ちょームズい講義を聞くということ

GLSL50連発

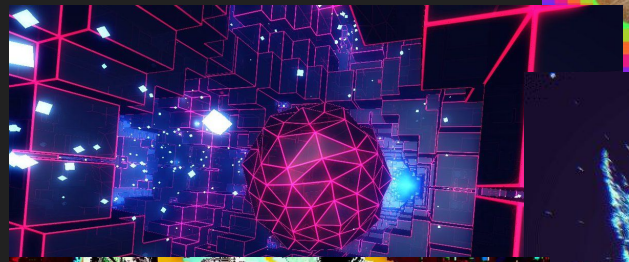
シェーダ味付け101

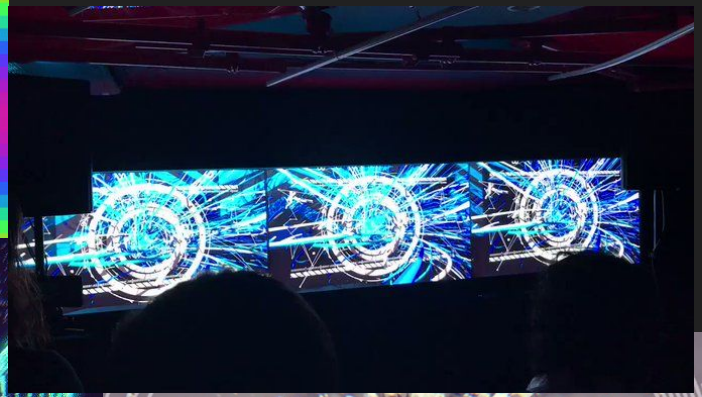
小渕 豊 (@FMS_Cat)

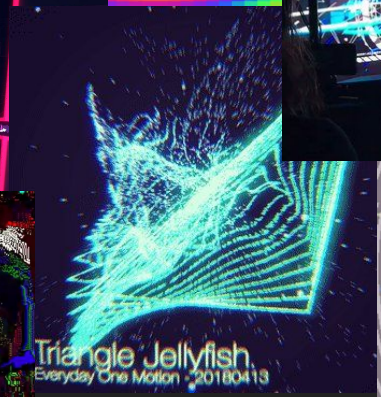
GLSLスクール 第3回 - 2018年11月13日

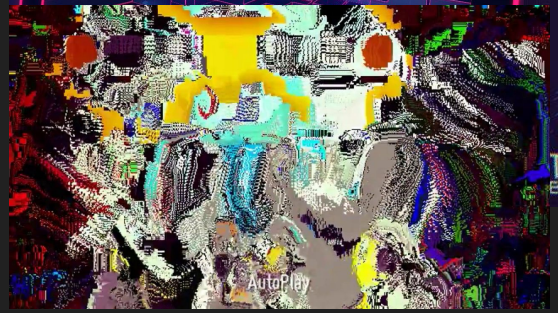
こんにちは

FMS_Catです





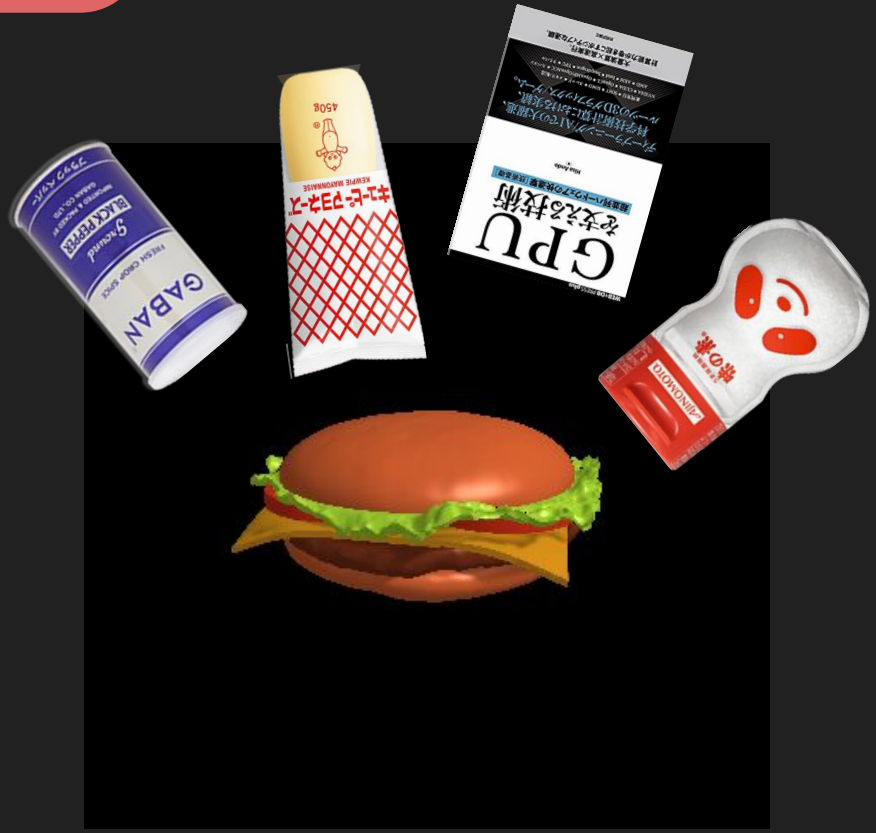




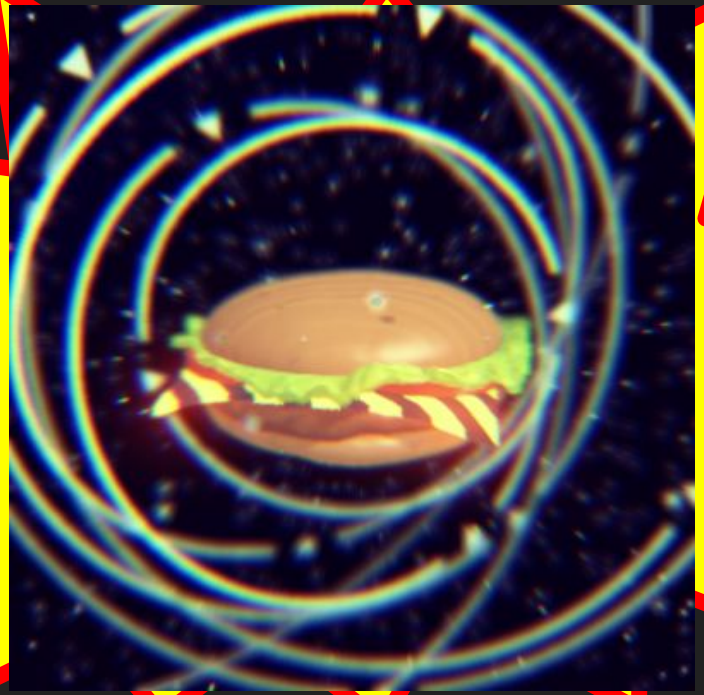
こんにちは


無題のプレゼンテーション
ファイル 編集 表示 挿入 表示形式 スライド 配置 ツール アドオン ヘルプ
変更内容をすべてドライブに保存しました
プレゼンテーションを開始
共有
Animations
スライド: 切り替えなし
アニメーションを追加
再生
シェーダ味付け101
小渕 豊 (@FMS_Cat)
GLSLスクール 第3回 - 2018年11月13日
クリックしてスピーカー ノートを追加
データ探索

こんにちは



こんにちは



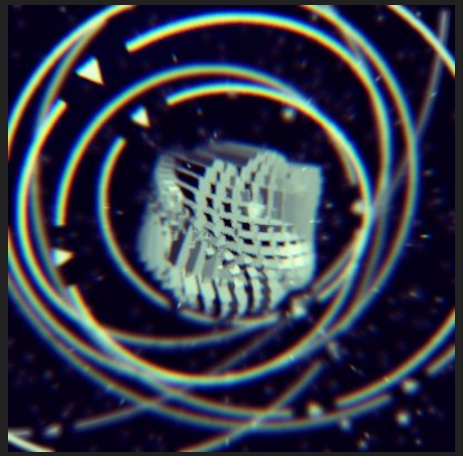
こんにちは

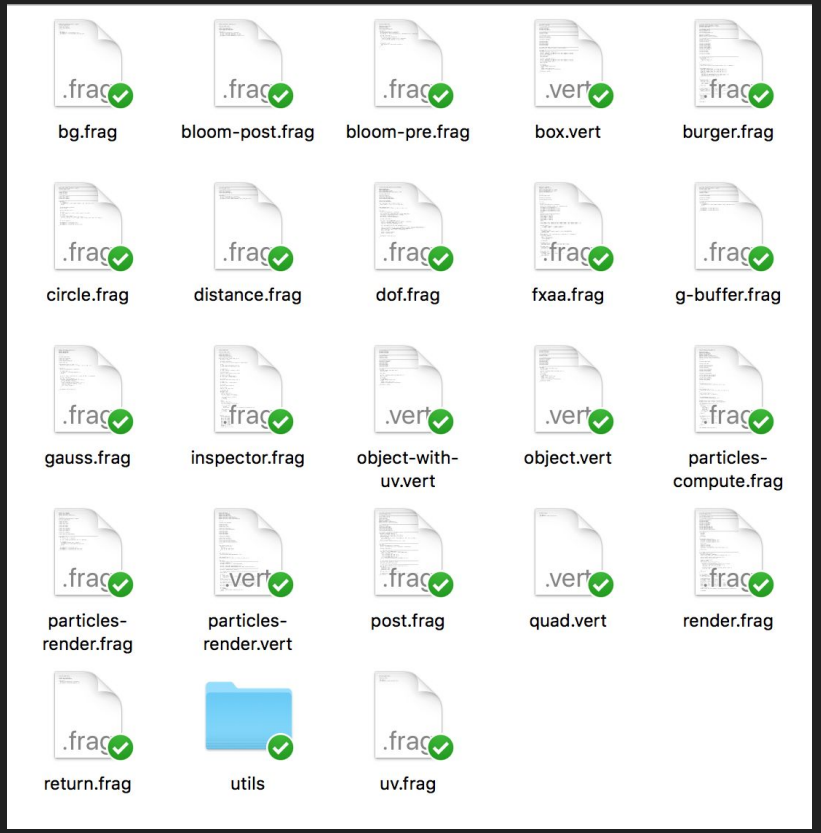
味付けします

- もとの絵があることを前提とする
- 一さじ加えてプロの味
- 調味料 = これ自体が重くなることは望まない
- 今日は味付け例を絵で紹介します

こんにちは

FMS_Catの
プロジェクトの様子





シェーダだけで22個！ (参照)

