

マテリアルやポストエフェクトが繰り広げる大冒険！



はじめに

今回の話は、マテリアルを使って色々やっていく話です。

マテリアル系の資料としてはもんしゅ氏の



『マテリアルデザイン入門』がおすすめです。といっても正直、皆さん同様に仕事が忙しくてほとんど読めてないのですが(泣)

それでもこの本から得たアイデアとか、インターネット上での皆さんが使用しているテクニック等、あとは製作の中で思いついた事についてお話しします。

まずは本を読まなくてもできる基本的なテクニックからお話しします。中級者の方にはつまらないかもしれませんが。

基本的なところ

最初は僕の好きなことをやってみます。僕はデッドスペースなどにみられるような、

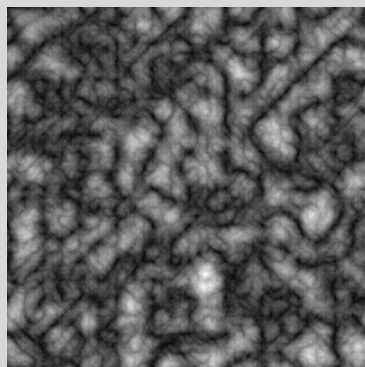


こういう表現とか



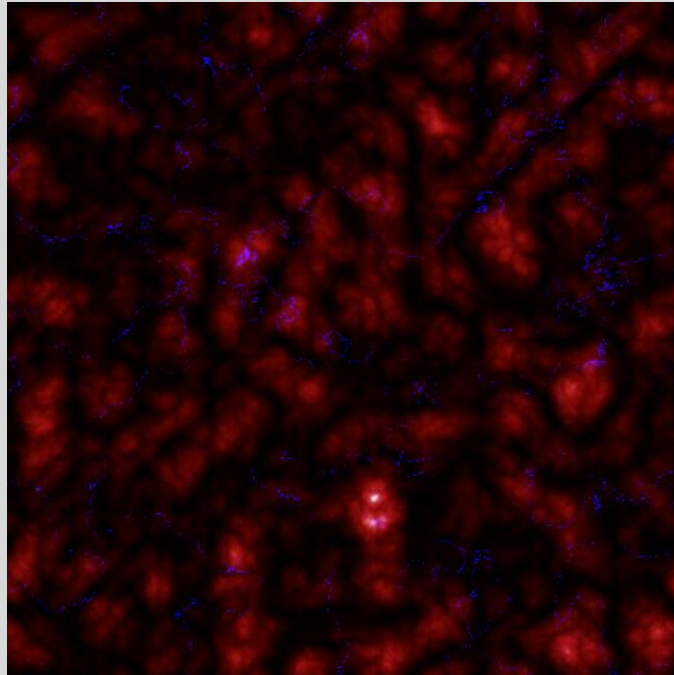
こういう表現が好きです。

こういう表現をするためには、赤っぽい模様と『ノーマルマップ』でそれっぽいのが作れます。
フリーソフトのGIMP を用いて

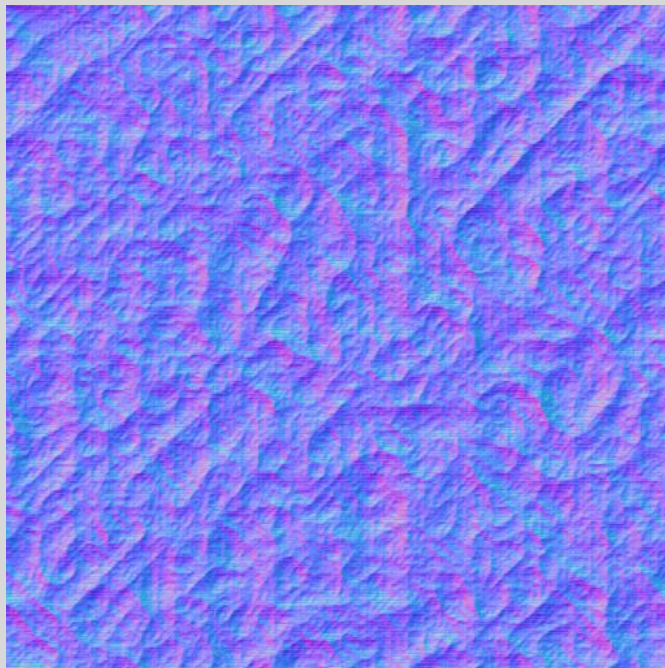


こんな感じの模様を作ります。雲模様のパラメータ次第でこういうのが作れます。この模様からさらに二つの模様を作ります。

まずは基本カラーとしてこいつを赤く染めて、ちょいちょいと青を混ぜて



こんなのと、後は先ほどの白黒画像をノーマルマップ化した

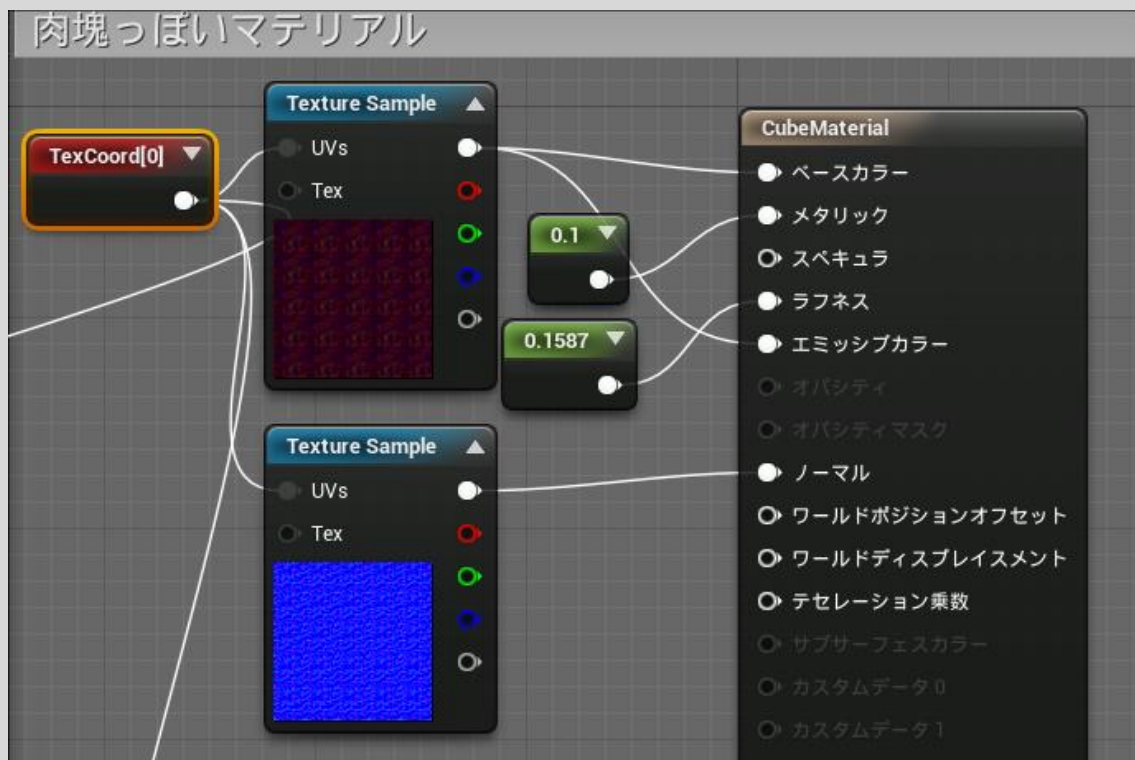


こんなのを用意します。ノーマルマップは GIMP, ノーマルマップで検索するとスグに出てきます。

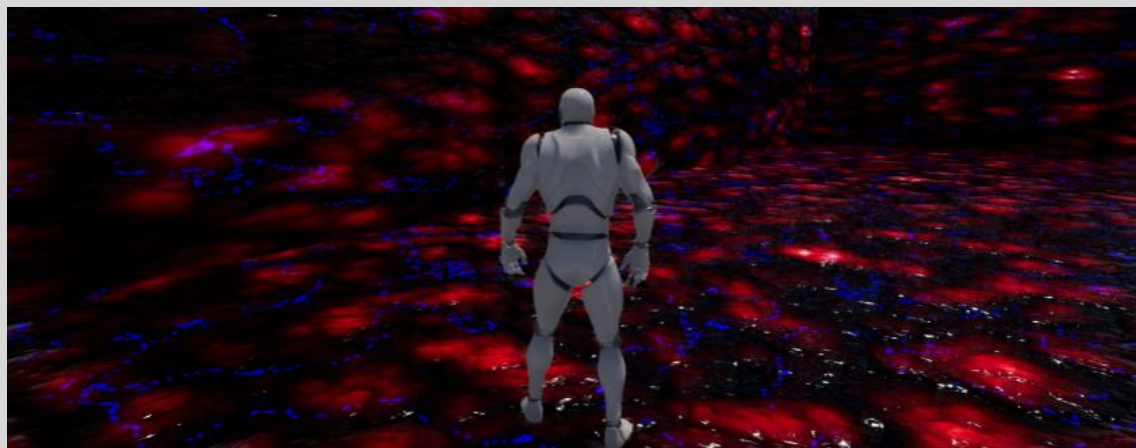
これを UE4 にドラッグアンドドロップして…
マテリアルを新規作成しましょう。

