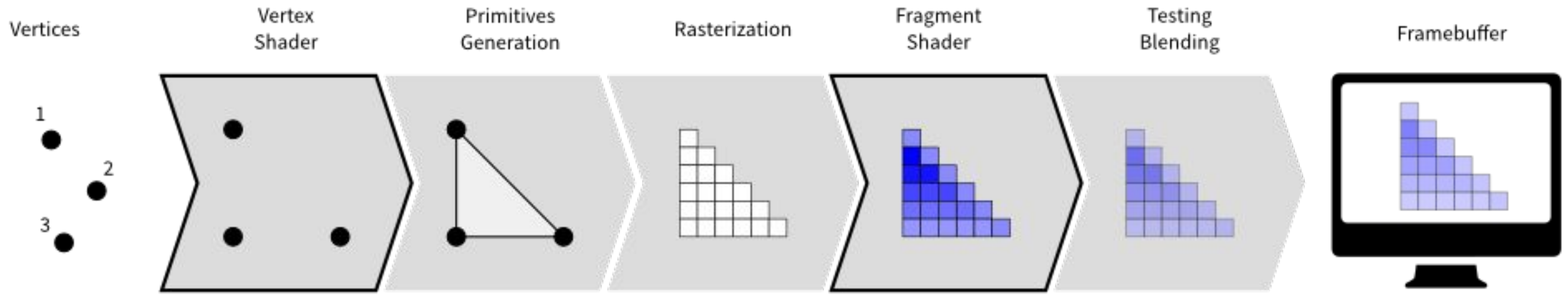
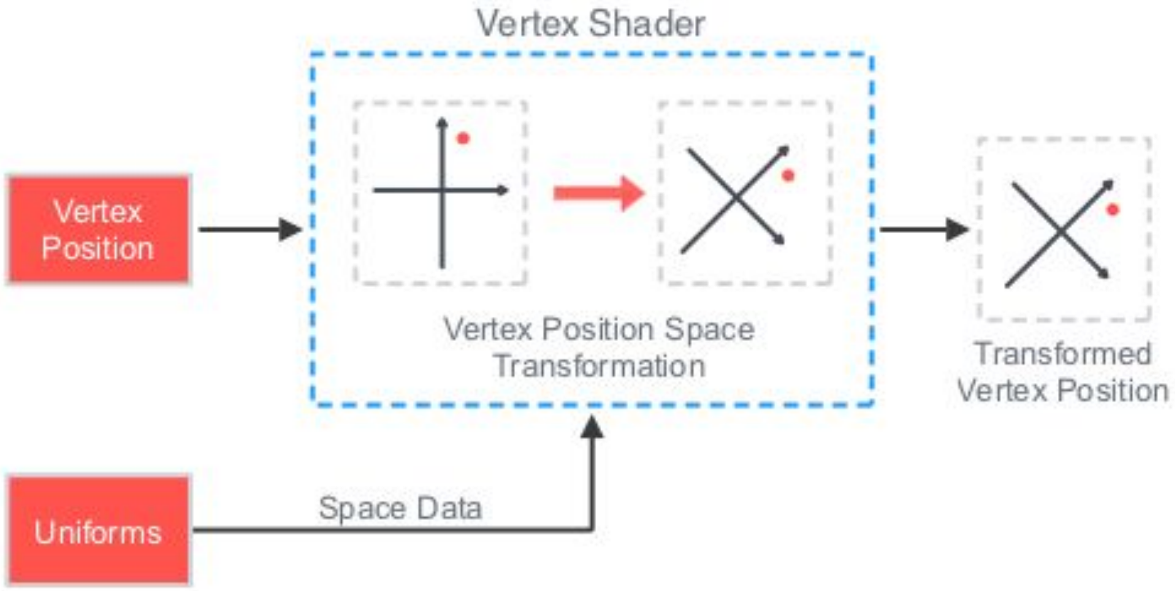
Clase 3: Shaders