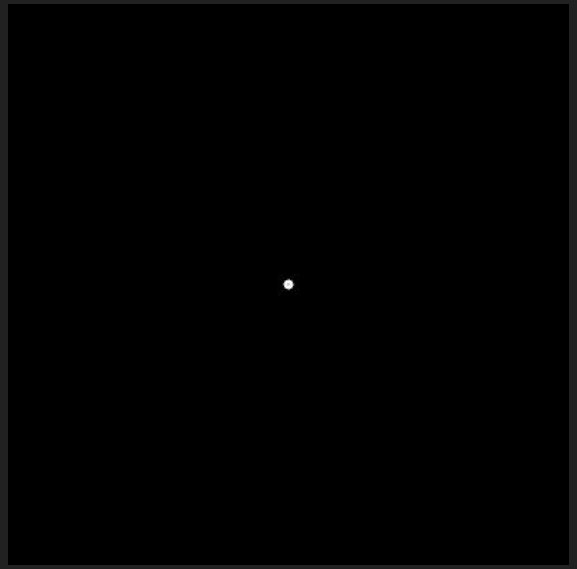
こんにちは

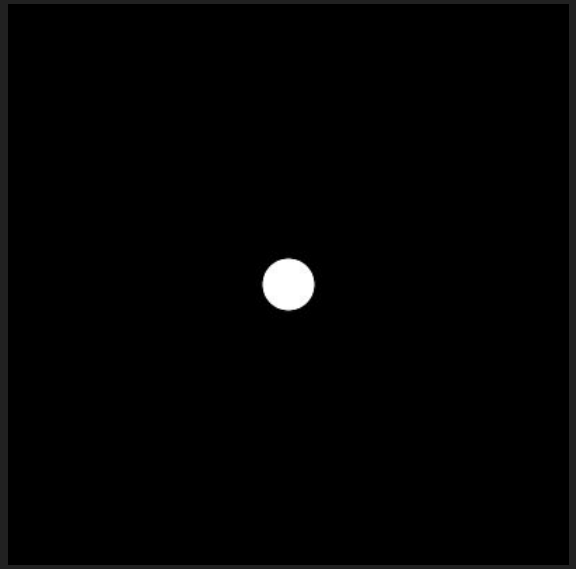
なるべく1パスで済むことを目指す

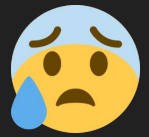
（けど2パス以上使うときのセオリーも紹介します）

シェーダだけで22個！

アニメーション





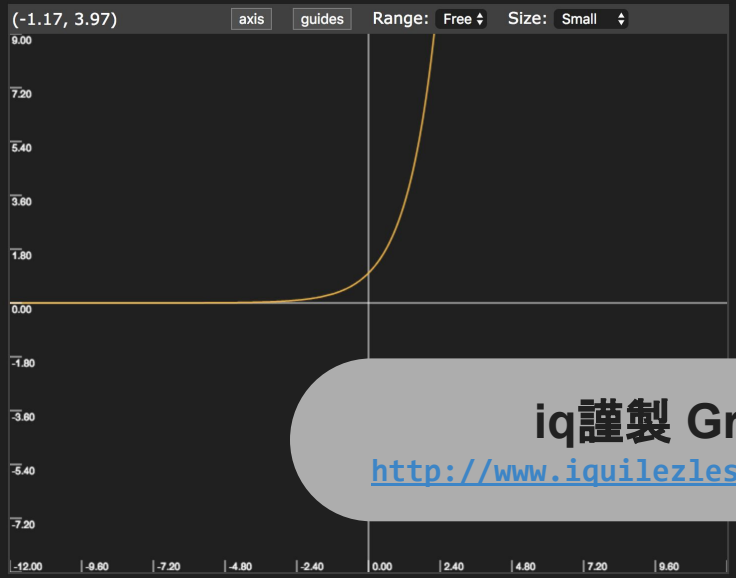


アニメーション

expを使え

アニメーション

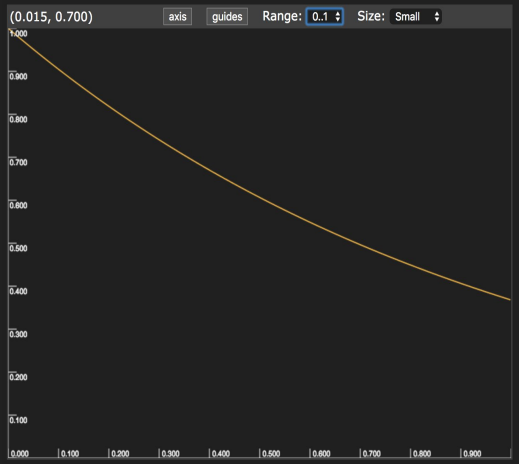
exp(x) = e^x = e(自然対数の底)のべき乗

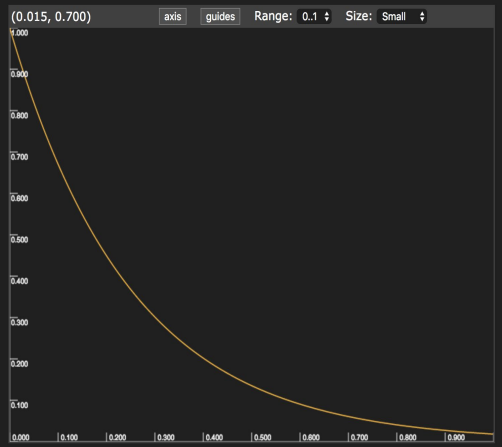


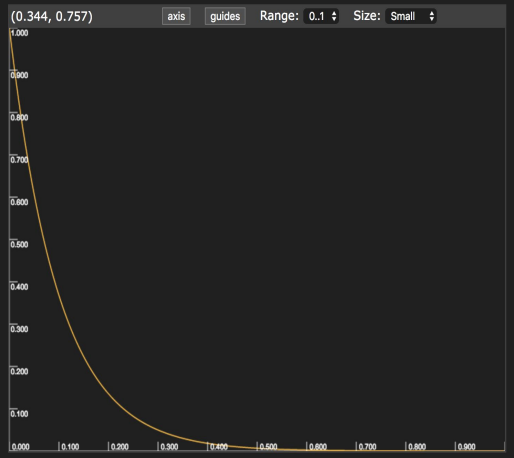
iq謹製 GraphToy

http://www.iquilezles.org/apps/graphtoy/

アニメーション







`exp(-x)`

`exp(-4*x)`

`exp(-10*x)`

アニメーション

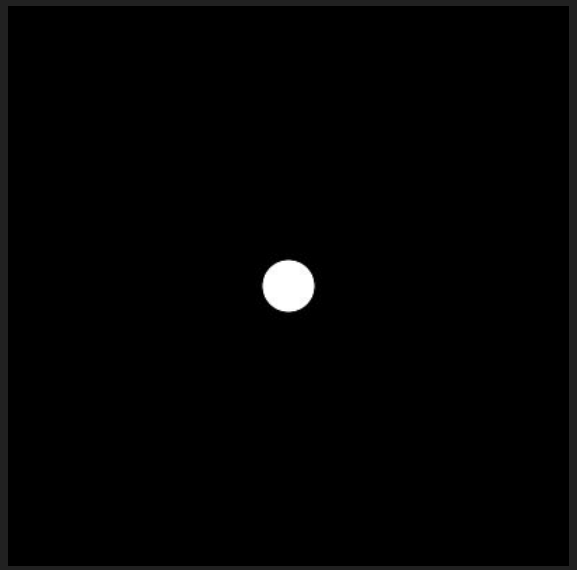
exp(-k*x)のここがスゴイ

- xの増加に従い0に漸近する（0になることはない）
- 速度変化も同じ形（ $d/dx\, e^{-kx} = -k\, e^{-kx}$ ）
- あらゆる自然現象でみられる曲線（指数関数的減衰）

アニメーション

expを使え

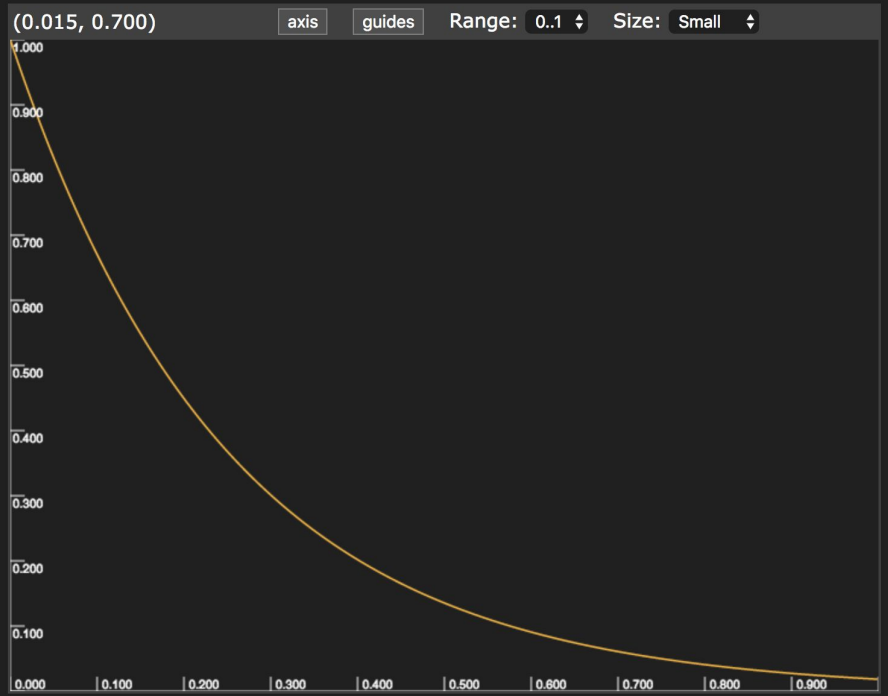
アニメーション



```
precision mediump float;
uniform float time;
uniform vec2 resolution;

void main(void) {
    vec2 p = (gl_FragCoord.xy * 2.0 - resolution) / resolution.y;

    // define animation
    float inner = 1.0 - exp(-5.0 * time);
    float outer = 1.0 - exp(-5.0 * (time - 0.2));

    // shape
    float len = length(p);
    float ssThr = 2.0 / resolution.y;
    float shape = smoothstep(0.0, ssThr, inner - len);
    shape *= smoothstep(0.0, ssThr, len - outer);

    gl_FragColor = vec4(vec3(shape), 1.0);
}
```