そしてベースカラーとエミッシブカラーに先ほどの基本カラー画像を、ノーマルにこの画像を割り当てます。

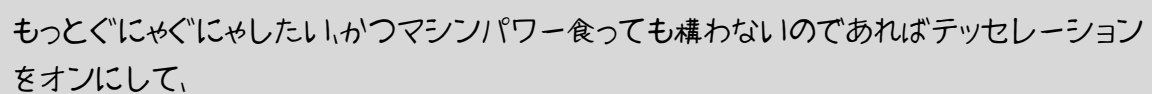
肉塊感を出したいので、ラフネスとメタリックもちょっとだけいじっておきます。



このマテリアルを壁とか床に適用すると…



振動は Time と fMod を使えばいいでしょう。所謂のこぎり波のような波形となりますので、



体型を変更しよう

いやね、結構なお金を払って、モデルデータとか買っちゃったりするんですけどね



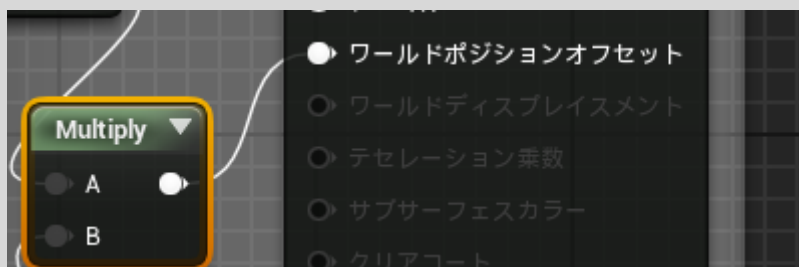
体型とか一緒じゃないですか。

モデリングツールとかで編集してもええんですけど、いちいちそんなんするの面倒じゃないですか？

マテリアルでどうにかありませんかねーと思ったわけですよ。で、色々試したり調べたりしてたら意外と簡単に体型を変更できるつばいなので、話してみようかと思います。

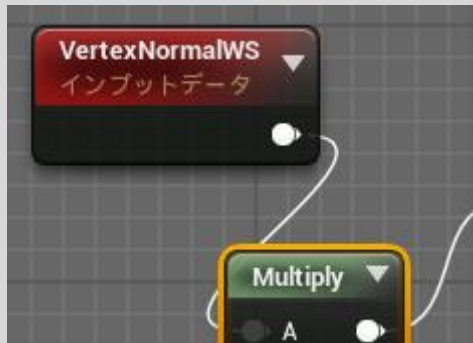
というわけで作りました。太ましいシェータ。ゾンビ君のマテリアルをコピーして作ります。

ポイントは「ワールドポジションオフセット」



これに加工することで、モデルにおける頂点座標を操作することができます。
さて、どう操作しましょう。

太くするには、それぞれの頂点の法線方向に頂点座標を移動させてやればいいですね？
そこで使えるのが、VertexNormalWS



ちなみに WS は WorldSpace の略です。ともかくこいつを使えば、頂点の法線を得ることができます。

あとはこの法線方向に座標を移動すればいいのだから、適当な数値を乗算してやってワールドポジションオフセットに入れてあげます。

そうすると



このように明らかに体型が変わってしまいます。
左の方がちょっとゴリラっぽくなってるでしょ？

テクスチャを使わずに模様を表現しよう

次にテクスチャを使わずに模様を表現してみたいと思います。

画像を使用しないで!?

そんなことがデキるのだろうか?

模様もなしに…?



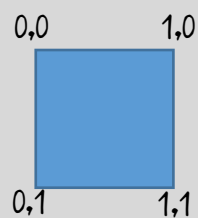
まあごもっとも。

ところがこいつを見てくれ…コイツをどう思う?



おわかりいただけたでしょうか?