Mandelbulb

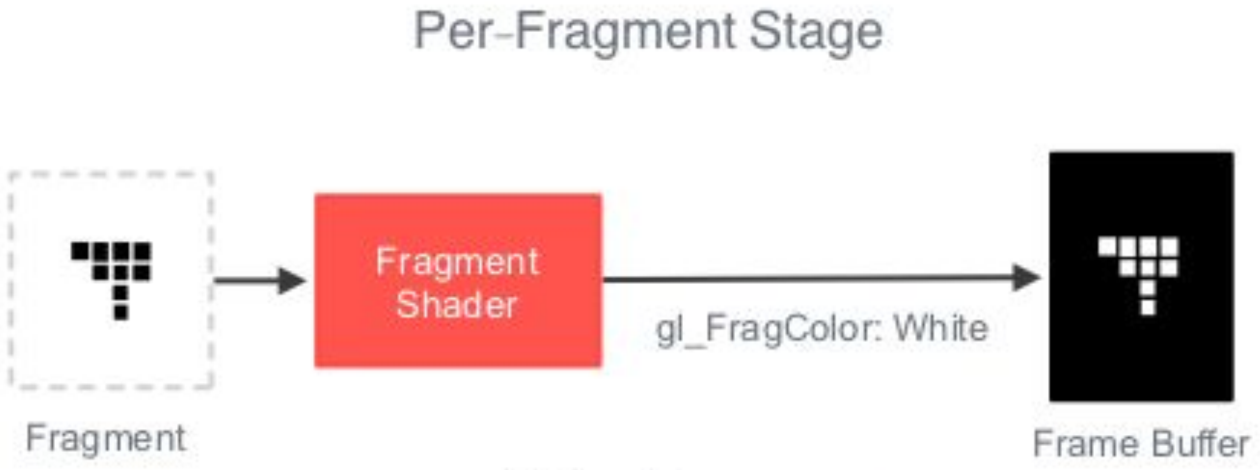
¿Qué vimos la clase pasada?



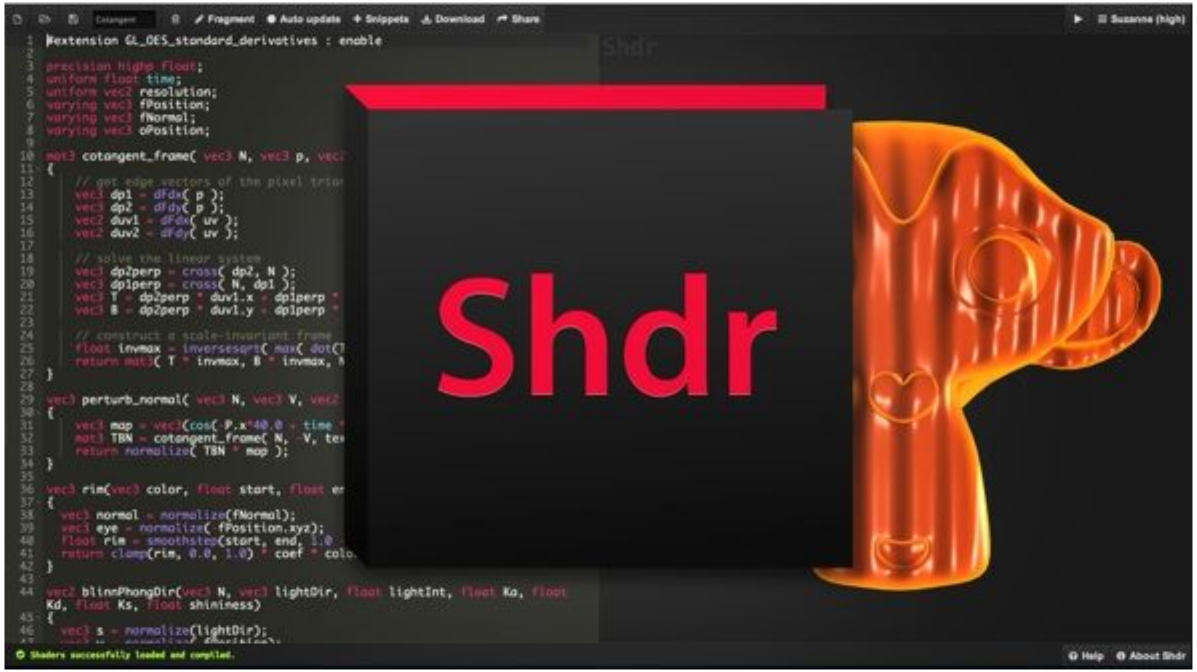
¿Qué vimos la clase pasada?