アニメーション

速度変化が急！



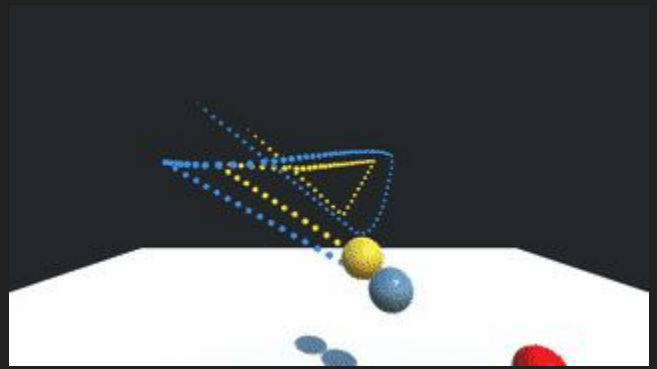
アニメーション

ほかにもいろいろアニメーション

アニメーション

Critically Damped Spring

https://github.com/keijiro/SmoothingTest

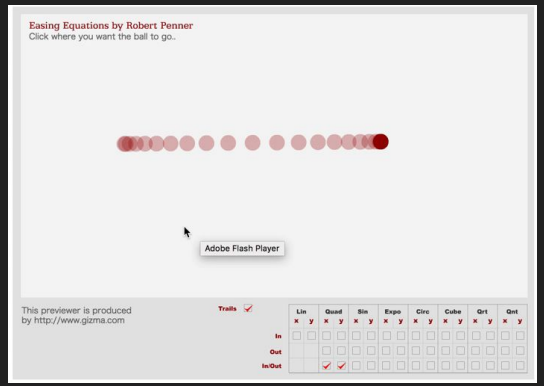


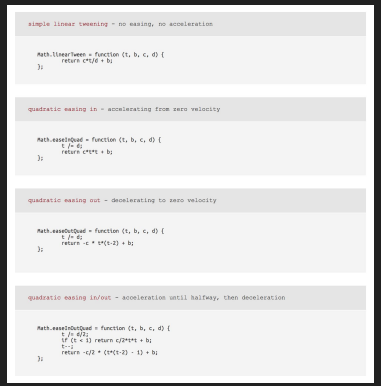
ステート保持が必要（位置・速度）

アニメーション

Easing Equations

http://gizma.com/easing/





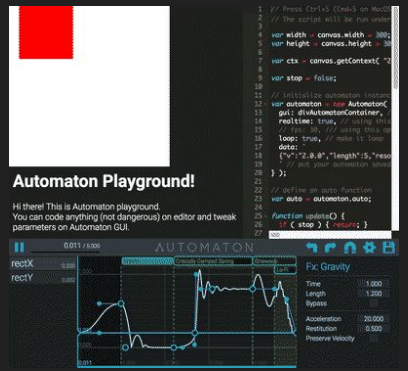
CSSのアレ

関数定義すればすぐ使える

アニメーション

Automaton

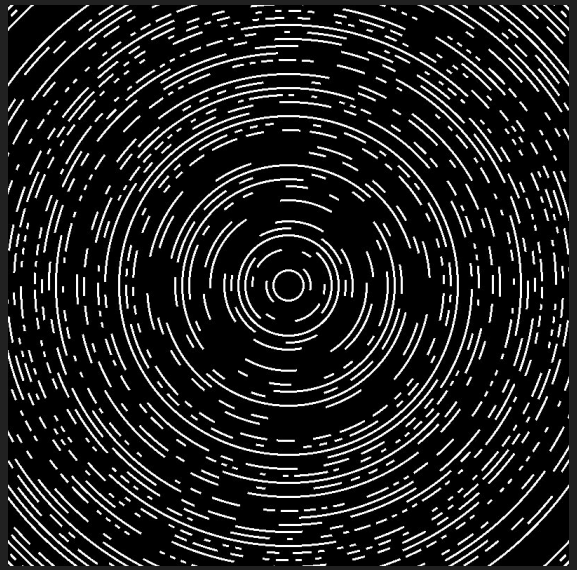
https://github.com/FMS-Cat/automaton

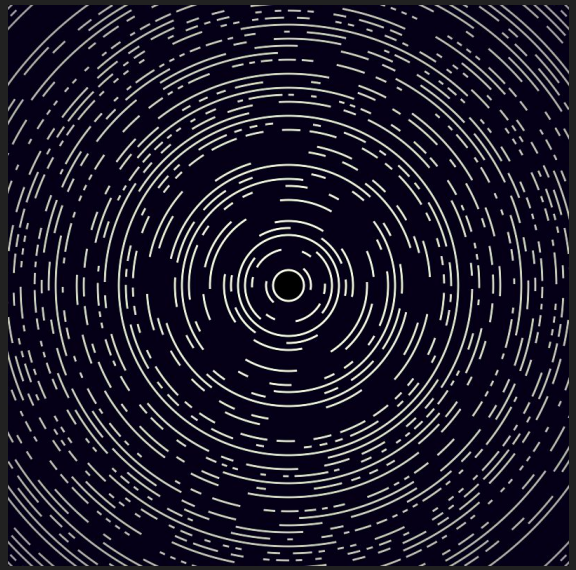


ベジエ曲線とFXでアニメーションしまくり

JavaScriptが得意なら是非！

色



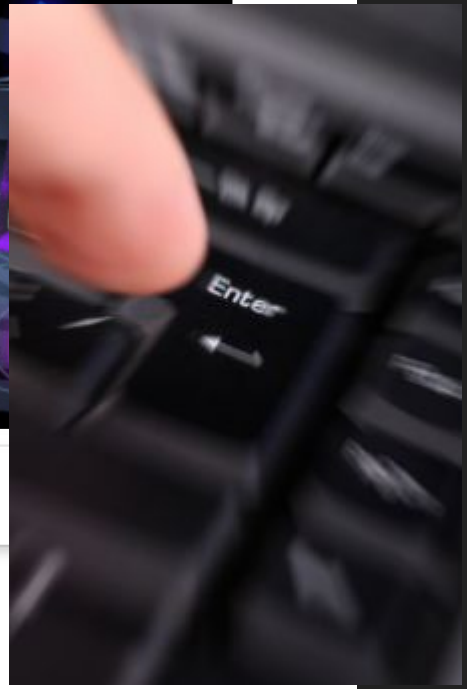


色

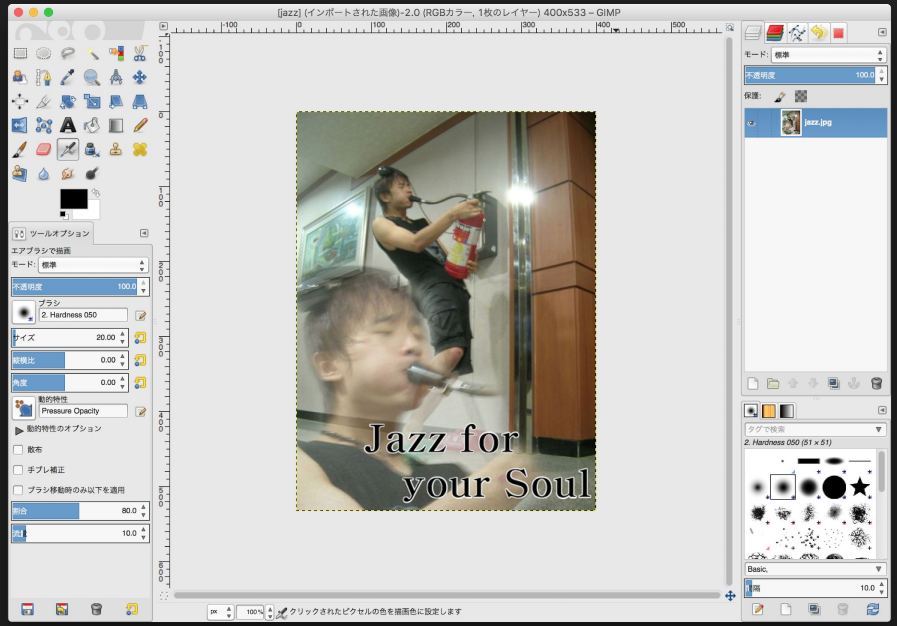
色

Instagram X-Pro II Photoshop

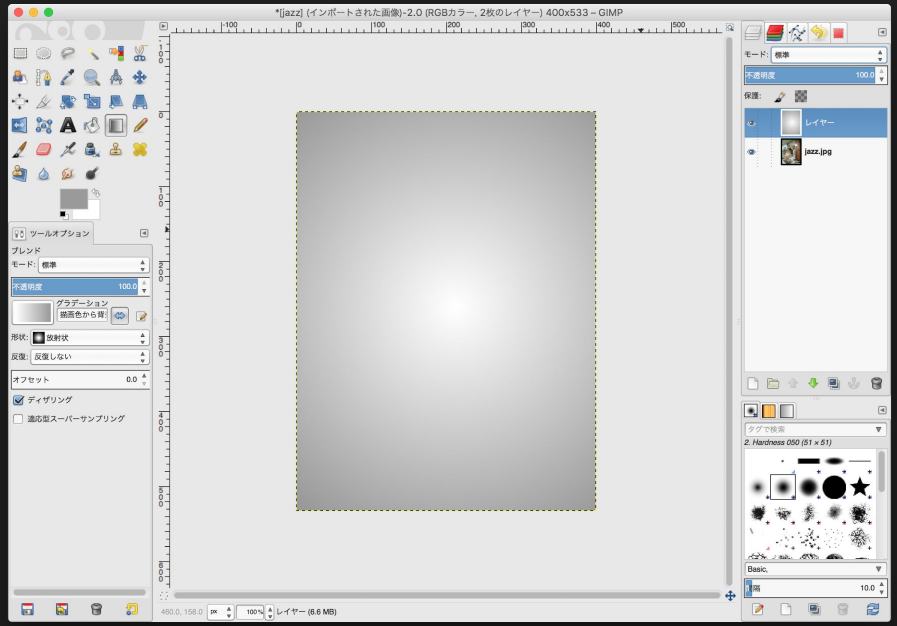
Google 検索　I'm Feeling Lucky



色

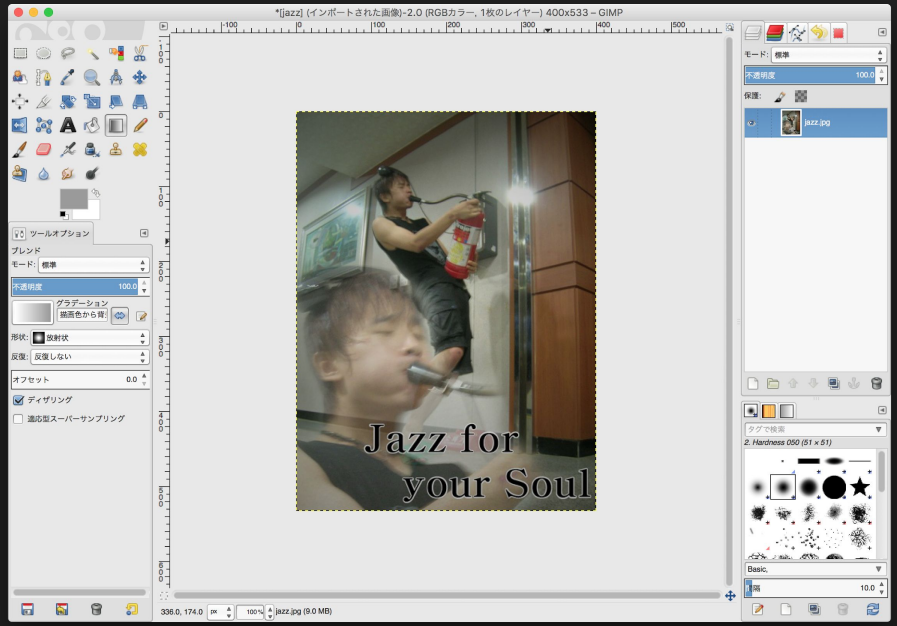

[jazz] (インポートされた画像)-2.0 (RGBカラー, 1枚のレイヤー) 400x533 – GIMP
Jazz for
your Soul
jazz.jpg