画像も使用してないのに色がついていますよね?これはUV値をそのまま『色情報』になおしています。UV値っていうのは…だいたいこういう風になっています。



左上が黒いのは

(R,G)の並びになっているので

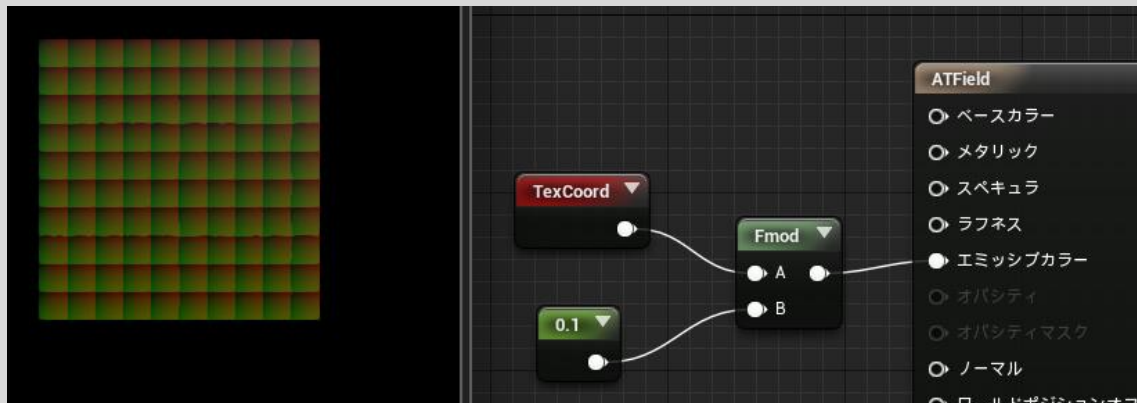
(0,0)→黒

(1,0)→赤

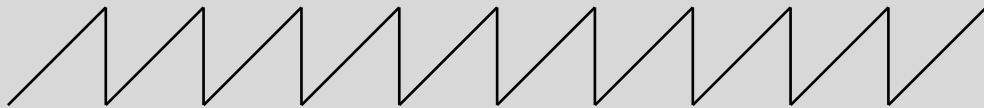
(0,1)→緑

(1,1)→黄色

という模様になっています。既に模様あるでしょ？これをちょっといじります。



fmod って関数があるでしょ？これは割った余りを返す関数です。例えば今、0.1 で割っています
が、この結果的に



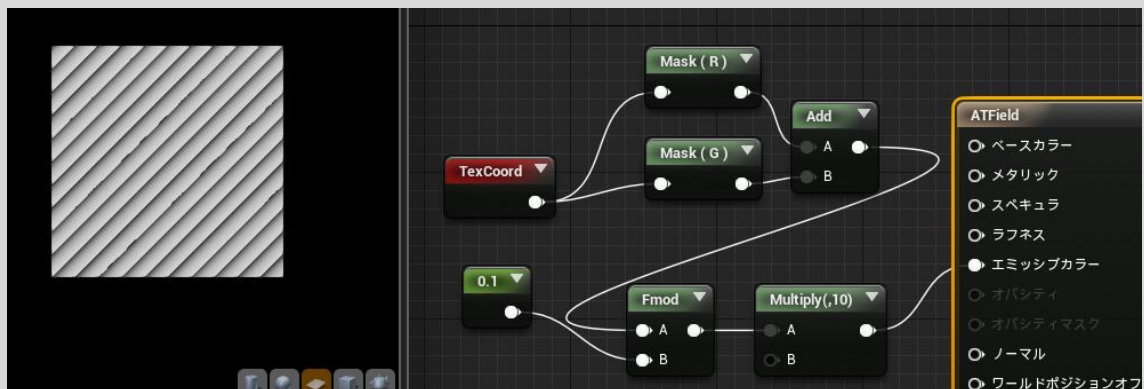
こういう感じになります…伝わりますかね？

結果的に値が 0.1 頭打ちなので暗くなっていますね。暗いのがイヤだったら 10 倍すればいい
ですよ。Fmod 結果を 10 倍しておきます。

でもこれじゃちょっと使えないですね。模様を斜めに見してみます。模様を斜めにするには
U+V を使用します。わかるかな～？U+V を利用すると斜めになります。

つまり

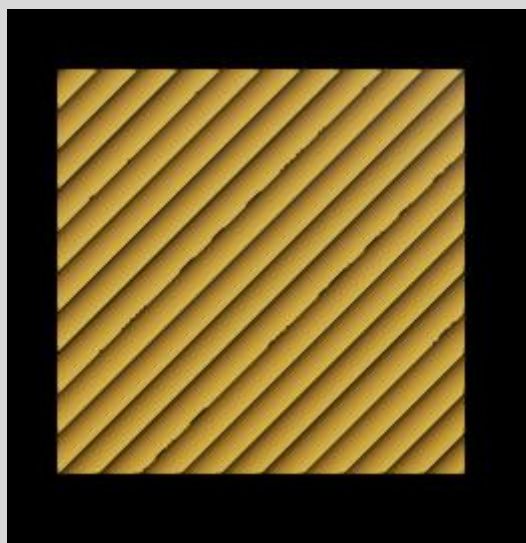
明るさ=U もしくは V ってやってるのを U+V にすると右斜に向かって大きくなる式になります。
結果として



こうなる。ちょっと色飛んじゃったけど、U と V を一緒に扱っちゃうのでこういうことになります。

ちなみに R と G と足したければ ComponentMask でいったん R と G に分割して Add すればいいんですが、これ hlsI だと、color.r+color.g で済むはずなので、正直ちょっと面倒だと感じます。

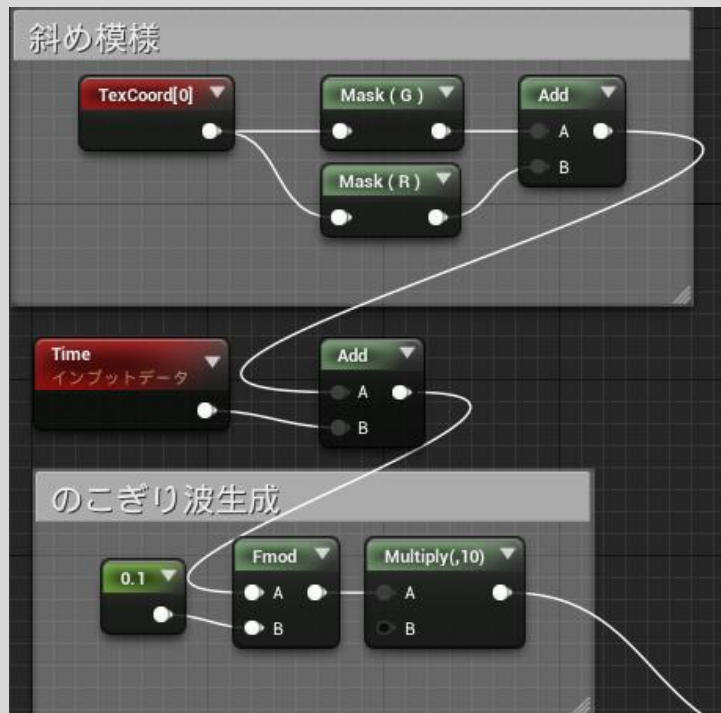
色付けたければ普通に ConstantVector3 でなんかの色作ってと掛け算しとけばいいですね。



ここまでがテクスチャ無しでできます。あとはこれを AT フィールドっぽくしてやるだけです。これだとつまらないので、動かしてみます。

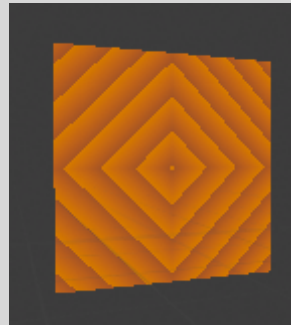
簡単です。

上の計算結果に Time 関数の計算結果を足すだけで斜めに動きますね？



さてさて、これでもまだATフィールドっぽくないですね？

こういう風にしたい



とするとどうすればいいんだろうか？後で話しますが『位相をずらす』という考え方で解決します。

いったんさかのぼって考えてみましょう。

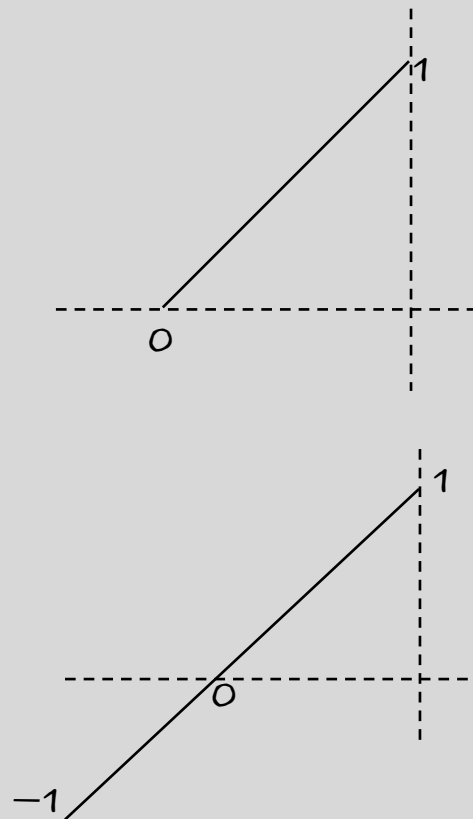
以下のぎざぎざはどうやって作ったものでしょうか？



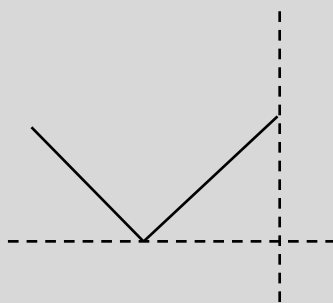
これはもともと 0~1 の範囲において

このようになっていたものを Fmod かけた結果が今回の結果でした。

なお、0~1でなく。結果*2-1=-1,1にすると



このように考えられますね？0を境にマイナスになっています。さらにこの位相を 0.5 ずらす→見ているポイントをずらす…と(-0.5~0.5)の範囲になり、値も(-0.5~0.5)になります。この状態に対して Abs 関数(絶対値にする)を適用すると

