

7 Lines GLSL Graphics Compoに挑戦してみよう！

- これはTokyo Demo Fest 2014(<http://tokyodemofest.jp/2014>) の7 Lines GLSL Graphics Compoのチュートリアルです。
- 7 Lines GLSL Graphics CompoではGLSLのFragment Shaderを用いて7行(7x80 characters)でグラフィックスを表示します。
- すべての作品は最新のGLSL sandboxで表示します。

GLSL sandboxについて

- GLSL sandboxについては以前にも記事(<http://www.demoscene.jp/?p=1154>)で紹介されました。
- GLSL sandboxはwebGLによりglslを用いてシェーダーを記述・確認できるサイトです。

やってみよう

<http://glsl.heroku.com> まず、はじめに"Create new effect!"を選択します。すると以下のようなコードが表示されます。

```

#ifdef GL_ES
precision mediump float;
#endif

uniform float time;
uniform vec2 mouse;
uniform vec2 resolution;

void main( void ) {
    vec2 position = ( gl_FragCoord.xy / resolution.xy ) +
mouse / 4.0;
    float color = 0.0;
    color += sin( position.x * cos( time / 15.0 ) * 80.0 )
+ cos( position.y * cos( time / 15.0 ) * 10.0 );
    color += sin( position.y * sin( time / 10.0 ) * 40.0 )
+ cos( position.x * sin( time / 25.0 ) * 40.0 );
    color += sin( position.x * sin( time / 5.0 ) * 10.0 )
+ sin( position.y * sin( time / 35.0 ) * 80.0 );
    color *= sin( time / 10.0 ) * 0.5;
    gl_FragColor = vec4( vec3( color, color * 0.5, sin( co
lor + time / 3.0 ) * 0.75 ), 1.0 );
}

```

ここで、最初の3行は行数には数えません。webGLで浮動小数点数の精度を指定するものですが、おまじないのようなものです。

```

#ifdef GL_ES
precision mediump float;
#endif

```

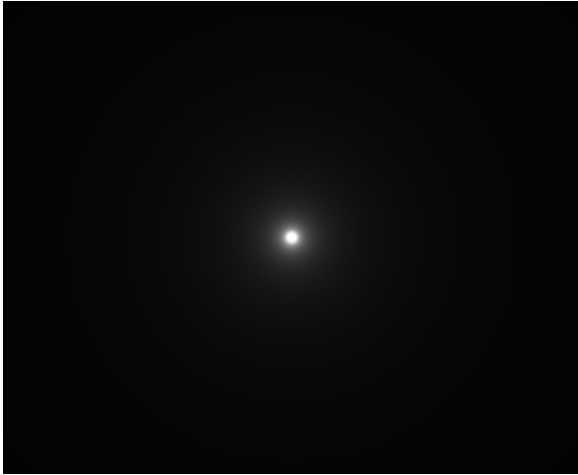
次の3行は外から画面への入力を与えるものです。

```

uniform float time;          //時間
uniform vec2 mouse;         //マウスの位置 (0~1の範囲)
uniform vec2 resolution;    //解像度 (ピクセル数)

```

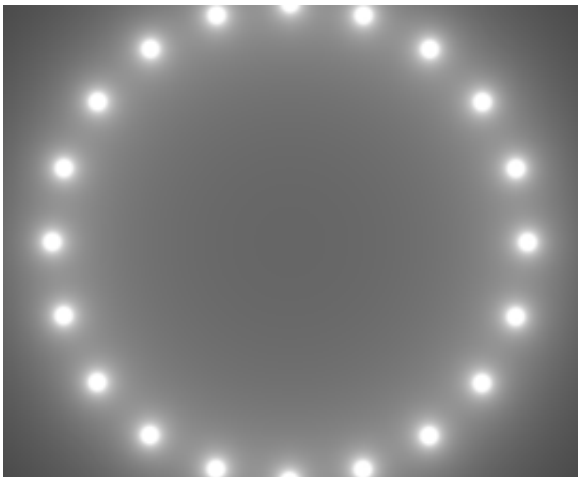
これらの入力から、最終的にgl_FragColorへ色を代入すればよいということになります。先のシェーダーコードは22行・最初の3行を除いても19行になります。ここから、まずは玉出してみましようか。玉。