色



グラデーションを作成

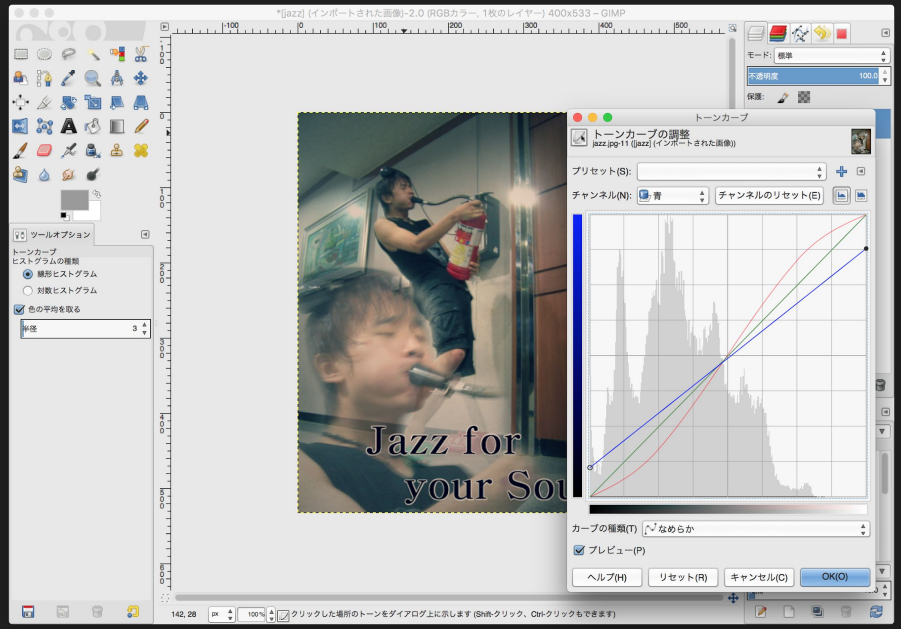
色



グラデーションを作成

乗算合成

色



グラデーションを作成

乗算合成

トーンカーブを適用

色

グラデーションを作成

乗算合成

トーンカーブを適用

色

これシェーダでできるよね

グラデーションを作成

乗算合成

トーンカーブを適用

色

```
col *= 1.0 - length( p ) * 0.2;
```

```
col = vec3(
    smoothstep( 0.0, 1.0, col.x ),
    col.y,
    0.1 + 0.8 * col.z
);
```

グラデーションを作成

乗算合成

トーンカーブを適用

色

おたくはすぐ 色収差

色

```
col.x = texture2D( samplerLenna, uv + vec2( 0.01, 0.0 ) ).x;
col.y = texture2D( samplerLenna, uv ).y;
col.z = texture2D( samplerLenna, uv - vec2( 0.01, 0.0 ) ).z;
```

色

```
col.x = texture2D( samplerLenna, 0.5 + 1.03 * ( uv - 0.5 ) ).x;
col.y = texture2D( samplerLenna, uv ).y;
col.z = texture2D( samplerLenna, 0.5 + 0.97 * ( uv - 0.5 ) ).z;
```

色


```
col.x = texture2D( samplerLenna, 0.5 + 1.03 * ( uv - 0.5 ) ).x;
col.y = texture2D( samplerLenna, uv ).y;
col.z = texture2D( samplerLenna, 0.5 + 0.97 * ( uv - 0.5 ) ).z;
```

色

おたく臭さが若干抜けた

色収差

色

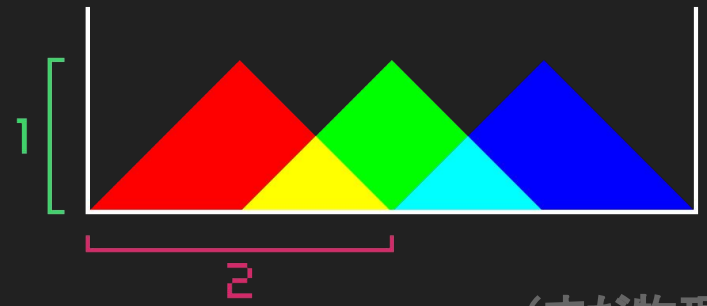
色

拡大方向にブラー

色

拡大方向にブラー

R, G, Bそれぞれの総和が1に
なるようにグラデーション



（まだ物理的に正しくない）

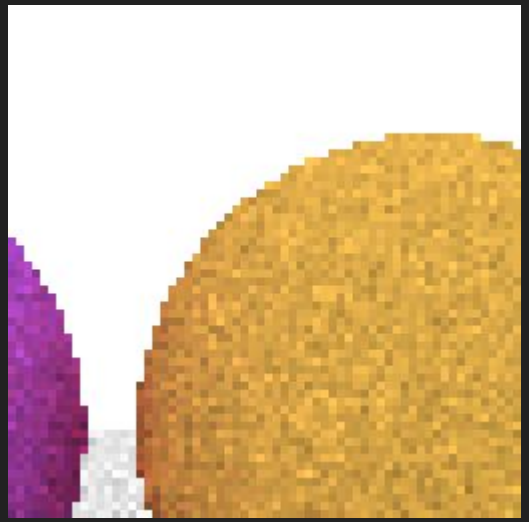
色

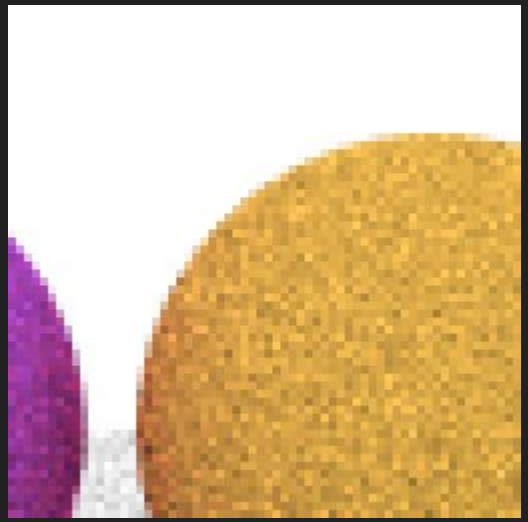
アンチエイリアス

色

アンチエイリアス

物体の縁に注目！





色

アンチエイリアス

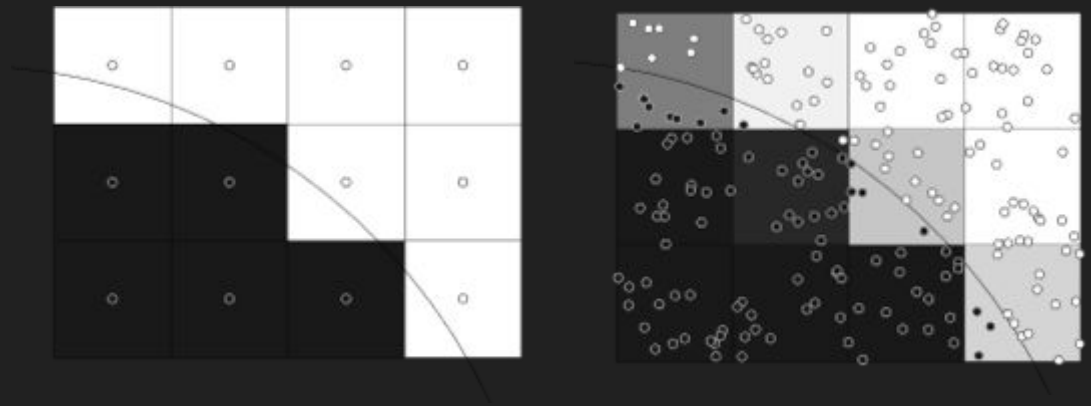
物体の縁に注目！

描画手法ごとに作戦が変わってくる

色

Supersampling AA (SSAA)

Oldschool, Bruteforce



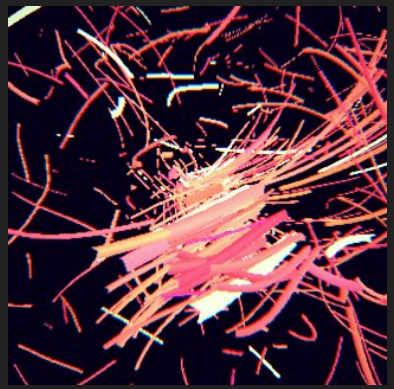
1ピクセル内でランダムサンプリング

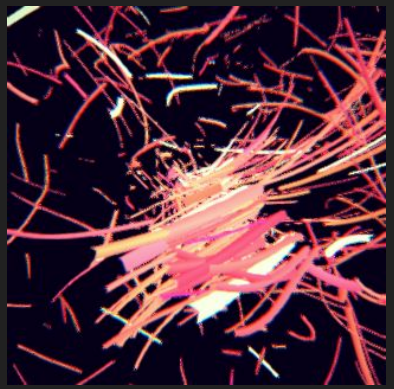
当然めっちゃ重い！

色

Fast Approximate AA (FXAA)