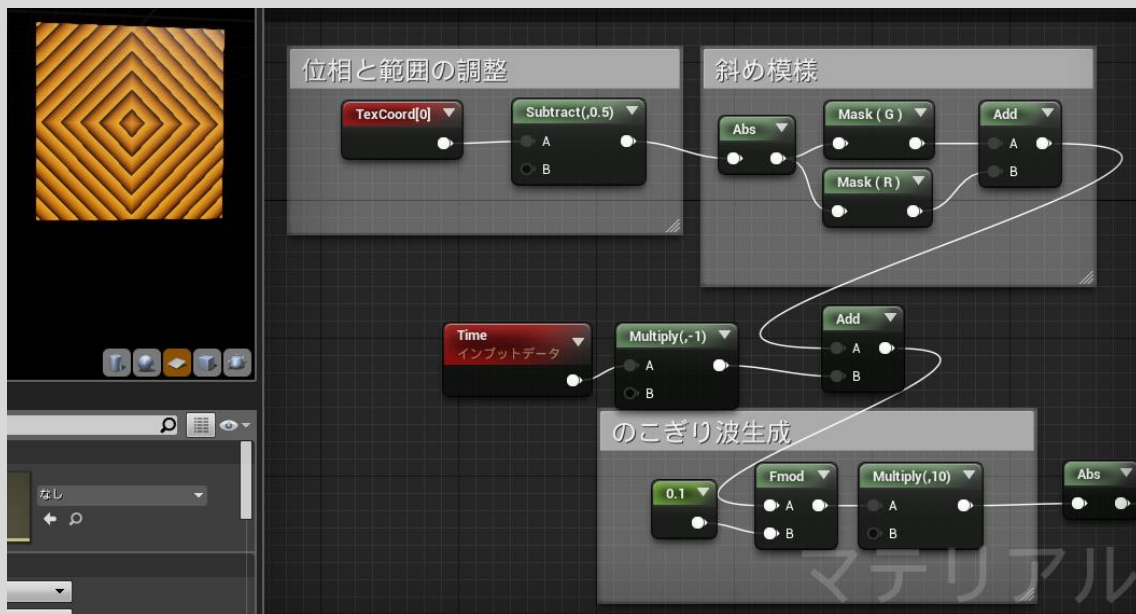


こうなるわけ。絶対値ですので。
さらにこれに FMod がかけると



こうなり、真ん中から広がるような形になるわけです。

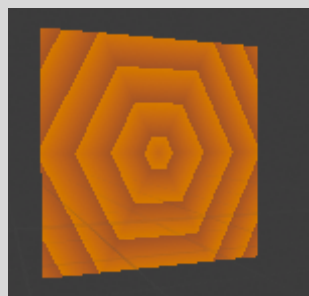
なお、真ん中に収束するようになった場合は、Time にマイナスの値をかけてください



さて、ここまででもわりとATフィールドっぽくなったと思います。

ただ、ATフィールドは6~8角形っぽいものですよね？

試しにも角形っぽくしたいならどうしたらいいんでしょう？



今度はMax関数とMultiply関数を使用します。

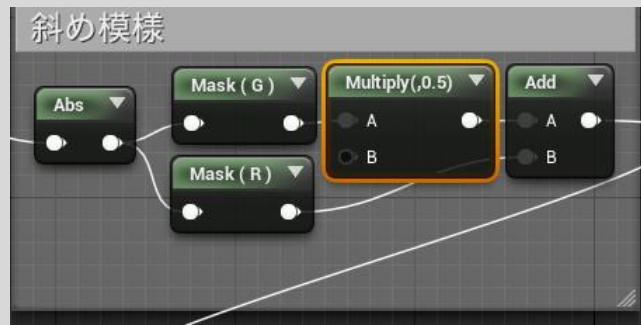
Maxはどちらか大きいほうを返す関数で、Multiplyは単なる掛け算。

さて、現在の明るさ情報を

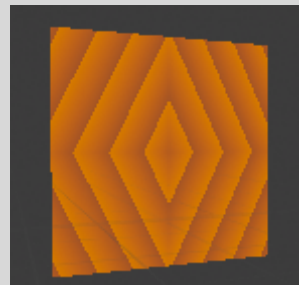
U+Vにしているから、斜め45°なのだ

これを

$U/2+V$ とかにしたらどうだろう？



U の値の進み方が緩やかになり、傾きが変わるわけです。
結果として

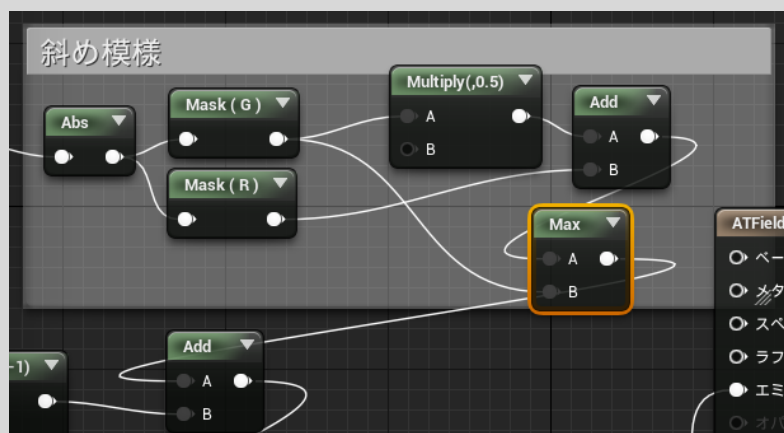


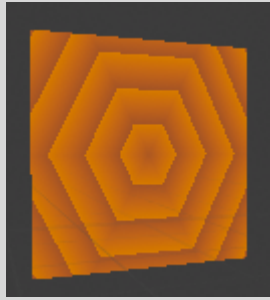
こうなるわけです。

そもそも一次式である以上は 4 角形以上にはならない。ならないんですがここで Max 関数を利用します。

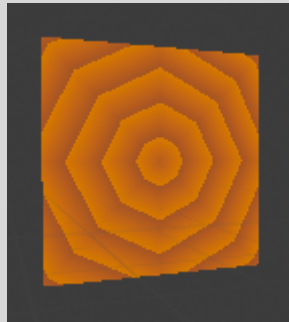
どういうことかという、上のやり方の場合

U と $0.5U+V$ のどちらか大きいほうを選択した場合、一部は (V が $0.5U$ より小さい場合) U のほうが大きく、(V が $0.5U$ より大きい場合) $0.5U+V$ となるようにすれば上下の部分が水平になるわけ。これを Max 関数で場合分けすれば結果として、

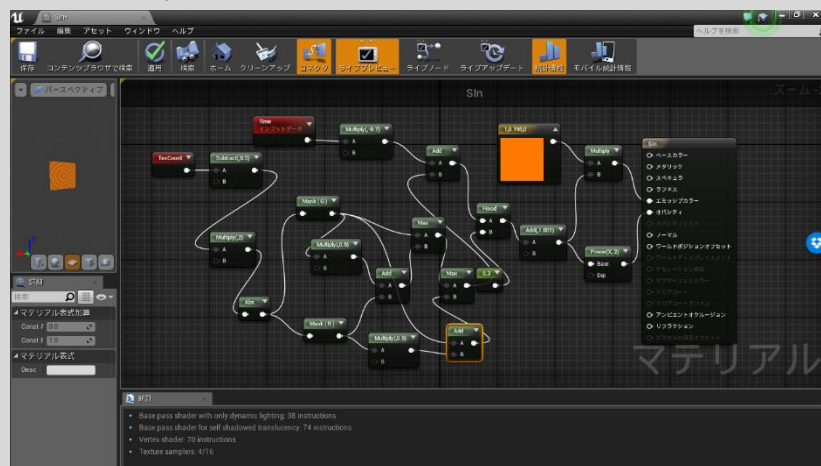




このようになります。これで AT フィールドっぽくなりました。
ちょっと応用すると



このくらいにはなります。なお、これを実行するマテリアルは



だいたいこんな感じです。

炎のマテリアル

実は UE4 のマテリアル自体にはそれほど興味がなかったのですが、



こういうものを見せられては、血が滾らざるを得ないわけです。

マテリアル本に書いてありますが、これは GDC2014 のデモらしいです。UE4ランチャーから見れますので、興味がわいた人は見てみるといいと思います。インターネットで検索してもムービーばかりが出てくるので、注意しましょう。

ちなみにこのデモはバージョン 4.11 用なのですが、コメント欄には

『4.16 に対応させてくれ〜』などと書かれており、根強い人気があります。まあね、こういう効果出せたらカッコいいしね。

ちなみにこの手のマテリアル(シェータ)を使いこなすには、ある程度の 3DCG の知識と、頂点シェータ、ピクセルシェータ、テッセレーション、ノーマルマップについては最低限知っておく必要があります。『使うだけ』なら、持ってくればいいのですが、自分で新しいのを作ろうと思うと、そういう知識は必要です。

