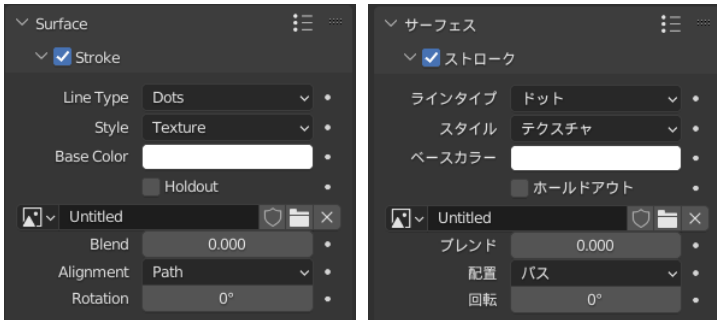
これで、模様レイヤーを固定したままネコちゃんレイヤーを動かせば、模様固定のまま形で動かすアニメーションも作れます。(ただ、普通にオブジェクトとして動かすと、全レイヤーが動くので模様とネコちゃんをバラバラにした意味がありません。) 後述するキーフレーム間補間を使って、ネコちゃんだけを動かしたアニメーションを、作例にしました。



この例は、UVAndMask.blend として同封しました。
(動画 TexCoord05.gif *pdfでは最初の1コマが表示されています)

点描とテクスチャマッピング

線と塗りを使って描く場合を見てきましたが、他に重要なスタイルとして点による描画があります。ストロークのタイプを Dots(ドット)にすることで、コントロールポイントを結んだ線ではなく、点の集合体として描画をすることができます。



半透明と点描

先のセクションで少し述べたように、線で描画する場合、ある程度太い線は半透明を使った描画と相性が悪いです。そういう場合は、点の連なりとして線を引きこのモードが解決策になる場合があります。

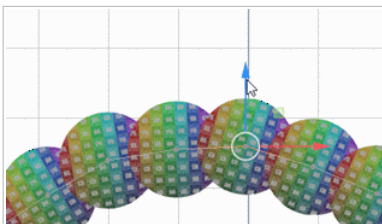


太い線で描いた 半透明の絡んだ Line の線は、Self Overlap を有効にすると、細かいゴミが乗ることが多いです。(かといってSelf Overlapを切ると、交差した線で片方の線が隠れてしまいます)

Dots による点の集合体として描画することで、問題が解決することがあります。ただし、点が重なる部分はどんどん濃くなっていくので、コントロールポイントの密度に濃さが依存するようになる点に注意してください。Lineとしての描画をするときは、点から点へ結ぶように線を引いていたので、コントロールポイントの密度は濃さに関係しませんでした。

点描のテクスチャマッピング

さて、「線を点の集合体として描く」という機能なのですが、その特徴から、線なのに塗りのような使い方をすることもできます。そうした使い方をするには、テクスチャを使用するマテリアルとの相性が非常に良いです。

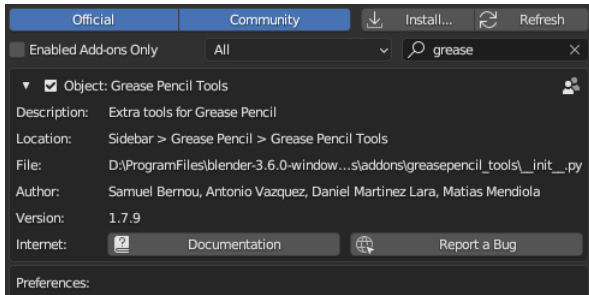


頂点の間隔と点のサイズ(線の太さ)を大きく調整してみれば、仕組みは一目瞭然でしょう。線として描いた場合と違って、一点一点に元のテクスチャを貼った丸を配置します。(丸くではなく四角にする場合は Squares(四角形)を選択します)

Alignment(配置)が「パス」だと線に沿った向きに回転します。この回転をさせたくない場合は、パス以外のモードにしておきます。
(動画 Dots01.gif *pdfでは最初の1コマが表示されています)