NVIDIA’s Post-Processing Anti-Aliasing



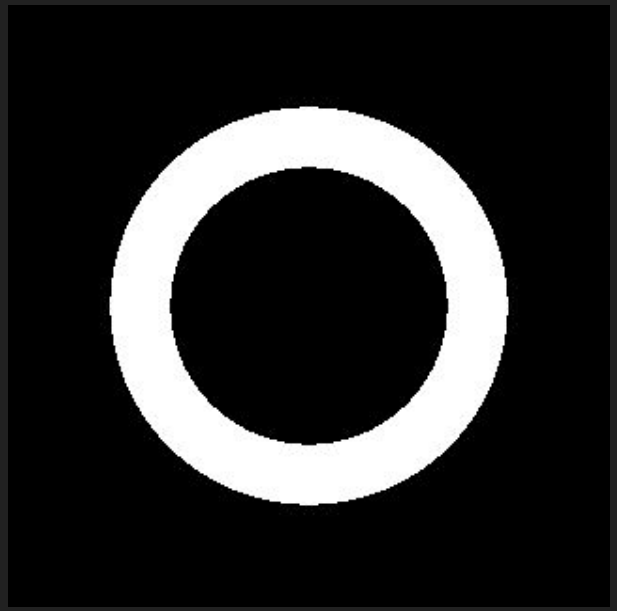


エッジ検出　→　ぼかし

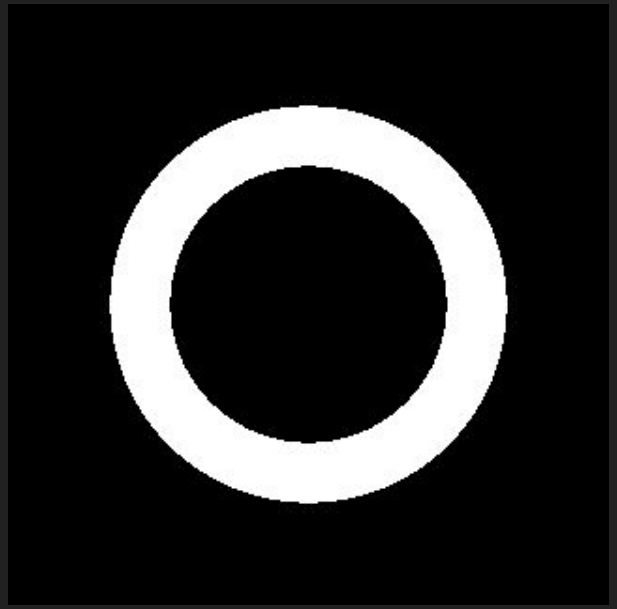
ポストエフェクトとして実装できる

色

境界を線形補間

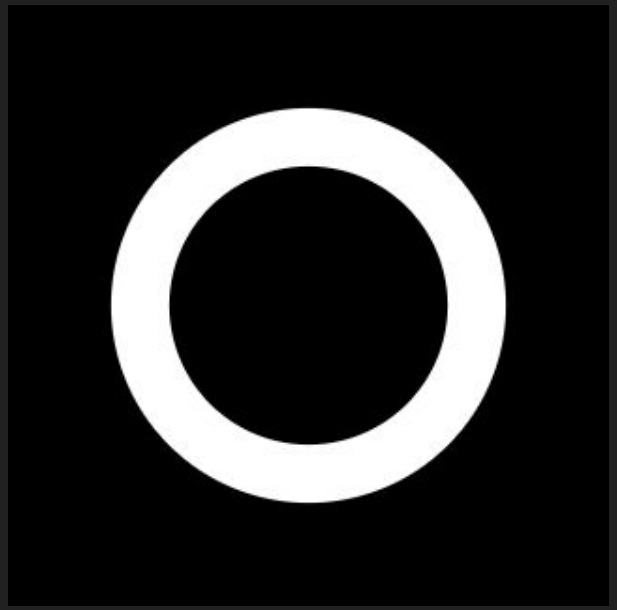


色



```
precision mediump float;
uniform float time;
uniform vec2 resolution;



void main(void) {
    vec2 p = (gl_FragCoord.xy * 2.0 - resolution) / resolution.y;

    // define animation
    float inner = 1.0 - exp(-5.0 * time);
    float outer = 1.0 - exp(-5.0 * (time - 0.2));

    // shape
    float len = length(p);

    float shape = step(len, inner);
    shape *= step(outer, len);

    gl_FragColor = vec4(vec3(shape), 1.0);
}
```

色



```
precision mediump float;
uniform float time;
uniform vec2 resolution;

#define saturate(x) clamp(x,0.,1.)
#define linearstep(a,b,x) saturate((x-a)/(b-a))

void main(void) {
    vec2 p = (gl_FragCoord.xy * 2.0 - resolution) / resolution.y;

    // define animation
    float inner = 1.0 - exp(-5.0 * time);
    float outer = 1.0 - exp(-5.0 * (time - 0.2));

    // shape
    float len = length(p);
    float ssThr = 2.0 / resolution.y;
    float shape = linearstep(0.0, ssThr, inner - len);
    shape *= linearstep(0.0, ssThr, len - outer);

    gl_FragColor = vec4(vec3(shape), 1.0);
}
```