実はそういう基礎知識があれば作者のシェータコード見なくても大丈夫!『たぶんこうやってんだろー?』って予測が立てられます。

後の話で話しますが、この『予測』は学習効率を上げるためには非常に重要なので、分からないなら分からないなりに予測しましょう。

https://www.youtube.com/watch?v=FcxA_xYnHZ8&feature=youtu.be&t=3m55s

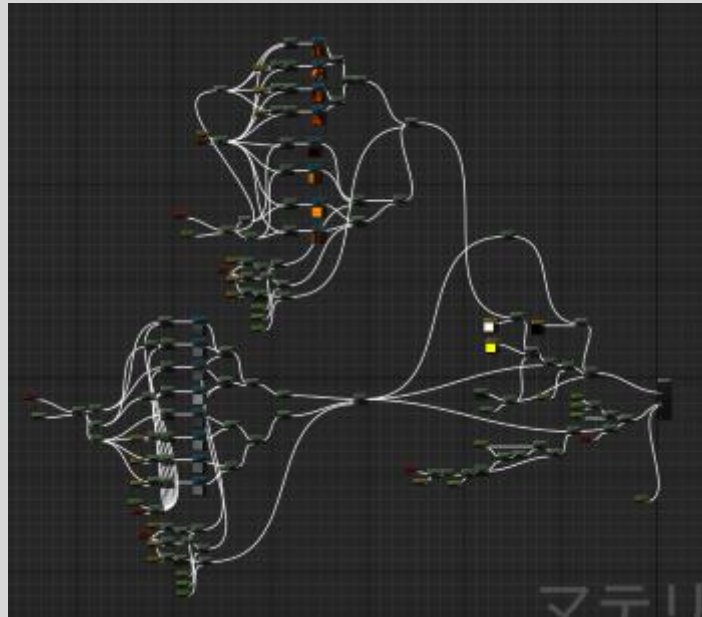
これの丁度 4 分ぐらいの映像を見えます。

この映像からわかることは、

- 頂点が波打っている
- 炎テクスチャが UV アニメーションしている

ぐらいです。UE4 初心者の僕にわかるのはこんなもんです。でいきなりシェータコードを見

たいところですが、まあぶっちゃけた話をしちゃうと



うわあ…ご覧の有様だよ!!

これじゃあ解説の 30 分すぐ使っちゃうよ!!というわけでシェーダコードを見ずに先の二つの観察から試作版を作ります。

上のファイアーマテリアルの詳しい解説は次回に単品としてやっていきます。今回は勘弁してください。ちなみに、今回の試作は『失敗例』です。もし、正解『だけ』を知りたい人は、前述のサンプルを落として自分で解析するなり使うなりしましょう。

まず『頂点を波打たせる』ためには恐らく

- ワールドディスプレイメント
- ノイズテクスチャ
- テッセレーション

が使用されていると思われますので、試してみましょう

まず、下準備として、FireTest というマテリアルを作ります。テクスチャに炎のテクスチャを適用します。

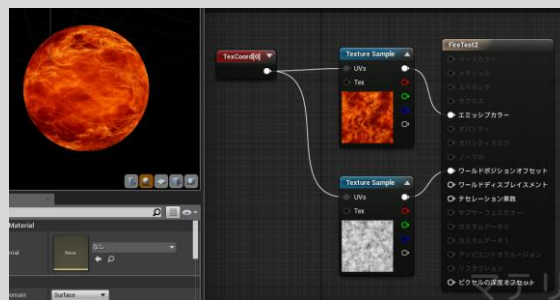
特にシェーディングの必要がないので、ShadingModel は Unlit にしておきます。

エミッシブカラーに T_Fire_Tiled_D を設定し、



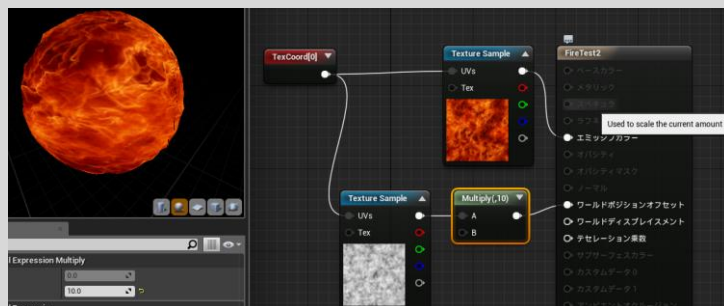
こんな感じにします。

次に形をぐにやぐにやにしたいので、ワールドオフセットから、線を引っ張って TextureSample につなぎます。で、テクスチャを適当なノイズ画像にします。

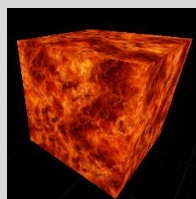


(3)全然変わらねーじゃん。って感じなので、ノイズのカラーに対して 10 倍くらいの値をワールドポジションオフセットにつなぎます。

そうすると

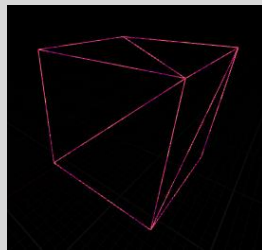


若干ではありますが、形にノイズが入ります。ところが、左のサンプル形状を立方体にすると全然歪みません。

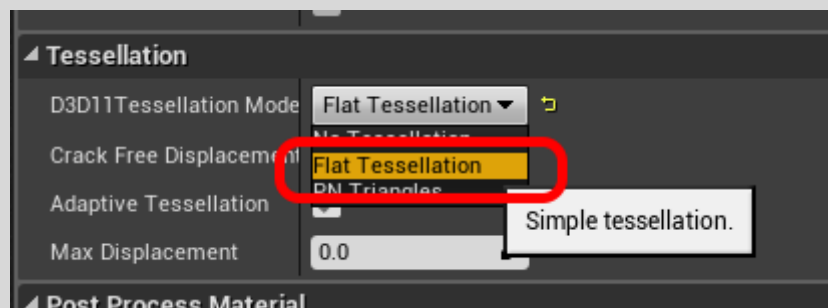


これは頂点数が 8 点しかないため、ノイズを入れようがポジションオフセットしようが形が

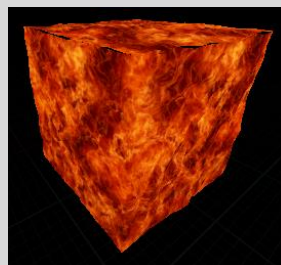
大して歪まないからです。



というわけでテッセレーションの出番です。まず「マテリアルのテッセレーションパラメータを変更します。NoTessellation から FlatTessellation にします。これで頂点を動的に増やせます。