Vertex Shader
Vertex Position
Vertex Position Space Transformation
Transformed Vertex Position
Uniforms
Space Data

¿Qué vimos la clase pasada?



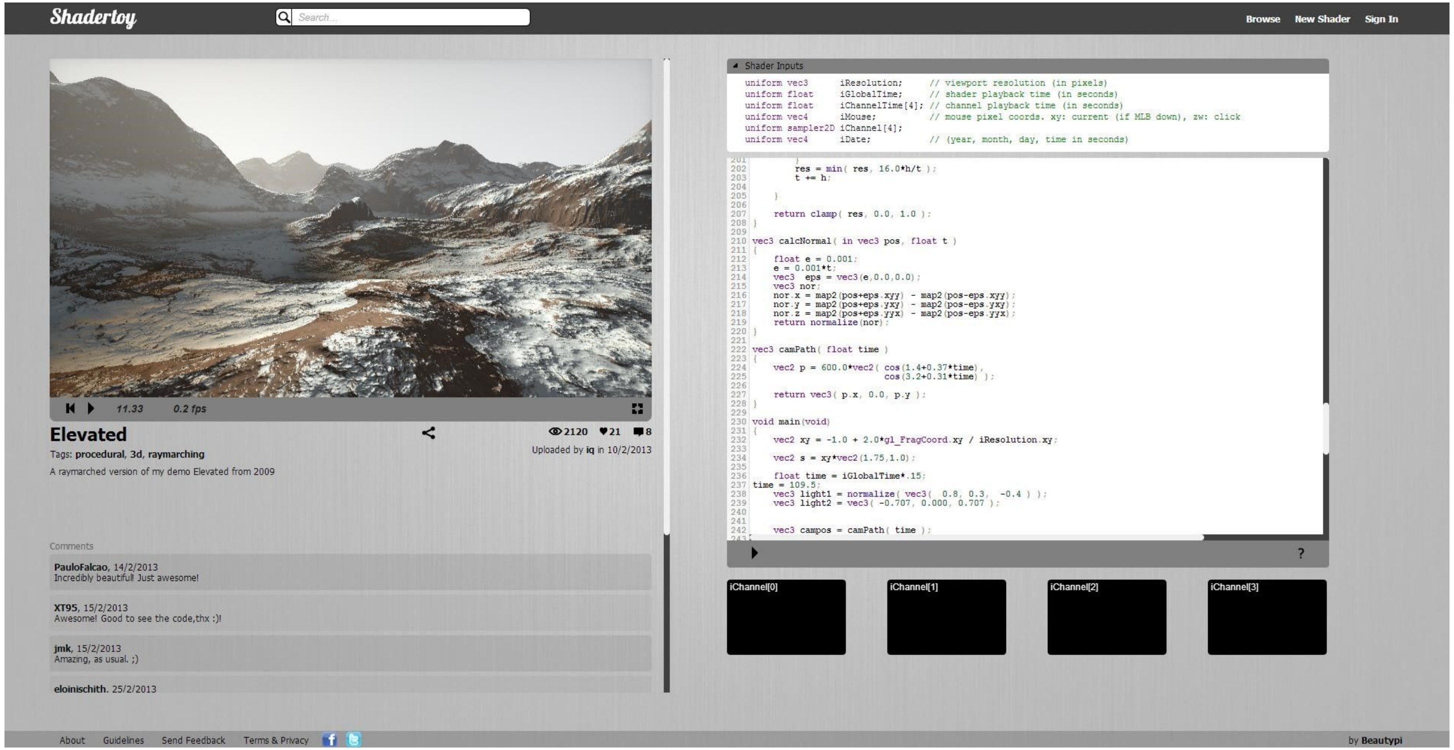
Herramientas online



```
#extension GL_OES_standard_derivatives : enable

precision highp float;
uniform float time;
uniform vec2 resolution;
varying vec3 fPosition;
varying vec3 fNormal;
varying vec3 oPosition;

mat3 cotangent_frame( vec3 N, vec3 p, vec2
{
    // get edge vectors of the pixel trian
    vec3 dp1 = dFdx( p );
    vec3 dp2 = dFdy( p );
    vec2 duv1 = dFdx( uv );
    vec2 duv2 = dFdy( uv );

    // solve the linear system
    vec3 dp2perp = cross( dp2, N );
    vec3 dp1perp = cross( N, dp1 );
    vec3 T = dp2perp * duv1.x + dp1perp *
    vec3 B = dp2perp * duv1.y + dp1perp *

    // construct a scale-invariant frame
    float invmax = inversesqrt( max( dot(T
    return mat3( T * invmax, B * invmax, N
}

vec3 perturb_normal( vec3 N, vec3 V, vec2
{
    vec3 map = vec3(cos(-P.x*40.0 + time *
    mat3 TBN = cotangent_frame( N, -V, tex
    return normalize( TBN * map );
}

vec3 rim(vec3 color, float start, float en
{
  vec3 normal = normalize(fNormal);
  vec3 eye = normalize(-fPosition.xyz);
  float rim = smoothstep(start, end, 1.0
  return clamp(rim, 0.0, 1.0) * coef * colo
}

vec2 blinnPhongDir(vec3 N, vec3 lightDir, float lightInt, float Ka, float
Kd, float Ks, float shininess)
{
  vec3 s = normalize(lightDir);
```