<http://gls1.herokuapp.com/e#14932.1>

```
#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;
void main( void ) {
    vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x
,resolution.y);//-1~+1の座標系
    vec2 o = vec2(0,0);
    vec3 c = 0.01/(length(p-o))*vec3();
    gl_FragColor = vec4(c,1);
}
```

ほんじゃまあ20個くらいサクッと出してみましようか



<http://glsl.heroku.com/e#14932.2>

```
#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;
void main( void ) {
    vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x
,resolution.y);//
    vec3 c=vec3(0);
    for(int i=0;i<20;i++){
        float x = .5*cos(2.*3.14*float(i)/20.);
        float y = .5*sin(2.*3.14*float(i)/20.);
        vec2 o = vec2(x,y);
        c += 0.01/(length(p-o))*vec3(1);
    }
    gl_FragColor = vec4(c,1);
}
```

なんか味気ないですね。じゃあtimeを使ってアニメーションさせてみましょう。

```
#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;
void main( void ) {
    vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x
,resolution.y);//
    vec3 c=vec3(0);
    for(int i=0;i<20;i++){
        float t = 2.*3.14*float(i)/20. * fract(time*0.5);
        float x = cos(t);
        float y = sin(t);
        vec2 o = .5*vec2(x,y);
        c += 0.01/(length(p-o))*vec3(1);
    }
    gl_FragColor = vec4(c,1);
}
```

以上です。

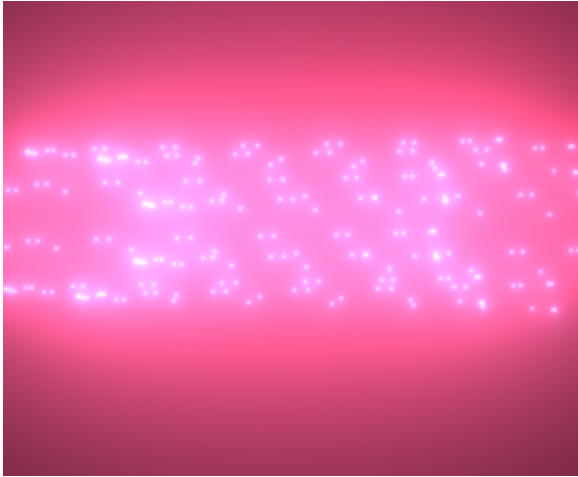
単純な円形ではつまらないので、8の字を書いてみましょう



<http://glsl.heroku.com/e#14932.5>

```
#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;
void main( void ) {
    vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x,resolution.y);//
    vec3 c=vec3(0);
    for(int i=0;i<20;i++){
        float t = 2.*3.14*float(i)/20. * fract(time*0.5);
        float x = cos(t);
        float y = sin(2.*t);
        vec2 o = .5*vec2(x,y);
        c += 0.01/(length(p-o))*vec3(1);
    }
    gl_FragColor = vec4(c,1);
}
```

mouse変数を使ってインタラクティブに動かせるようにしてみましょう。



<http://glsI.herokuapp.com/e#14932.7>

```

#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;
float l = .5*(sin(6.28*fract(time))+.5); //ループ用変数
void main(void ) {
    vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x
,resolution.y); //
    vec3 c=vec3(0);
    for(int i=0;i<30;i++){
        float t = 6.28*fract(float(i)/20.*(0.5*1+.8)+0.3 +.
1*mouse.y) ;
        float tt = 6.28*float(i)/20.;
        float x = (1.+mouse.x)*cos(t);
        float y = sin(5.0*mouse.x*t);
        for(int j=0;j<8;j++){
            float s = 6.28*(.3*fract(time)+float(j)/8.);
            float dx = 0.4*cos(s);
            float dy = 0.4*sin(6.28*fract(time))*sin(s);
            vec2 o = .5*vec2(x+dx,y+dy);
            float r = 0.5*t;
            float g = l;
            c += 0.001/(length(p-o))*vec3(r,g,1);
        }
    }
    vec3 d = vec3(0.1-0.1*1*mod(gl_FragCoord.y,2.));
    gl_FragColor = vec4(c-d,1);
}