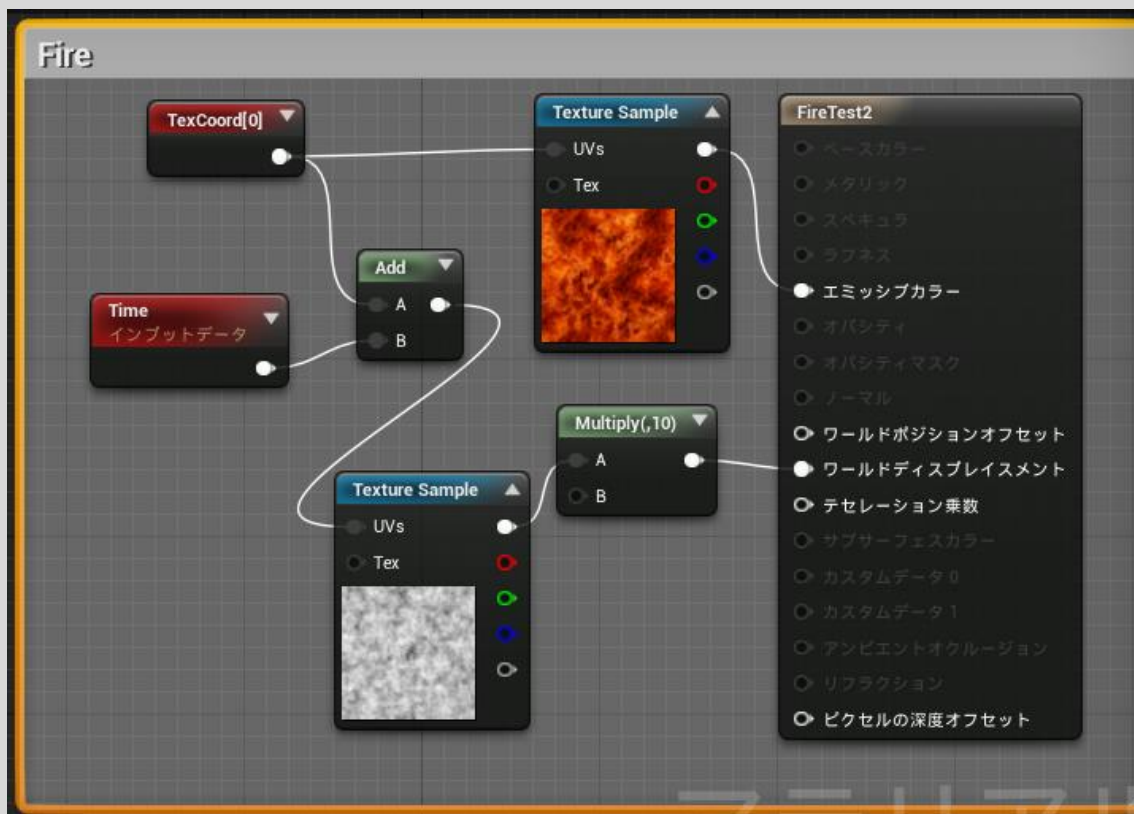


そうしたうえで、先ほどまでワールドポジションオフセットにつないでた部分をワールドディスプレイメントにつなぎ替えます。そうすると



立方体も若干ですが歪みましたね？

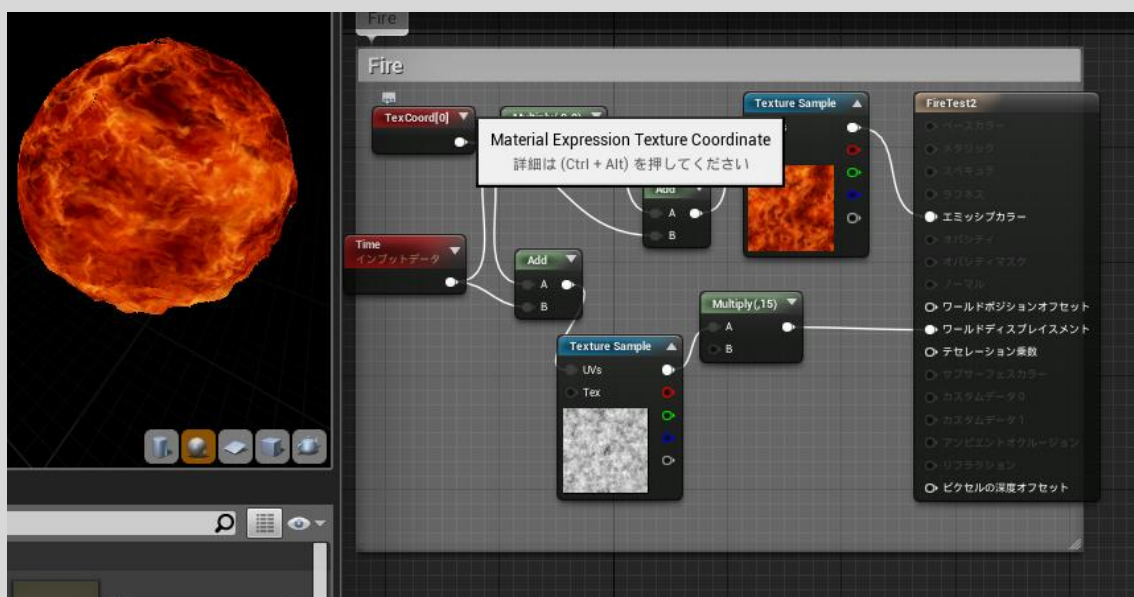
あとはこのノイズの UV を時間(Time)ですらしてあげれば



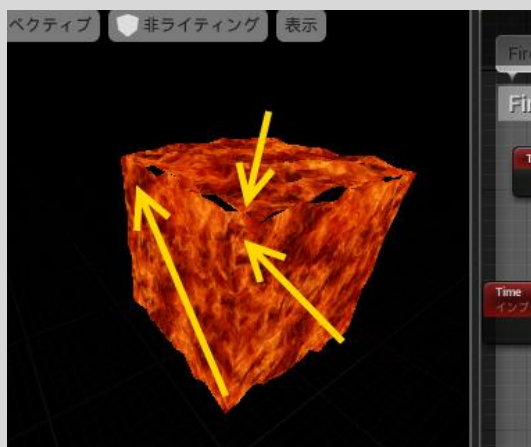
微妙に炎つぽい動きをするマテリアルになります。

頂点の方はこの辺でいいでしょうが、テクスチャの方はどうしましょうか？これも時間で動かしましょう。ただ Time そのまま UV に Add しては早すぎるので 0.2 くらいをかけてみます。

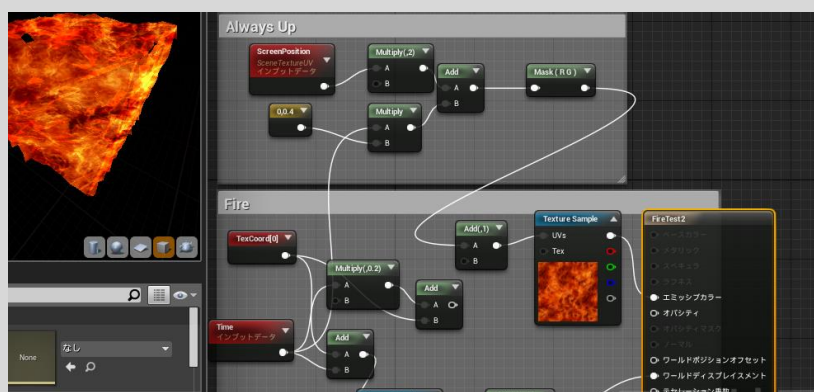
そうすると



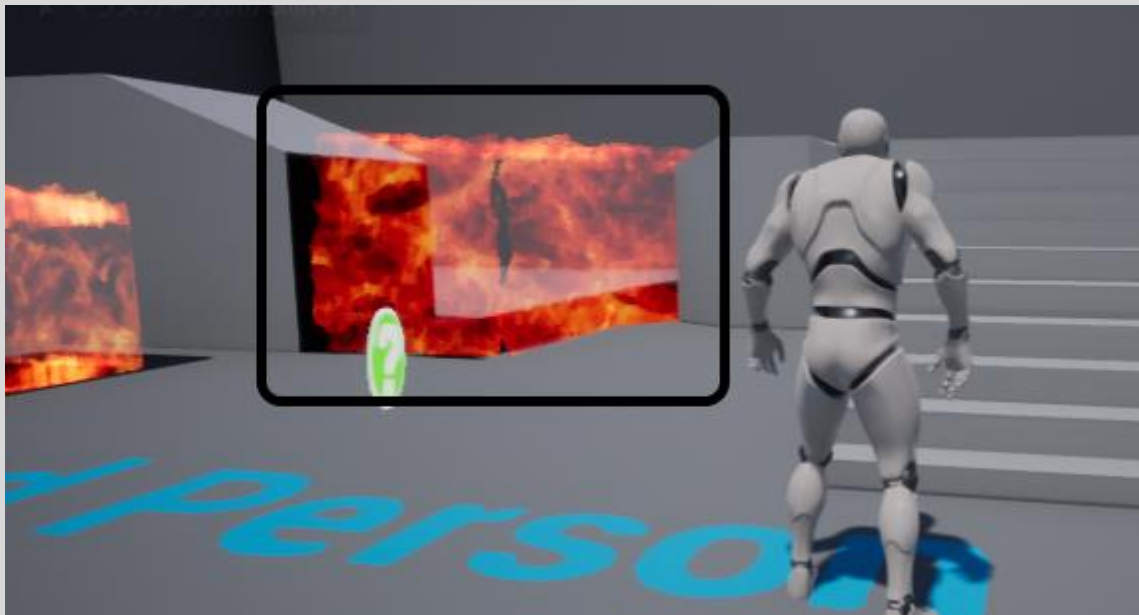
うん、それっぽくはなってきた…かな？でもまたオブジェクトを立方体にするとう問題が表面化します。UV の動きの向きが