Shaders successfully loaded and compiled.

Herramientas online



Elevated

Tags: **procedural**, **3d**, **raymarching**

Uploaded by **iq** in 10/2/2013

A raymarched version of my demo Elevated from 2009

Comments

PauloFalcao, 14/2/2013
Incredibly beautiful! Just awesome!

XT95, 15/2/2013
Awesome! Good to see the code,thx :)!

jmk, 15/2/2013
Amazing, as usual. ;)

eloinischith, 25/2/2013

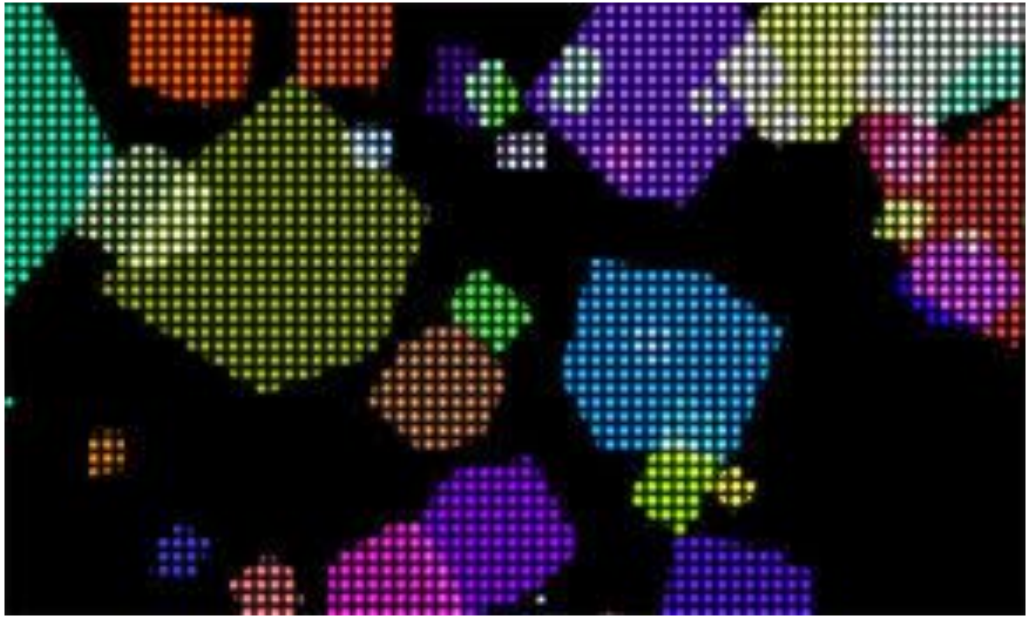
```
uniform vec3      iResolution;     // viewport resolution (in pixels)
uniform float     iGlobalTime;     // shader playback time (in seconds)
uniform float     iChannelTime[4]; // channel playback time (in seconds)
uniform vec4      iMouse;          // mouse pixel coords. xy: current (if MLB down), zw: click
uniform sampler2D iChannel[4];
uniform vec4      iDate;           // (year, month, day, time in seconds)
```

```
        }
        res = min( res, 16.0*h/t );
        t += h;

    }

    return clamp( res, 0.0, 1.0 );
}

vec3 calcNormal( in vec3 pos, float t )
{
    float e = 0.001;
    e = 0.001*t;
    vec3  eps = vec3(e,0.0,0.0);
    vec3 nor;
    nor.x = map2(pos+eps.xyy) - map2(pos-eps.xyy);
    nor.y = map2(pos+eps.yxy) - map2(pos-eps.yxy);
    nor.z = map2(pos+eps.yyx) - map2(pos-eps.yyx);
    return normalize(nor);
}

vec3 camPath( float time )
{
    vec2 p = 600.0*vec2( cos(1.4+0.37*time),
                         cos(3.2+0.31*time) );

    return vec3( p.x, 0.0, p.y );
}

void main(void)
{
    vec2 xy = -1.0 + 2.0*gl_FragCoord.xy / iResolution.xy;

    vec2 s = xy*vec2(1.75,1.0);

    float time = iGlobalTime*.15;
time = 109.5;
    vec3 light1 = normalize( vec3(  0.8, 0.3,  -0.4 ) );
    vec3 light2 = vec3( -0.707, 0.000, 0.707 );


    vec3 campos = camPath( time );
```