```

さて、大体絵的にはこれくらいいいでしょう。大体普通に書いて20行くらいなら、最終的な7行にも収まりやすいようです。まずは、いらぬ空白やコメントを削除します。

<http://glsl.heroku.com/e#14932.9>

```

#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;
float l=.5*(sin(6.28*fract(time))+.5);
void main(){
    vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x
,resolution.y);
    vec3 c=vec3(0);
    for(int i=0;i<30;i++){
        float t=6.28*fract(float(i)/20.*(0.5*l+.8))+0.3+.1*mouse.y);
        float tt=6.28*float(i)/20.;
        float x=(1.+mouse.x)*cos(t);
        float y=sin(5.0*mouse.x*t);
        for(int j=0;j<8;j++){
            float s=6.28*(.3*fract(time)+float(j)/8.);
            float dx=0.4*cos(s);
            float dy=0.4*sin(6.28*fract(time))*sin(s);
            vec2 o=.5*vec2(x+dx,y+dy);
            float r=0.5*t;
            float g=1;
            c += 0.001/(length(p-o))*vec3(r,g,1);
        }
    }
    vec3 d=vec3(0.1-0.1*l*mod(gl_FragCoord.y,2.));
    gl_FragColor=vec4(c-d,1);
}

```

無駄な変数を削除します。

<http://glsl.heroku.com/e#14932.11>


```

#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;
float l=.5*(sin(6.28*fract(time))+.5),m=mouse.x,n=mouse.y;
void main(){
    vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x
,resolution.y);
    vec3 c=vec3(0);
    for(int i=0;i<30;i++){
        float t=6.28*fract(float(i)/20.*(l+.8))+.3+.1
*n);
        for(int j=0;j<8;j++){
            float s=6.28*(.3*fract(time)+float(j)/8.);
            vec2 o=.5*vec2((1+m)*cos(t)+.4*cos(s),sin(5.0*
m*t)+.4*sin(6.28*fract(time))*sin(s));
            c += 0.001/(length(p-o))*vec3(.5*t,1,1);
        }
    }
    vec3 d=vec3(.1-.1*l*mod(gl_FragCoord.y,2.));
    gl_FragColor=vec4(c-d,1);
}

```

あとはきっちり7行（最初の3行も入れて10行）にするだけです。ちなみに1行は79文字です。こんなになりました。



<http://gls.herokuapp.com/e#14932.14>

```
#ifdef GL_ES
precision mediump float;
#endif
uniform float time;uniform vec2 mouse;uniform vec2 resolution;float l=.5*(sin(6.28*fract(time))+.5),m=mouse.x,n=mouse.y;void main(){vec2 p=(gl_FragCoord.xy-.5*resolution)/min(resolution.x,resolution.y);vec3 c=vec3(0);for(int i=0;i<30;i++){float t=6.28*fract(float(i)/20.*(l+.8)+.3+.1*n);for(int j=0;j<8;j++){float s=6.28*(.3*fract(time)+float(j)/8.);vec2 o=.5*vec2((1+m)*cos(t)+.4*cos(s),sin(5.*m*t)+.4*sin(6.28*fract(time))*sin(s));c+=.001/(length(p-o))*vec3(.5*t,1,1);}}vec3 d=vec3(.1-.1*l*mod(gl_FragCoord.y,2.));gl_FragColor=vec4(c-d,1);}
```

最後に

もっと短くなるような気もしますが、今回はこんなところでどうでしょう。

7行というと短いようで書くことを厳選すると多い気がします。難しくはありません。

みなさんもチャレンジしてみてください。