こうなってしまうんですね。そこでこう考えました。
「そうだ!!常に上方向にしたいなら、スクリーン座標を基準にすればよくね?」
というわけで



こんな風にしちゃいました。そうすると、まあ、確かにすべて上方向に UV 値は移動してくれるのですが…これをゲーム中に置きちゃうと…



静止画じゃ分かりづらいですが、かなり気持ち悪いことになります。ぶっちゃけると UV 動かさないほうがまだ見栄えがいいですね。

今回は時間ありませんしこんなところで。次回に本格的な炎については話したいと思います。

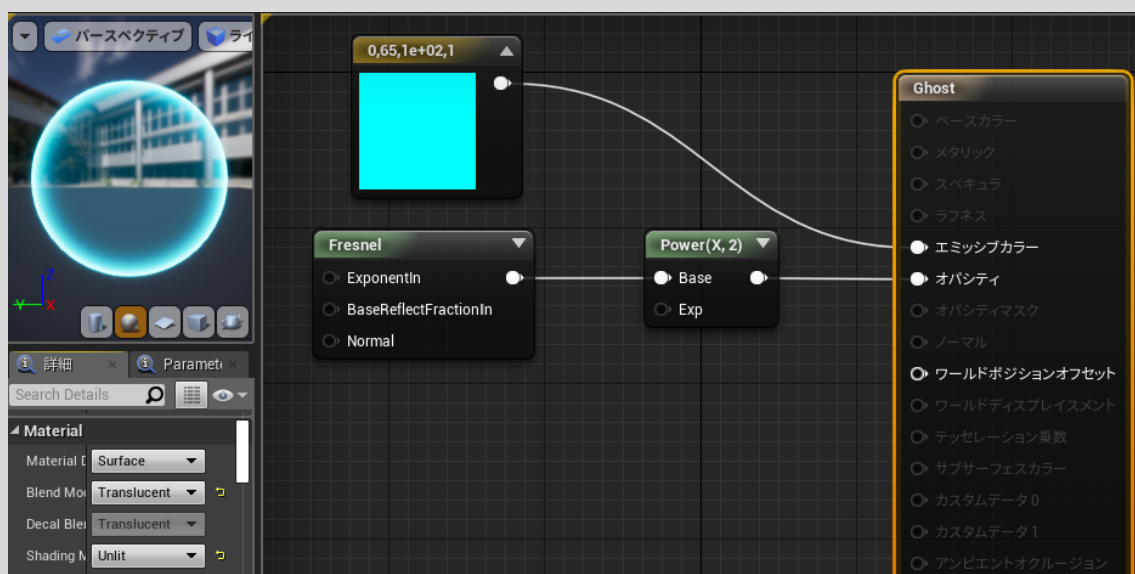
幽霊っぽいマテリアル

まま、とりあえずさ、幽霊っていうかお化けっぽいマテリアル…作ってみるか？意外と簡単なので、パパパッ…と作って、終わりッ!!

ここで使うのは

- Fresnel
- Emissive によるグロー

です。ゴチャゴチャいうよりマテリアル見てもらったほうが早いと思います。



ハイクツソ簡単。

Fresnel はもともと水面とかに使用するもので、澄んだ水面を思い浮かべてほしいんですが、真正面からだと水面の中まで見えて、斜めから見ると段々反射してしまうと思います。

それをシミュレートするためのパラメータだと思ってください。

基本的には視線ベクトルと法線ベクトルの内積をとって、それを $1-x$ で反転してるのと同じ意味です。

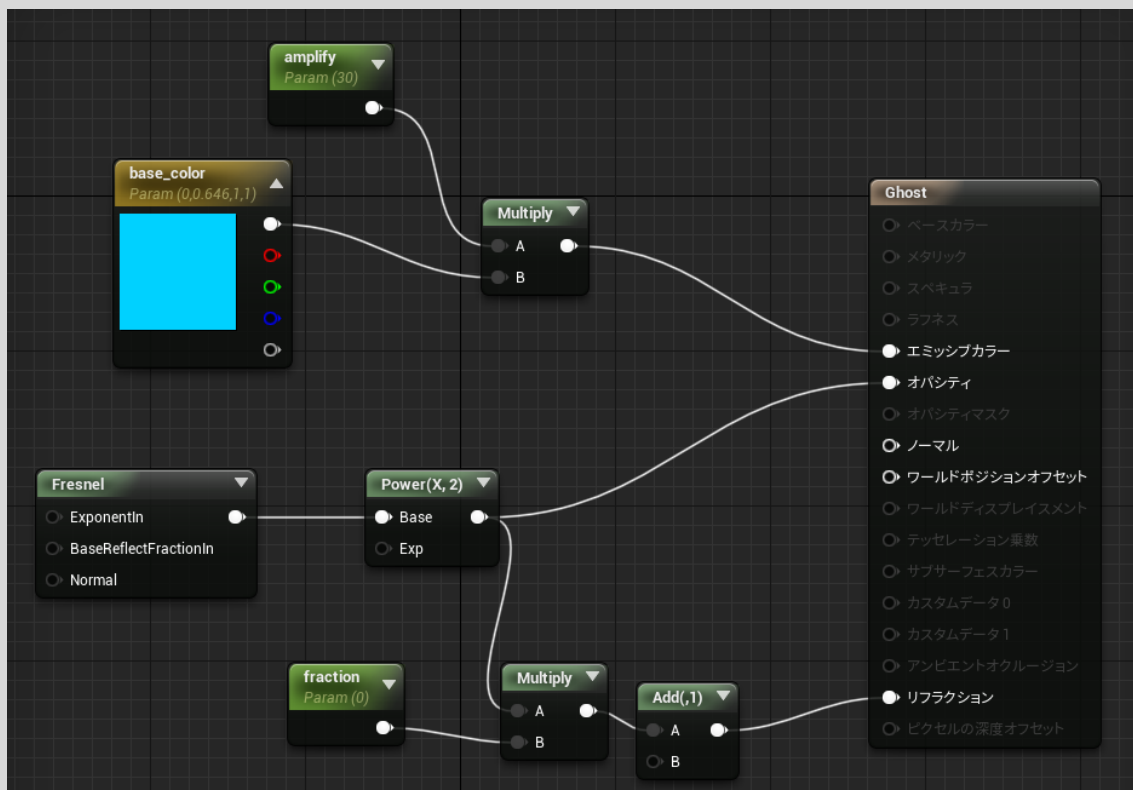
ちなみに Exponent は乗数のことで、↑の計算を n 乗します。1 乗だと真正面向いてる 1 点周りでしか透明にならずにオバケ感が薄いので、ExponentIn のパラメータを 5~10 にします。

次の BaseReflectFractionIn が実はよくわからないんですが、計算的には

$$\text{Fresnel} = \left(1 - \text{saturnate}(\vec{V} \cdot \vec{N} + \text{baseRef})\right)^{\text{exponent}}$$

こんな感じなんじゃないかなと思います。ちなみに下のほうに『リフラクション』ってありますが、こいつは 1 から離れれば離れるほどゆがみます。

なお、最後のパラメータの Normal ですが、これは物体の法線を使用するなら特に設定の必要はありません。とりあえず後から柔軟に扱うために、ベースカラーと発光具合(amplify)と歪み(fraction)をパラメータ化してマテリアルインスタンスにしておきます。