¿Cómo funcionan estas herramientas?

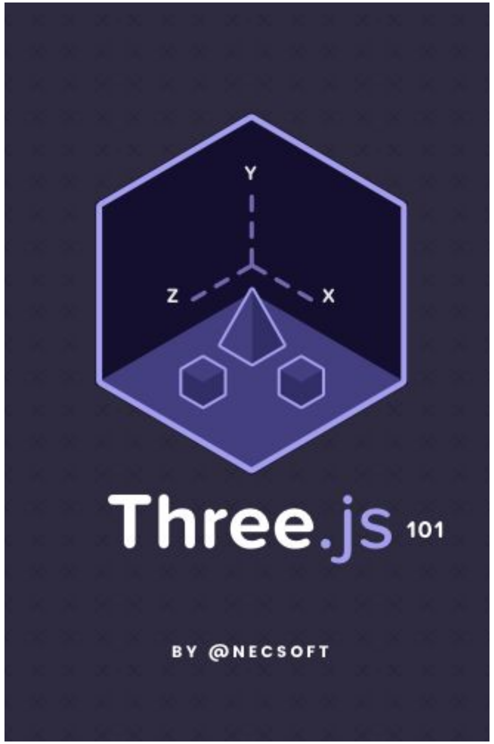
WebGL

Pixel Shaders



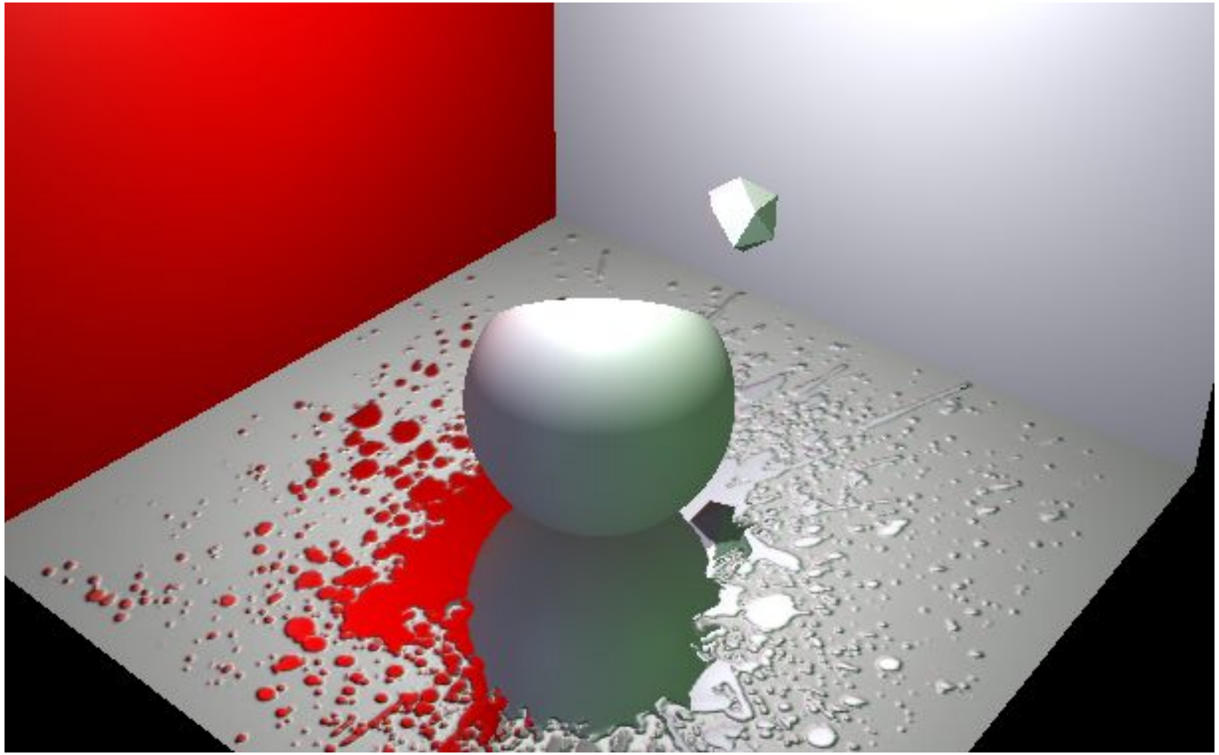