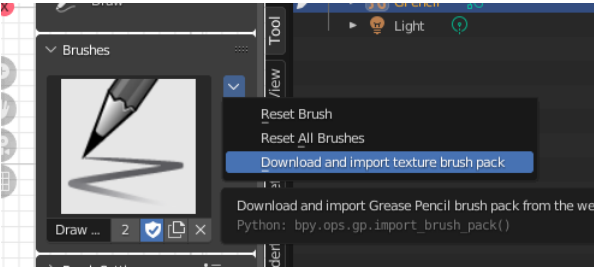
使い道として筆のストローク感をテクスチャで実現したり、木の葉や草などのパターンを配置する用途が典型的です。

自分でテクスチャを作成して試しても良いのですが、デフォルトアドオンの中に、そうしたブラシ設定が用意されているので、それを軽く見てみましょう。

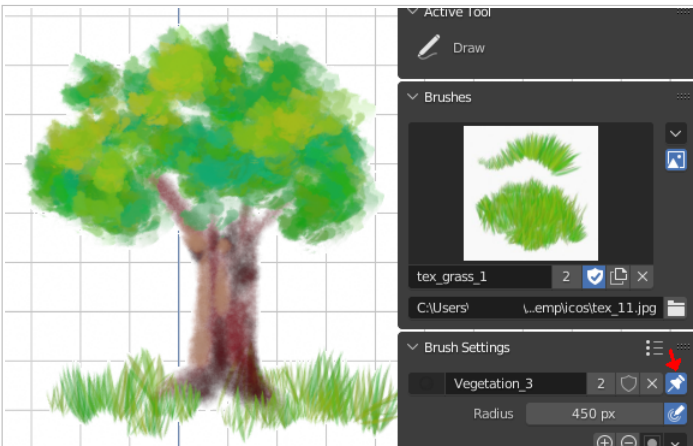


アドオンの Grease Pencil Tools を有効にします。

ブラシのメニュー（アイコンの右に表示されるボタン）の中に、アドオンで用意したブラシを設定する項目が出現します。これを実行すると複数のブラシが追加されて使用可能になります。



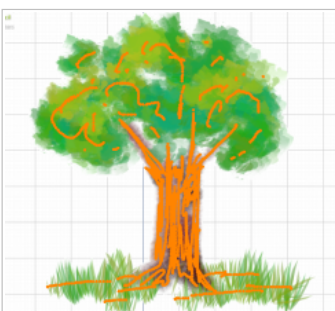
幾つかのブラシを組み合わせれば、このような風味の絵をかなり楽に描くことができます。興味があればパラメータを見て、仕組みを確認すると良いでしょう。



ブラシのセッティングとして、点のサイズや回転などの、ランダム分布といった配置の設定がされています。また、ブラシの材料の設定の部分でピン止めのアイコンが有効になっています。このピン止め設定をすると、材料がブラシに関連づけられるようになります。この草用のブラシの場合はVegetation_3が関連付けられています。

材料の Vegetation_3 では、使用するテクスチャなどの設定がされています。ピン止めが有効になっているので、今までの一般的なブラシとは違って、別の材料を選択していても、必ず草が描かれることとなります。

ピン止めの設定は、「ある材料用のブラシ」を作ったときに、ブラシを変えるごとに「ブラシと材料の設定をそれぞれで選択しないといけない」という事態を防ぐための仕組みですね



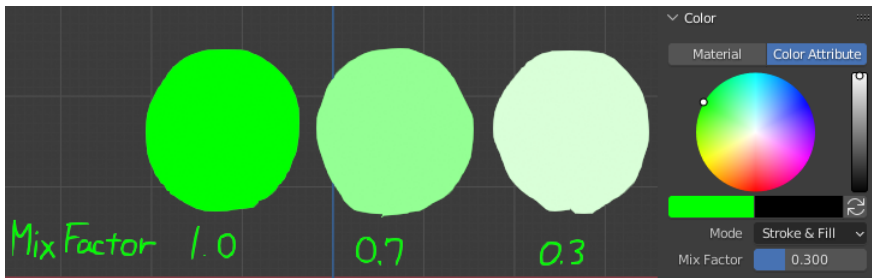
編集モードで、実際の線と頂点を表示させると、随分と少ない線（点）によってこの絵が作られていることが分かります。ただし、この絵をよく見てみれば、塗られているテクスチャには影色や日に当たって明るいところに相当する色のバリエーションがあることが分かります。

今まで作ってきた材料は、テクスチャを除けば単色の色設定だけ持っていました。（単色の他に2色グラデーションのモードもあるのですが、それでも2色です）その材料をそのまま使っても、線ごとに色を変えることはできません。