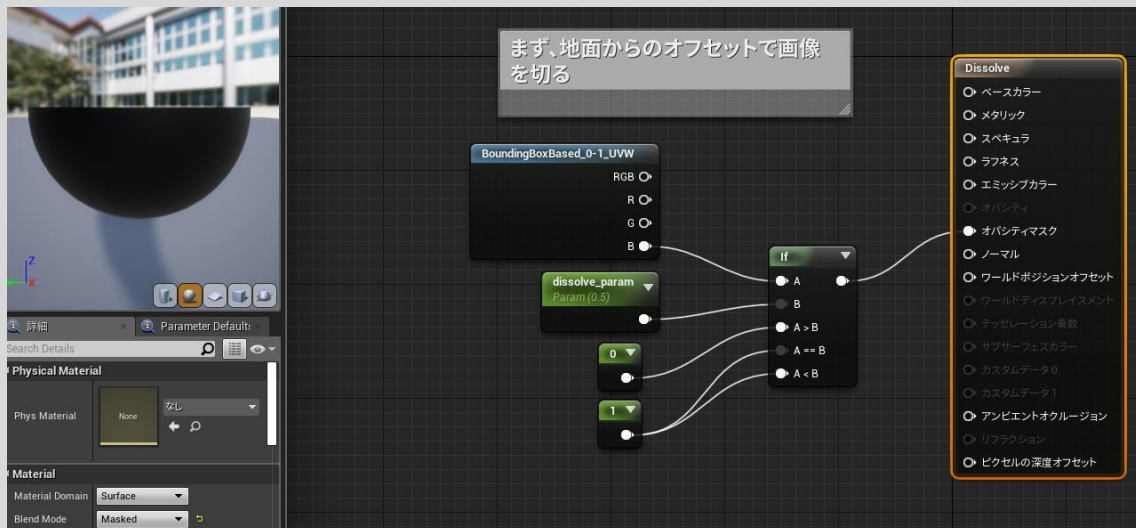


で、マテリアルインスタンス側で調整する…と。

ディゾルヴエフェクト

最も簡単なエフェクトです。ひとまず『雲模様』的な奴が生成できるグラフィクスツールを用意してください。で、雲模様(シームレス)を作っておいてください。

まずはその雲模様は置いていて…



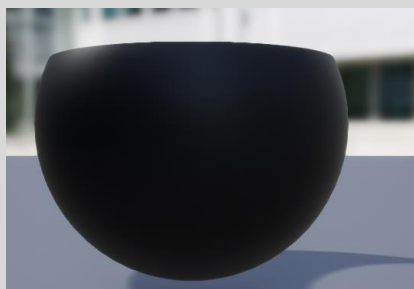
こんな感じのを作ります。『オパシティマスク』ってのが本当は透明度を元に可視不可視を決定するものです。

次にポイントになるのが『BoundingBoxBased_0-1_UVW』ってやつですが、これほんと探しづらいよね…知ってないと無理だわ。

で、これのBが高さ方向になります。これがバウンディングボックス内で0~1に正規化されているため、閾値としてパラメータ化している constant 値を使用します。

$$\begin{cases} Zvalue > Constant : 0 \\ Zvalue \leq Constant : 1 \end{cases}$$

こうすることによって、Constant を動的にいじることによって元のメッシュの特定の高さからぶった切られます。

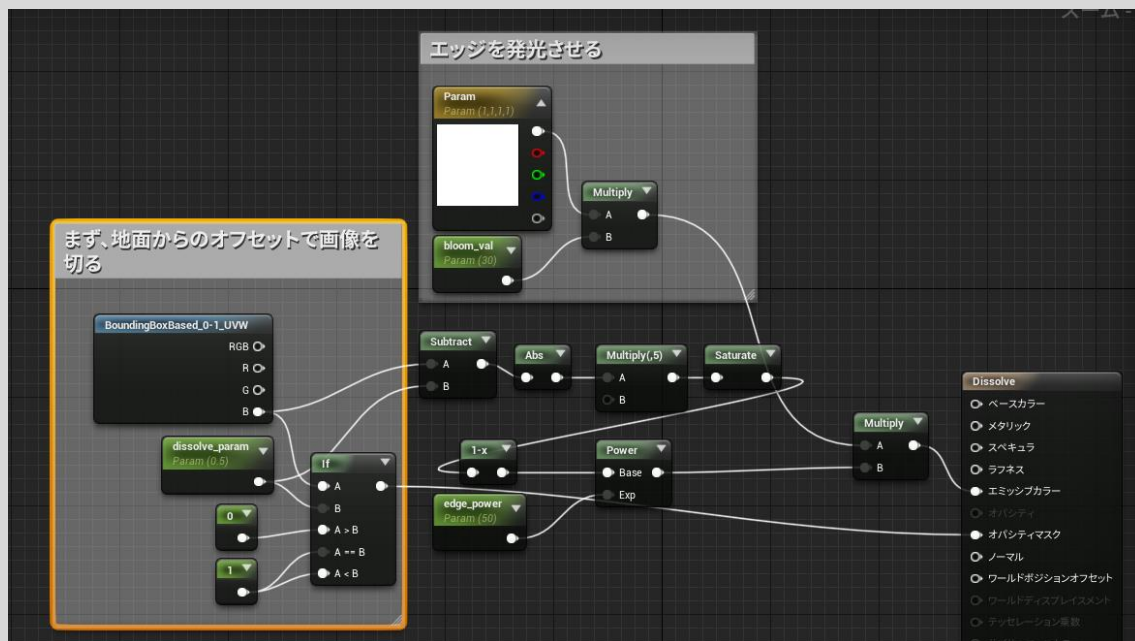


こんな感じ

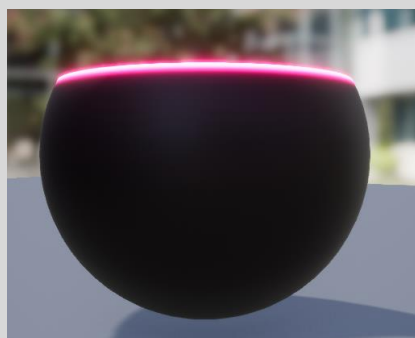
なので Constant はパラメータ化して、マテリアルインスタンスからいじれるようにしておいてください。

ここまではいいですか？

で、断面をそれっぽくするためにエッジ(断面近辺)を光らせたいと思います。ちょっとえげつないことになってますが、こうします。



で、マテリアルインスタンス側でエッジに色を付けるとこんな感じです。



黒以外にもしてみましょう

ベースカラーをいじれば黒以外の物体にもできますので、



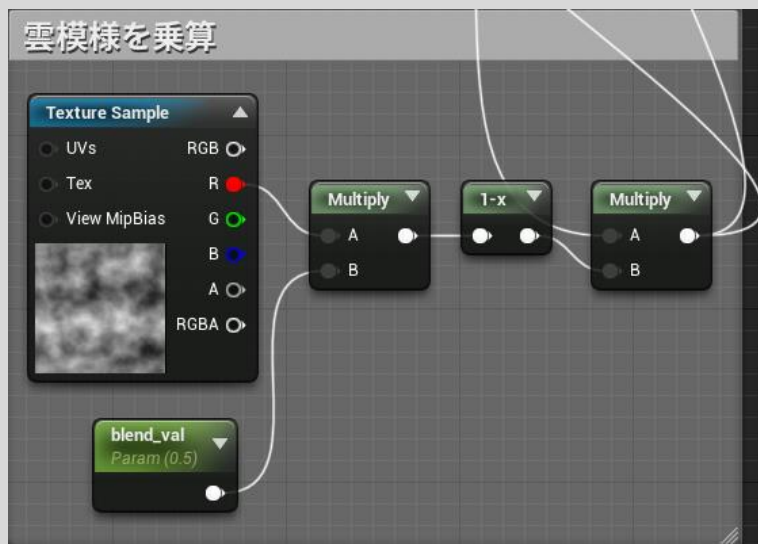
こんな感じにすることもできます

でもディゾルヴとしてはまだまだです。という事で、最初に作っておいた雲模様をインポート

してください。



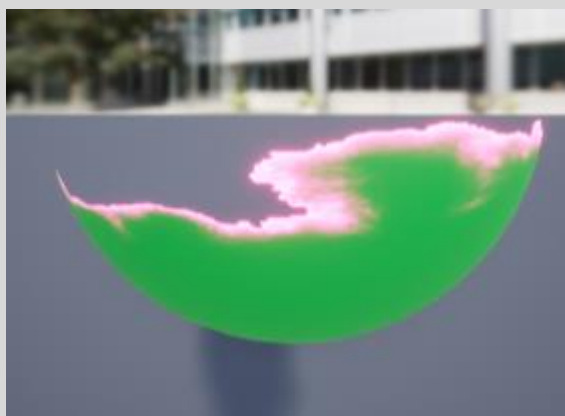
こっからよお…



こうします。1-x にしてるのは基本的に高さのディゾルヴを優先するからです。つまりパラメータ元を 0 に近くしています。

そして元のパラメータと乗算

if のところと emissive のところに乗算すれば…終わり!!



もちろん、この雲模様の適用率を blend_val としてやって、外部からパラメータでいじれるようにしておけば…



こんな感じでいい感じになります。ちなみに two_sided をオンにすると



こうなります。