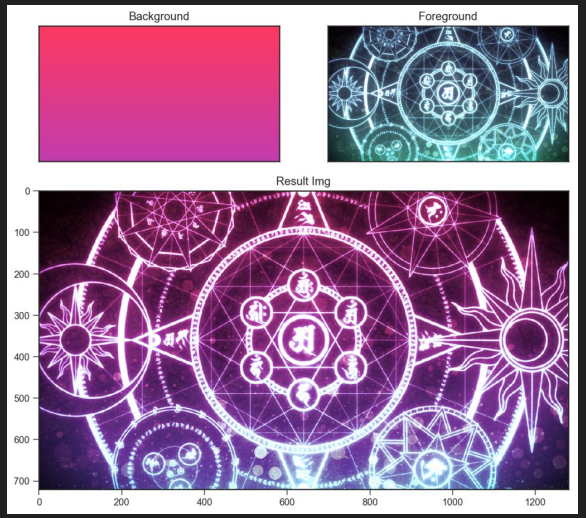
色

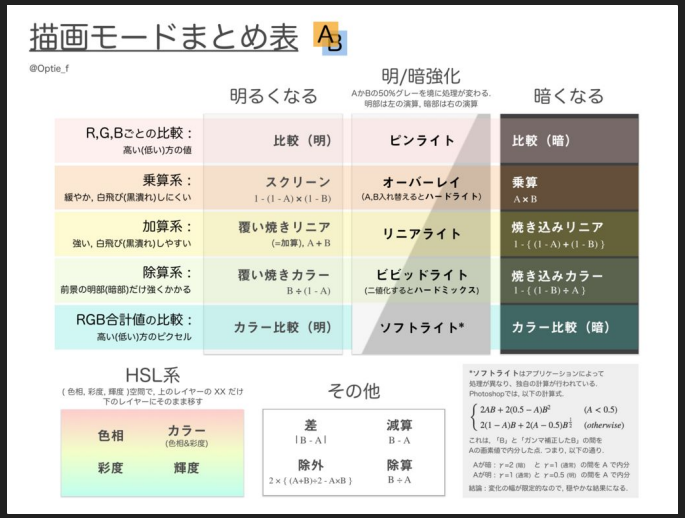
その他の色トピックス

色

描画モード

http://optie.hatenablog.com/entry/2018/03/15/212107

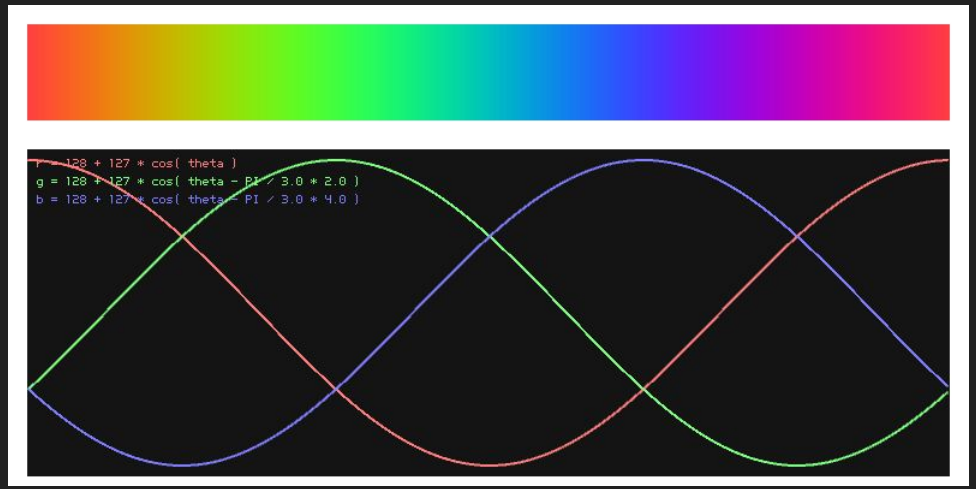




色

Sine Curve Gradient

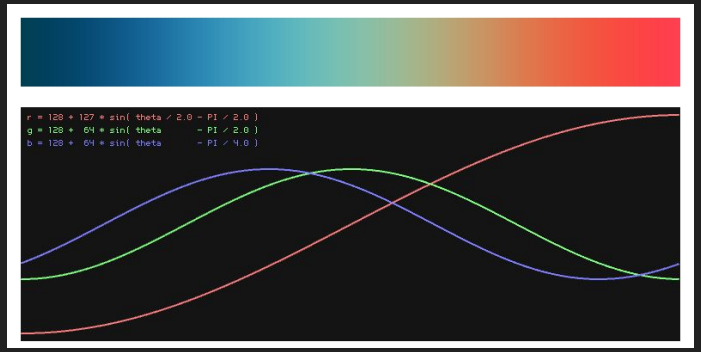
`catColor(float t)`

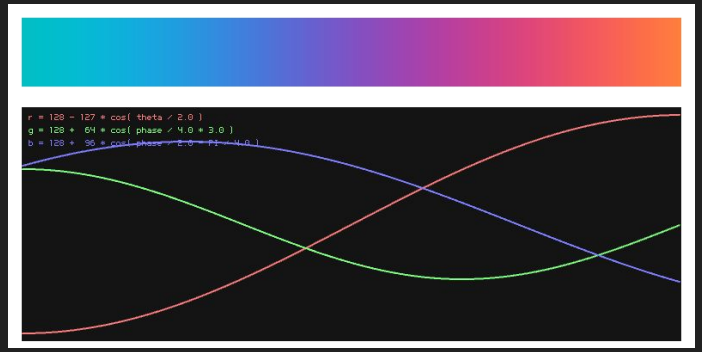


色

Sine Curve Gradient

other cases

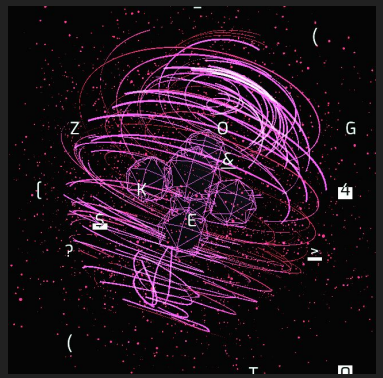


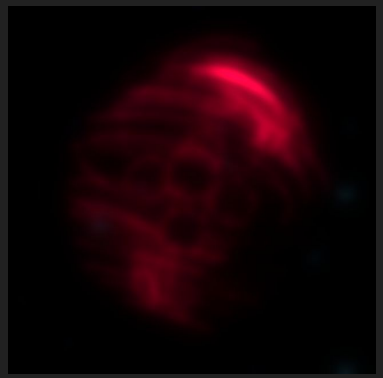


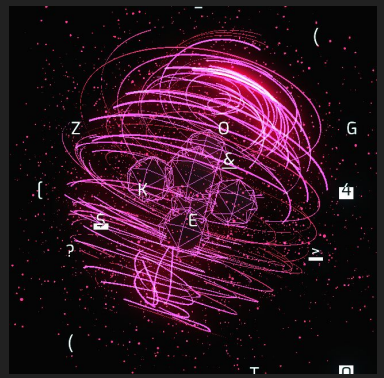
色

Bloom

まぶしい







明るいスポット抽出

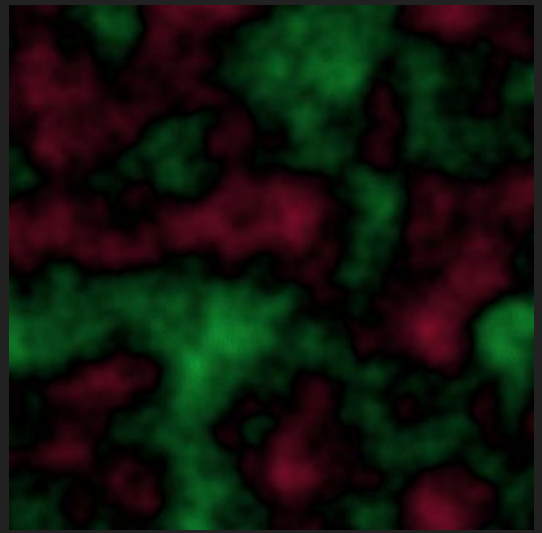
ぼかす

元の絵に足す

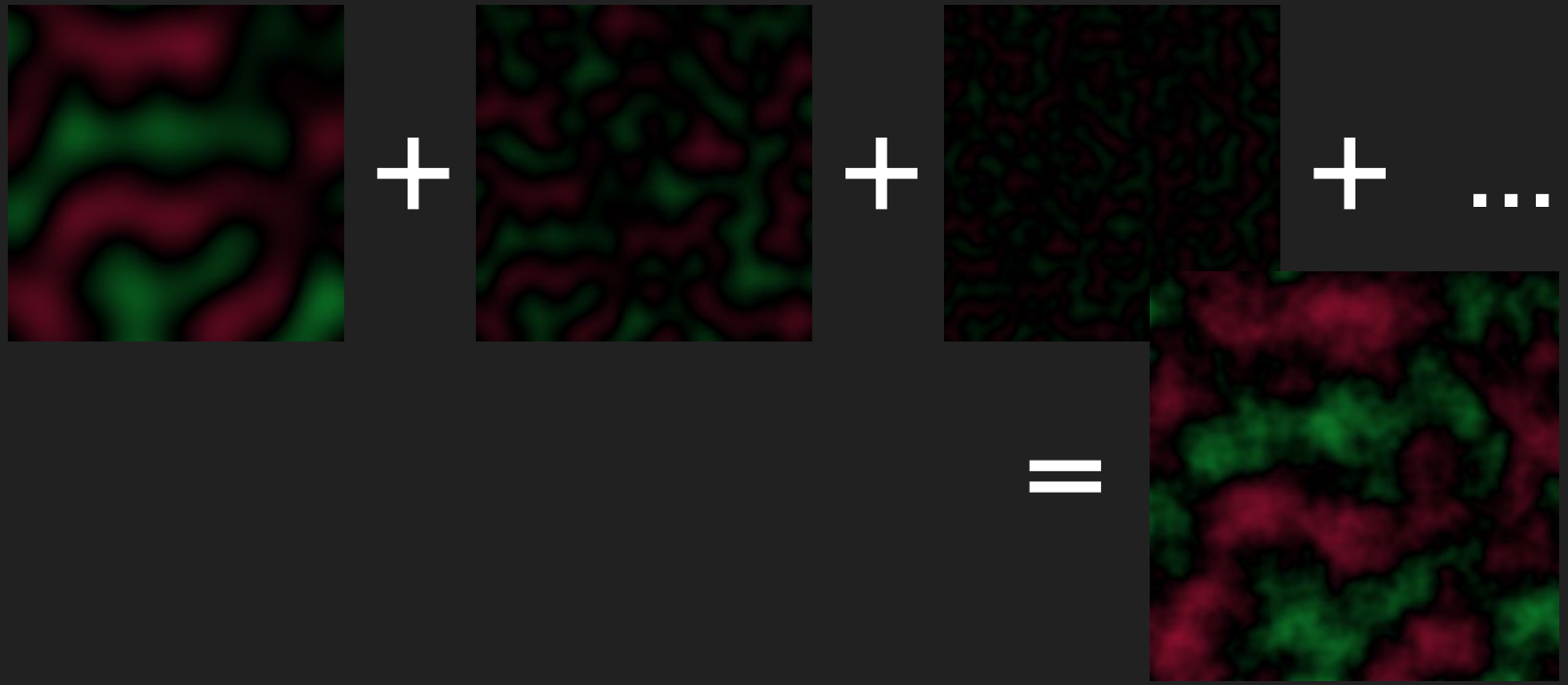
ノイズ

ノイズと一言で言うと
一般にはフラクタルノイズ（たぶん）

- 位相に応じてランダム
- でも連続的
- 有機的な表現に有用



ノイズ



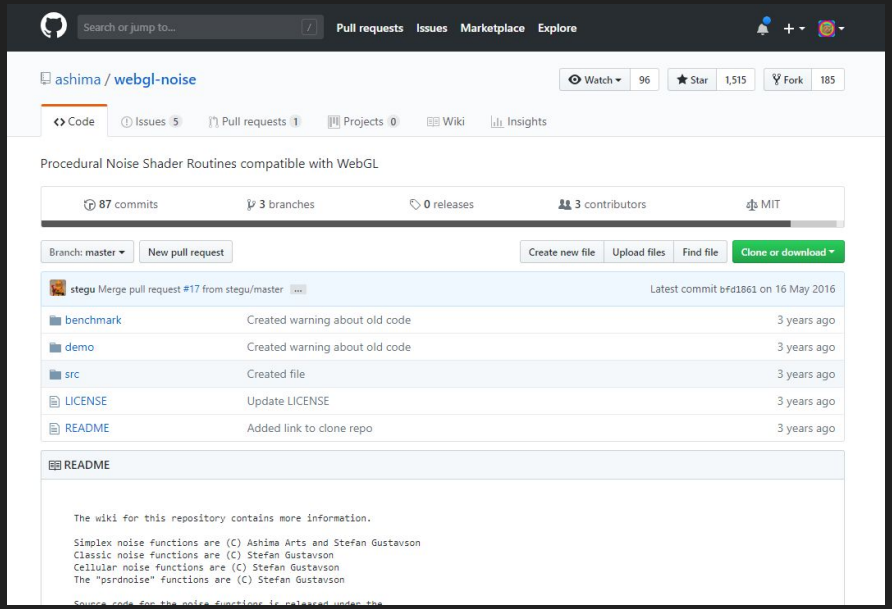
ノイズ

実装してみよう

```
float hash2D( vec2 v ) {
    return fract( sin( dot( v, vec2( 4.2787, 3.7915 ) ) ) * 2851.6893 );
}

float hash3D( vec3 v ) {
    return fract( sin( dot( v, vec3( 4.2787, 3.7915, 4.8911 ) ) ) * 2851.6893 );
}

float fade( float x ) {
    return ( ( 6.0 * x - 15.0 ) * x + 10.0 ) * x * x * x;
}

float perlin3D( vec3 v ) {
    vec3 vI = floor( v );
    vec3 vF = fract( v );

    float sum = 0.0;
    for ( int iz = 0; iz < 2; iz ++ ) {
        for ( int iy = 0; iy < 2; iy ++ ) {
            for ( int ix = 0; ix < 2; ix ++ ) {
                vec3 contF = vec3( ix, iy, iz );
                vec3 cont = vI + vec3( ix, iy, iz );
                vec3 cont2v = vF - contF;
                vec3 h = normalize( 2.0 * vec3(
                    hash3D( cont ),
```

ノイズ

https://github.com/ashima/webgl-noise



ノイズ

さて

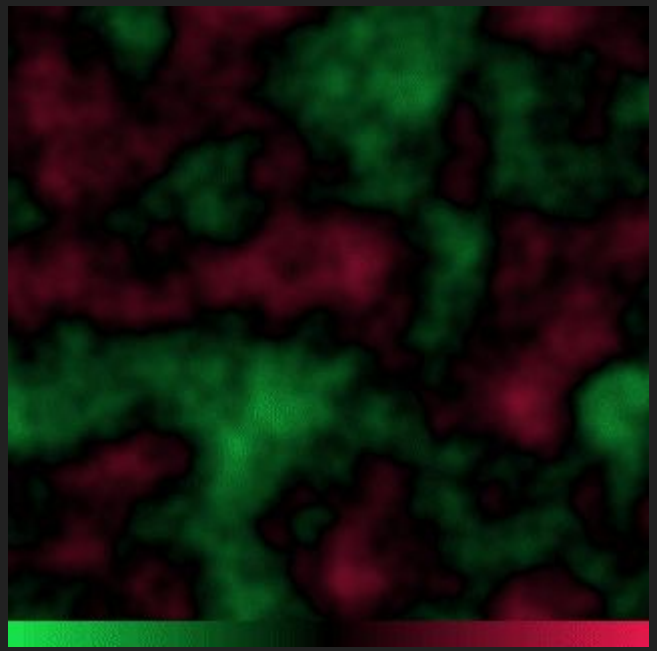
ノイズ

持っているのは……

- vec2 位置
- float 現在時刻
- vec3を渡すと模様が出る謎の関数
- （その他どうぐいろいろ）

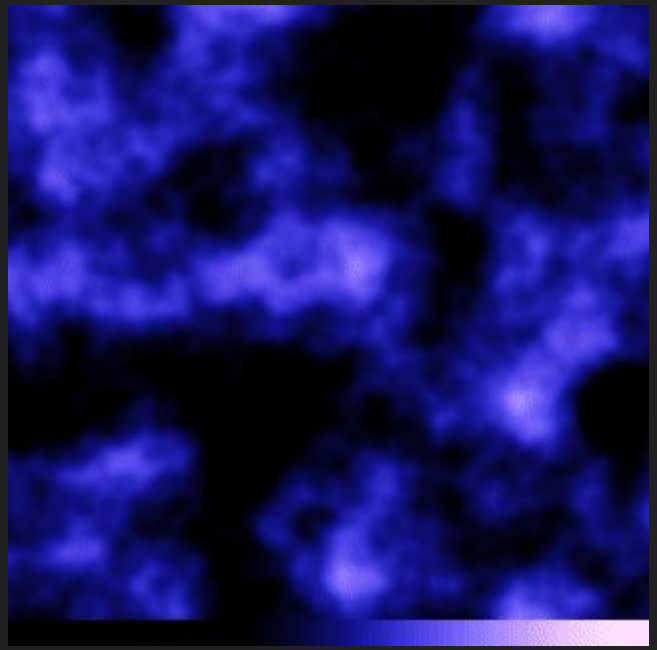
この3つを最大限活用して何ができるか？

ノイズ



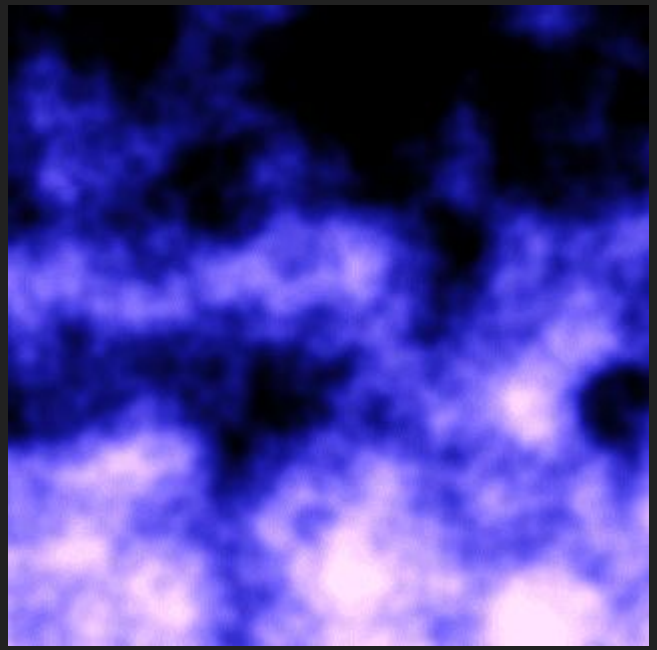
```
float sum = noise( uv, time );

vec3 col = vec3( 0.0 );
col += vec3( 0.9, 0.1, 0.3 ) * saturate( sum );
col += vec3( 0.1, 0.9, 0.3 ) * saturate( -sum );
```

ノイズ

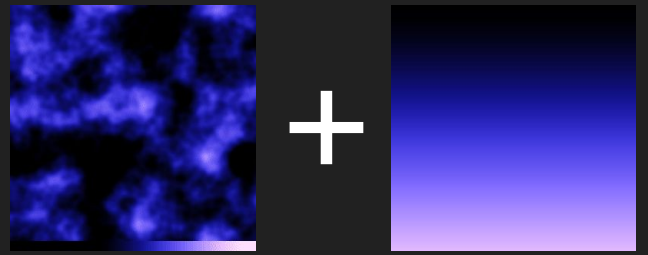


```
// purple gradient!
col = mix(
    vec3( 0.0, 0.0, 0.0 ),
    mix(
        vec3( 0.1, 0.1, 0.8 ),
        vec3( 1.1, 0.9, 1.3 ),
        smoothstep( 0.0, 1.0, sum )
    ),
    smoothstep( -0.3, 0.3, sum )
);
```