¿Cómo funcionan estas herramientas?

WebGL™



Ejemplos threejs

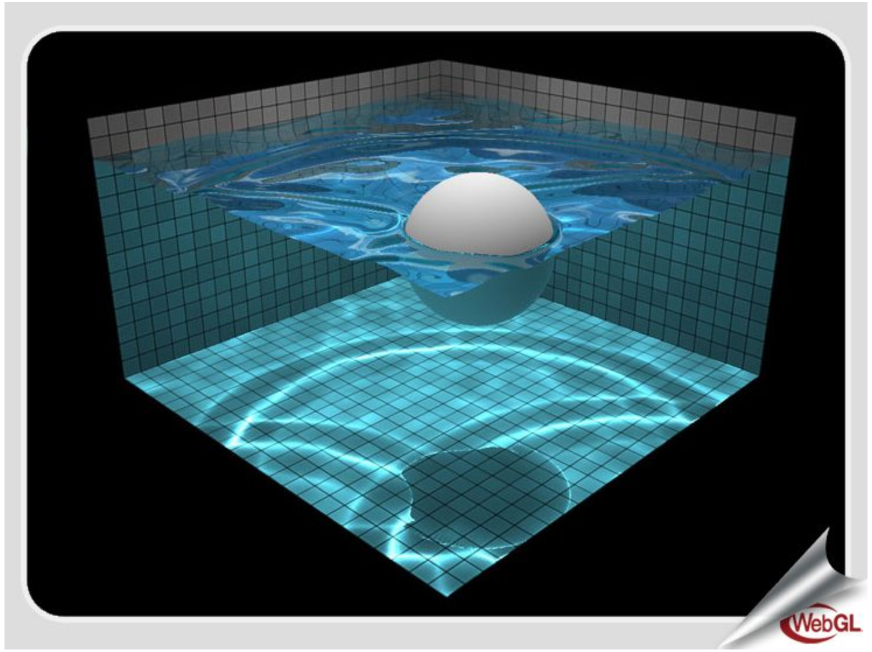
https://threejs.org/examples



¿Cómo funcionan estas herramientas?