材料ごとに設定された固有の色の他に、ストロークの各頂点に色情報を持たせて、そちらを表示に使うことができます。メッシュの3Dモデルでカラー属性を使った色を付けるのにほぼ対応する手法です。次のセクションで簡単に述べておきます。

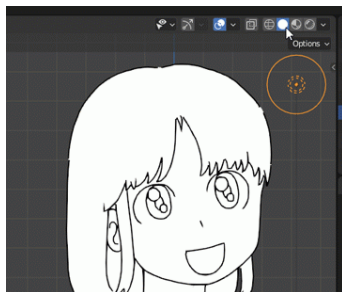
カラー属性

マテリアルの固有色ではなく、頂点ごとに色を割り当て、その色を基準に表示をすることができます。
blender 3.1 までは頂点カラー(Vertex Color)と呼ばれていましたが、3.2 以降ではカラー属性(Color Attribute)と呼ぶようになっています。
仕組み自体は単純なのですが、どちらの色を表示するのか、どのパラメーターで制御するのか、
といった設定がやや煩雑なので、多少とっつきにくい印象があるので、軽くおさらいをします。



ストロークを引いている時に、カラーの設定を Color Attribute にすることで頂点に色を設定します。
この時に色と一緒に設定しておく Mix Factor が重要で、1の時がカラー属性の色、0の時はマテリアルの固有色になります。
この図は、マテリアル固有色が白、カラー属性が緑の時の表示です。

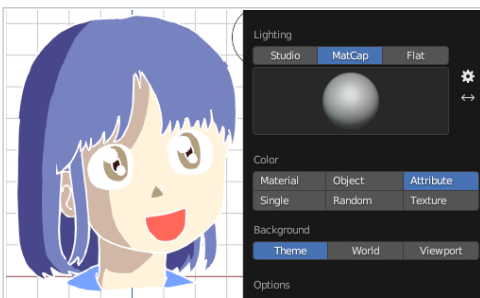
マテリアルを色ごとに用意する、という運用だとマテリアルが増えすぎてやりづらい場合には、
単色のマテリアルを1つ用意して、色は主にカラー属性で設定するというやり方もあります。
ただ、その場合には画面の表示モードにも気を付ける必要があります。



⚠ デフォルト設定時にカラー属性の色が表示されるのは、ビューポートの表示が、「マテリアル」もしくは「レンダー」時だけです。
ソリッド表示の時は、マテリアルの固有色の表示になるので注意が必要です。
(動画 VColorPreview01.gif) *pdfでは最初の1コマが表示されています

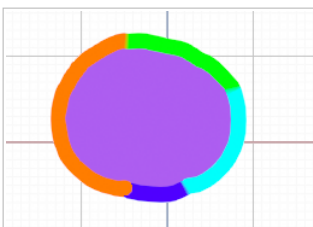
(ちゃんと色情報を塗っているのに、あれ？色が無い？と焦りやすいのです)

マテリアルモードの時は、(マテリアルモードのオプションがデフォルトのまま Scene Lights がオフになっていると)
光源に反応しないで素の色が表示されます。
アニメ絵のような単色塗りの場合、指定したそのままの色が出るので逆に好都合、ということがありますが、
いざレンダリングをしたら光源との関係で全然違う色になってビックリすることがあります。
(そういう場合は、[グリースペンシルの Use Lights を無効](#)にすれば良いわけです)



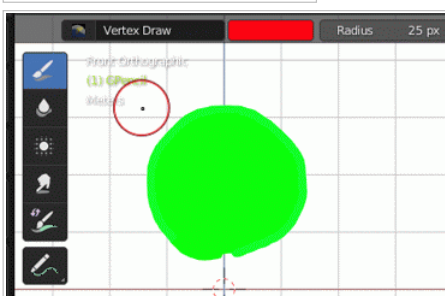
ビューポート表示でも、表示色の設定を Attribute(属性)にすれば色の表示はできるのですが…
例えば実線は黒いマテリアル (カラー属性は無設定) というような場合には、
実線が白になってしまっていて困ります。