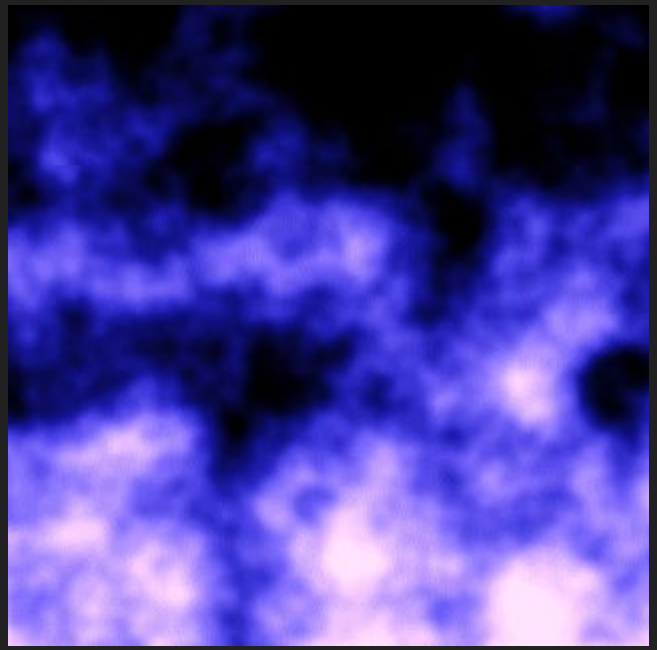
ノイズ



```
sum += 0.7 - uv.y;
```

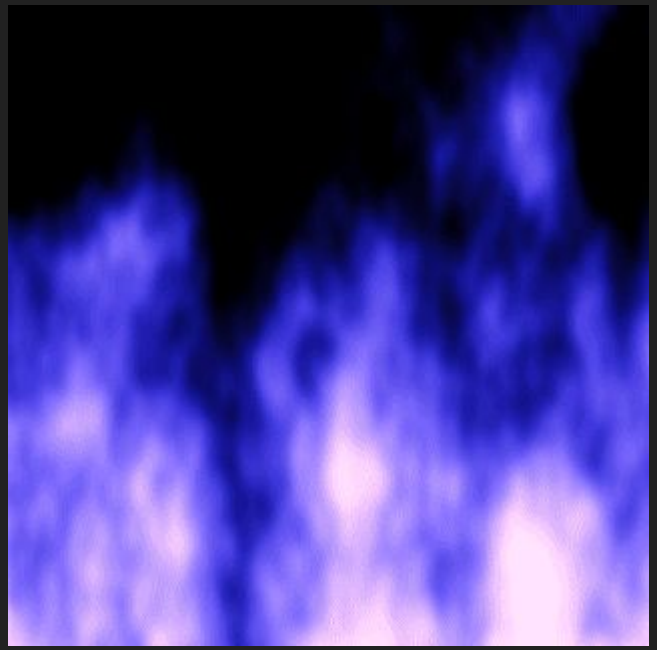


ノイズ



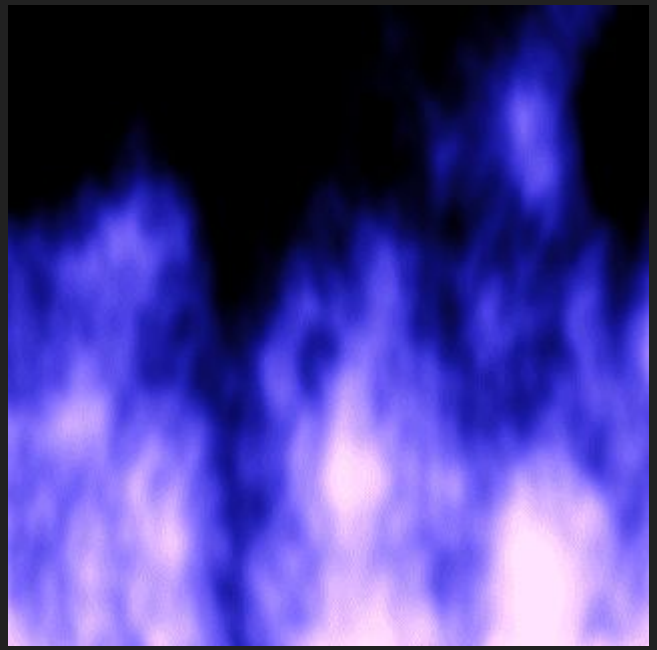
```
sum += 0.7 - uv.y;
uv.y -= 1.5 * time;
```

ノイズ

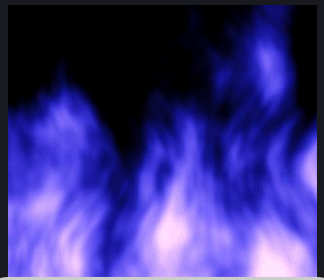


```
sum += 0.7 - uv.y;
uv.y *= 0.5;
uv.y -= 1.5 * time;
```

ノイズ

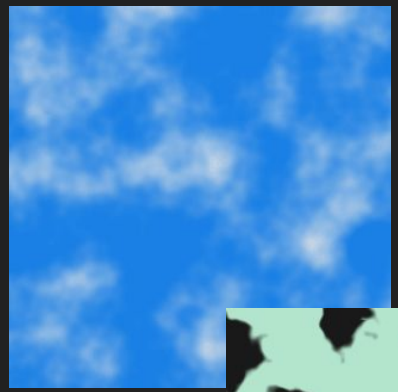
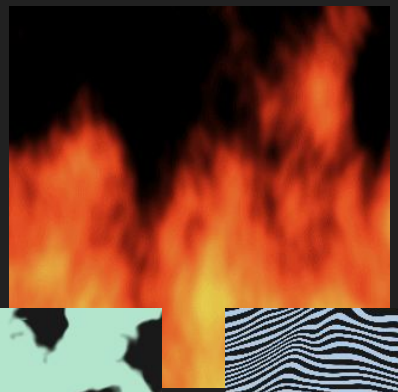


```
sum += 0.7 - uv.y;
uv.x += 0.01 * sin( 10.0 * uv.y ) * sin( 5.0 *
uv.x );
uv.y *= 0.5;
uv.y -= 1.5 * time;
```



やりすぎた場合

ノイズ

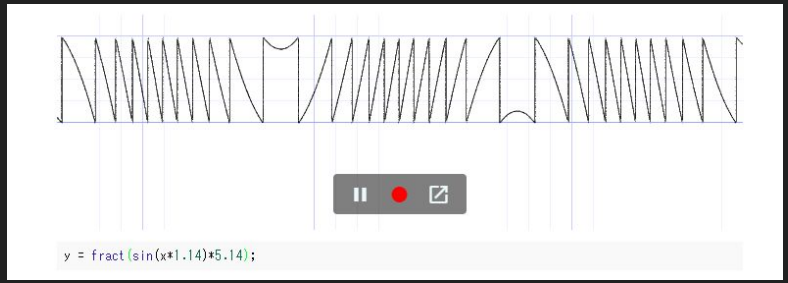


ノイズ

余談

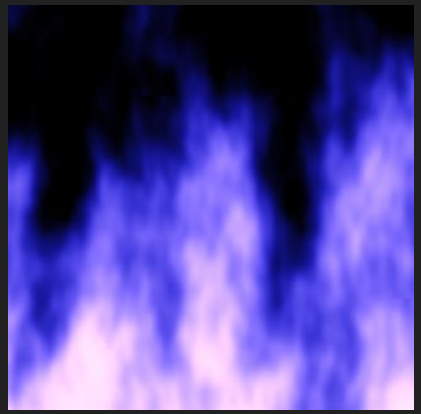
```
float hash3D( vec3 v ) {
    return fract( sin(
        dot( v, vec3( 1.14, 5.14, 1.919 ) )
    ) * 810.893 );
}
```

いわゆる “FractSin” [Ref.]