Vertex Shader

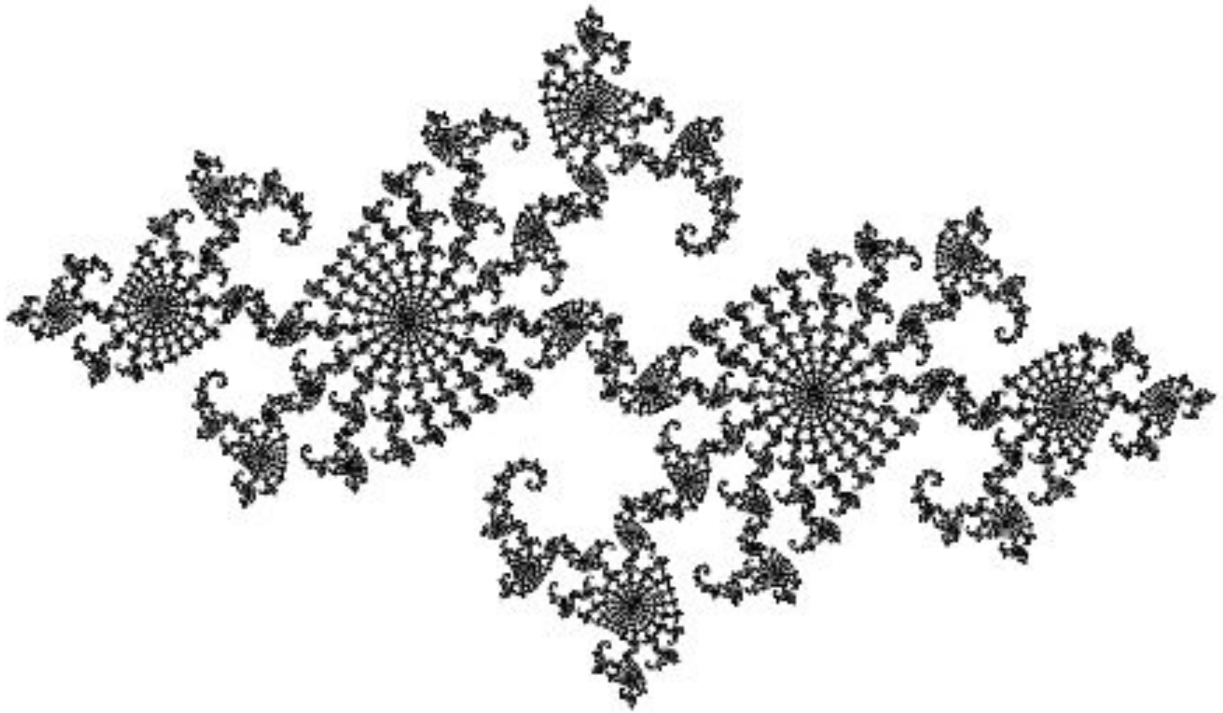
Fragment Shader



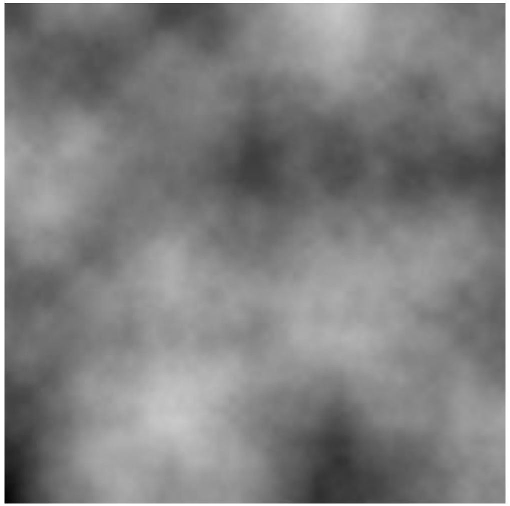
¿Cómo funcionan estas herramientas?

```
var positionBuffer = [
  0, 0, 0, 0,
  0, 0.5, 0, 0,
  0.7, 0, 0, 0,
];
var attributes = {};
var gl_Position;

drawArrays(..., offset, count) {
  var stride = 4;
  var size = 4;
  for (var i = 0; i < count; ++i) {
     // copy the next 4 values from positionBuffer to the a_position attribute
     const start = (offset + i) * stride;
     attributes.a_position = positionBuffer.slice(start, start + size);
     runVertexShader();
     ...
     doSomethingWith_gl_Position();
}
```

Fractal de julia



Perlin Noise



Domain Warping