描き方のスタイル次第かと思いますが、私の場合はマテリアル色とカラー属性の混在した絵が見つらいので「マテリアル」か「レンダー」で表示しています。



⚠ 線のカラー属性は頂点ごとに違った色をつけることができます。
色違いの線を接続したり、後から Tint(チント)ツール、
または頂点ペイントモードのブラシで別の色に塗りつぶすことができます。

しかし、塗りの色は1色のみで、線の色とは別に定義されていることに注意してください。
(線を赤と青に塗って、塗りも赤青の2色で…とはできません)



描いた後からのカラー属性の編集は、Draw モードの Tint(チント)ツールか、
頂点ペイントモードのツールで行えます。

通常の塗りツールは、(ブラシのFalloff(減衰)設定に応じた)ぼんやりとした塗り変えです。
くっきりとした輪郭線で色を変更するときは、Replace(置き換え)ツールを使います。

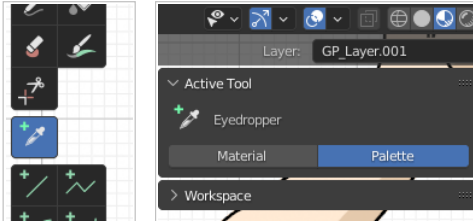
(動画 VertexPaint01.gif) *pdfでは最初の1コマが表示されています

スポイト(Eye Dropper)で色取得

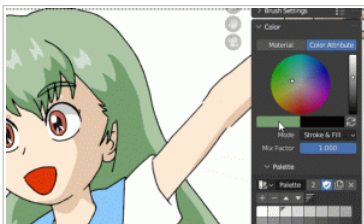
カラー属性を多用する場合、パレット機能などを利用して色を変更するのが常套手段なのですが、画面の色を直接取り出すスポイト機能が使えるとなお便利です。

日本語ではスポイトツールという呼ばれ方をしますが、日本でのスポイトという器具名の由来はオランダ語の注射器だそうです。英語でのこうしたツールの名前は、Eye Dropper で目薬を差す器材の名前です。

そのため、ショートカットキーなどで"E"などが使われると、日本人には分かりにくい名前になります。ブレNDERに限らず、スポイト機能はお絵かきなどでもよく出てくるので、この際 Eye Dropper も覚えてしまいましょう！(Dropperなので、ポタッと色を垂らす様子が連想されて、色を吸い上げる印象が弱いのが困りものです！)



お絵かきツールによくあるスポイトツールは、グリースペシル用には意外にもありません。と言うより、あるにはあるのですが、「取得した色のマテリアルを作る」もしくは「色をパレットに登録する」と、単純にアクティブな色を変更するよりも強い効果のツールになっています。



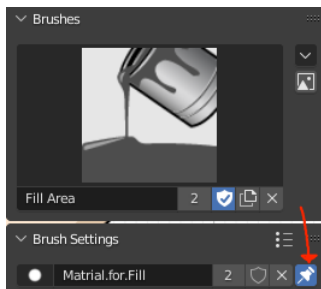
単純に色を取得するには、色のボタンをクリックしてパネルを開いた後で、スポイトアイコンをクリックすることで色を取得できます。

ただ、この工程は2行程あるので、カラー属性を使って絵を描くときに、頻繁にスポイトしながらアクティブな色を変えるには使いづらいところがあります。(動画 EyeDropper01.gif) *pdfでは最初の1コマが表示されています



意外に盲点で知られていないのですが、色ボタンの上で E のショートカットで、パネルを開くことを省略してスポイトモードに入ることができます。(Eye Dropper の E です。日本人には覚えづらいショートカットです。)(動画 EyeDropper02.gif) *pdfでは最初の1コマが表示されています

このショートカット"E"を使いこなせば、カラー属性を多用する使い方の時も色の選択時の手間(クリック数)がだいぶ少なくなります。



カラー属性でバケツ塗りをメインとした絵を描く場合、線を引くときは「黒単色の実線用マテリアル」塗るなら「カラー属性使用時のマテリアル」と使うマテリアルがほぼ確定しているような場合があります。

先ほど使用マテリアルのピン止めの機能を紹介しましたが、バケツ塗りのブラシには「バケツ塗り用のマテリアル」のように固定してしまうのも手間を削減する一つの手になります。

※) ただし、設定の仕組みを忘れてしまうと「今度は別マテリアルにしたい」という時に混乱するので注意してください！