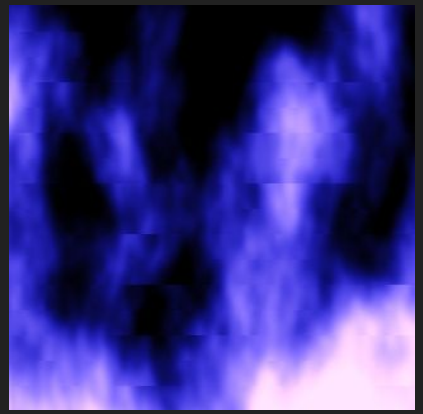


ノイズ

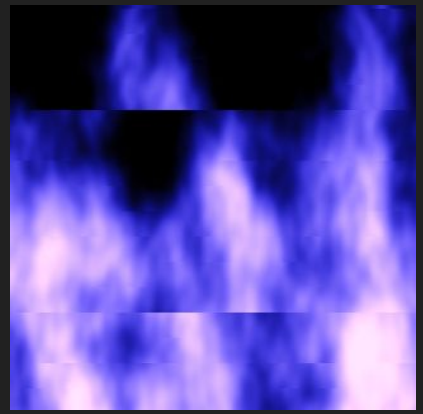
余談



time = 2.0



time = 20.0

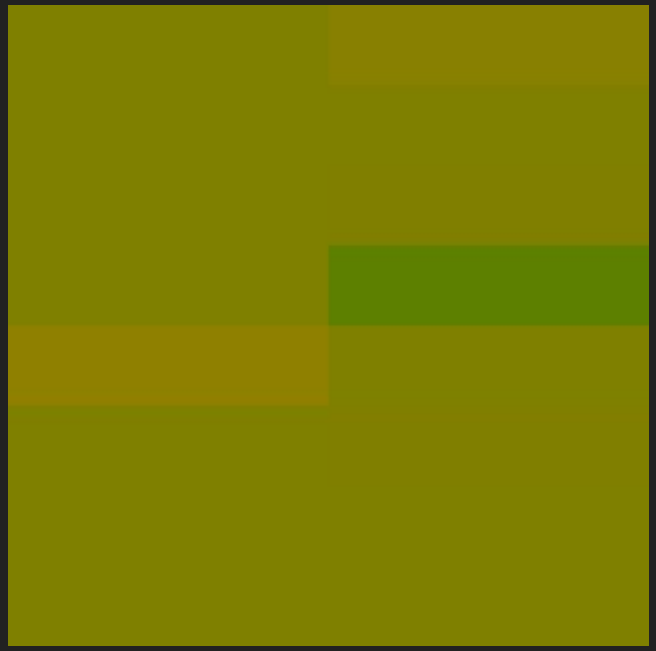


time = 200.0

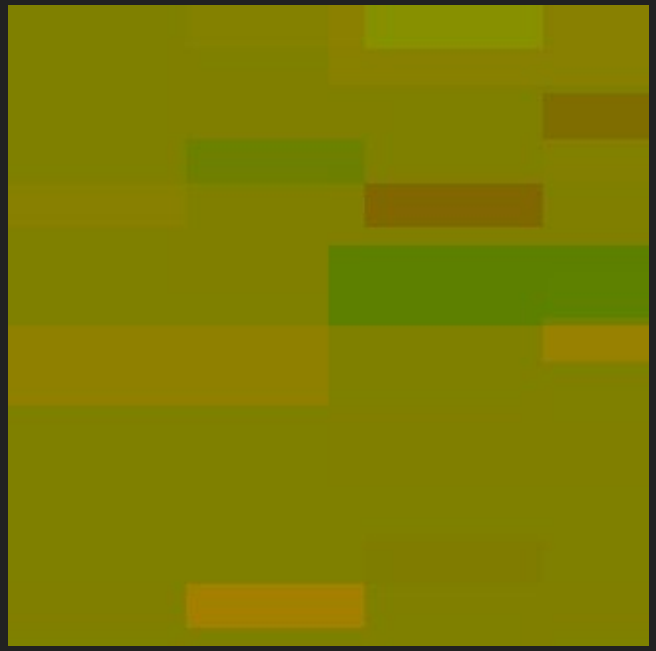
ノイズ

ノイズ

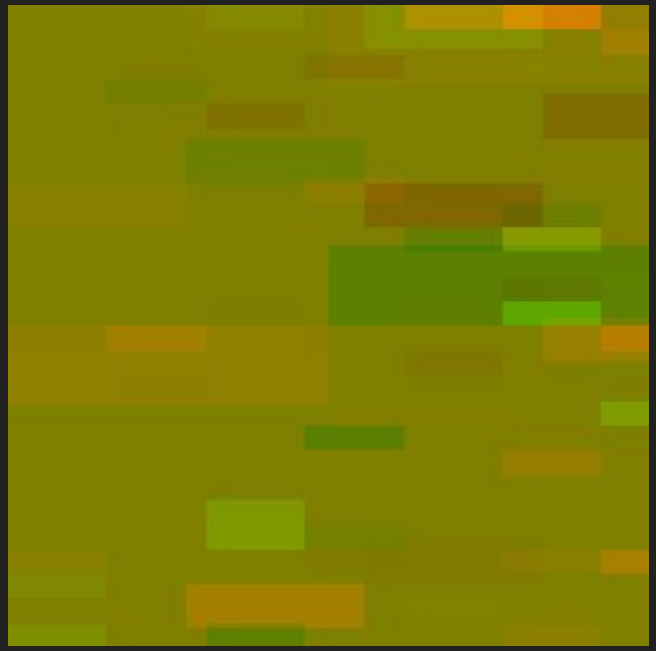
ノイズ



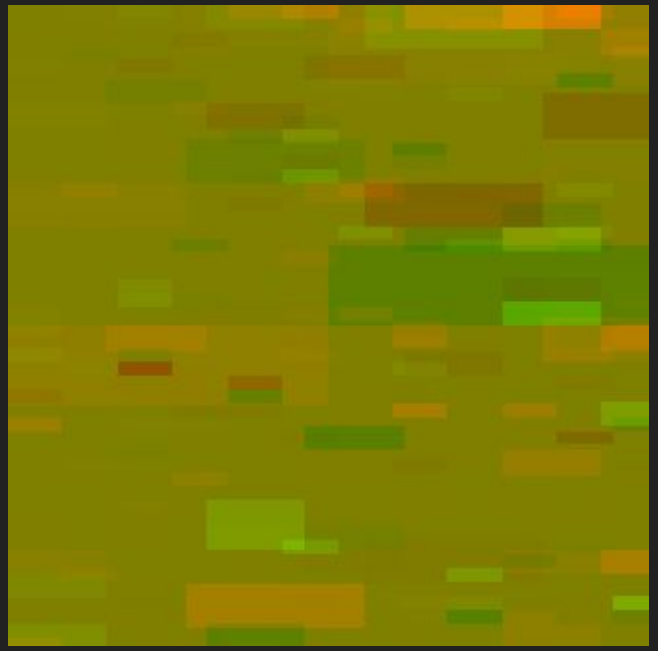
ノイズ



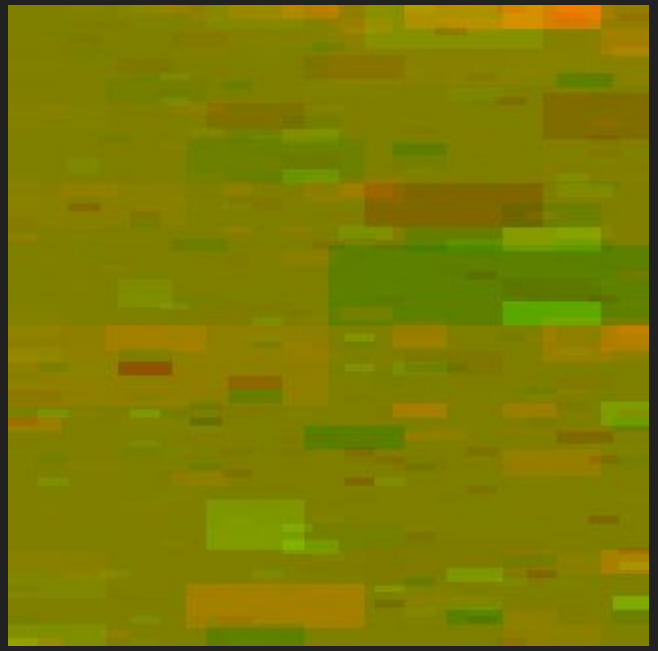
ノイズ



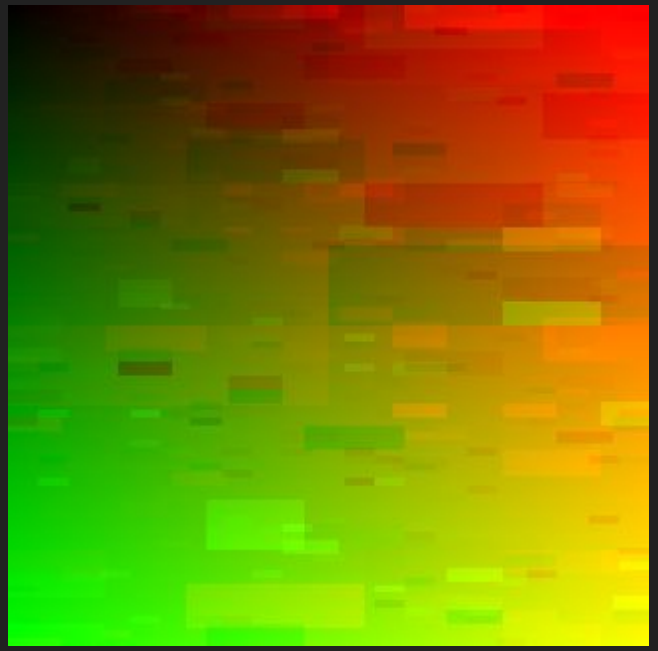
ノイズ



ノイズ



ノイズ



Sample

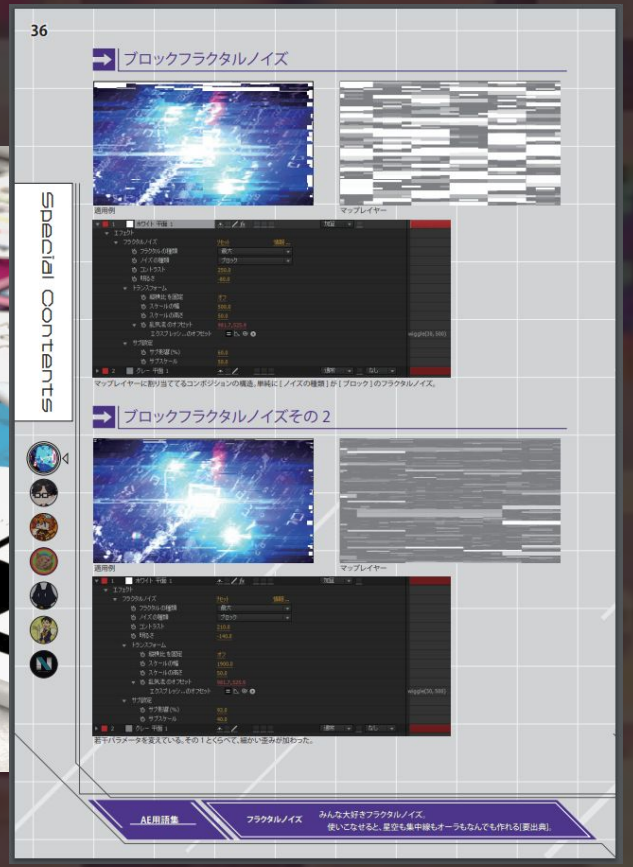
ノイズ

ノイズ

ノイズ

ノイズ


EOM
MAKINGBOOK
EverydayOneMotion メイキングブック

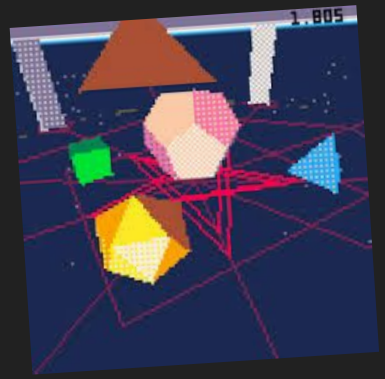

36
Special Contents
ブロックフラクタルノイズ
適用例
マップレイヤー
マップレイヤーに割り当ててるコンポジションの構造。単純に［ノイズの種類］が［ブロック］のフラクタルノイズ。
ブロックフラクタルノイズその２
適用例
マップレイヤー
若干パラメータを変えている。その１とくらべて、細かい歪みが加わった。
AE用語集
フラクタルノイズ
みんな大好きフラクタルノイズ。
使いこなせると、星空も集中線もオーラもなんでも作れる[要出典]。

割と本当に
伝えたかったこと

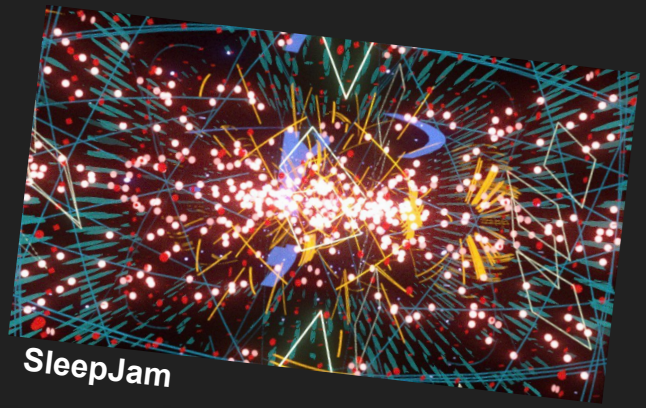
ほかの畑からパクれ

割と本当に
伝えたかったこと

百舌谷



rez



SleepJam

ほかの畑からパクれ

Keijiro Takahashi

beeple

割と本当に
伝えたかったこと

技術は違えど

表現手法・マインド

はパクれる

割と本当に
伝えたかったこと

割と本当に
伝えたかったこと

割と本当に
伝えたかったこと