Shaders en Processing Parte II

Materiales online

Tutoriales:

- http://www.lighthouse3d.com/tutorials/
- http://ogldev.atspace.co.uk/
- http://www.opengl-tutorial.org/
- https://open.gl
- http://learnopengl.com/
- http://www.songho.ca/opengl/index.html
- http://thebookofshaders.com/

Técnicas avanzadas

- http://prideout.net/blog/
- http://paulbourke.net/
- http://http.developer.nvidia.com/GPUGems3

Entornos de programación de shaders en la web

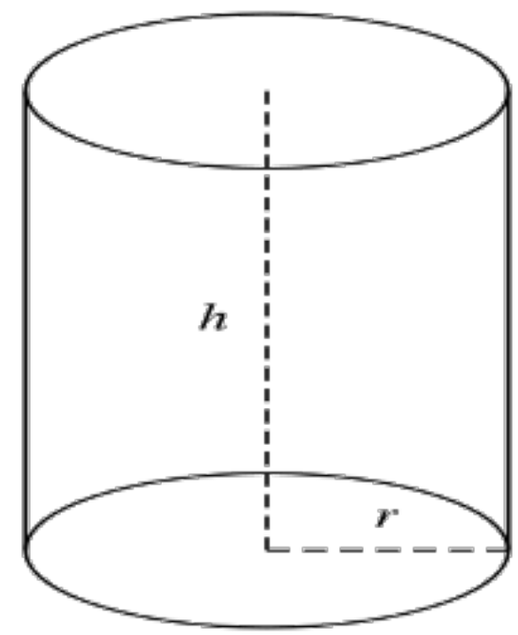
- https://www.shadertoy.com/
- http://glsl.heroku.com/
- https://www.vertexshaderart.com/

Distintos tipos de shaders

- Shaders de puntos (stroke points)
- Shaders de lineas (stroke lines)
- Shaders de color (geometría sin luces ni texturas)
- Shaders de texturado (geometría texturada sin luces)
- Shaders de iluminación (geometría iluminada sin texturas)
- Shaders de iluminación y texturado (geometría con luces y textura)

Un PShape cilíndrico

```
PShape createCan(float r, float h, int detail) {
  textureMode(NORMAL);
  PShape sh = createShape();
  sh.beginShape(QUAD_STRIP);
  sh.noStroke();
  for (int i = 0; i <= detail; i++) {
    float angle = TWO_PI / detail;
    float x = sin(i * angle);
    float z = cos(i * angle);
    float u = float(i) / detail;
    sh.normal(x, 0, z);
    sh.vertex(x * r, -h/2, z * r, u, 0);
    sh.vertex(x * r, +h/2, z * r, u, 1);
  }
  sh.endShape();
  return sh;
}
```



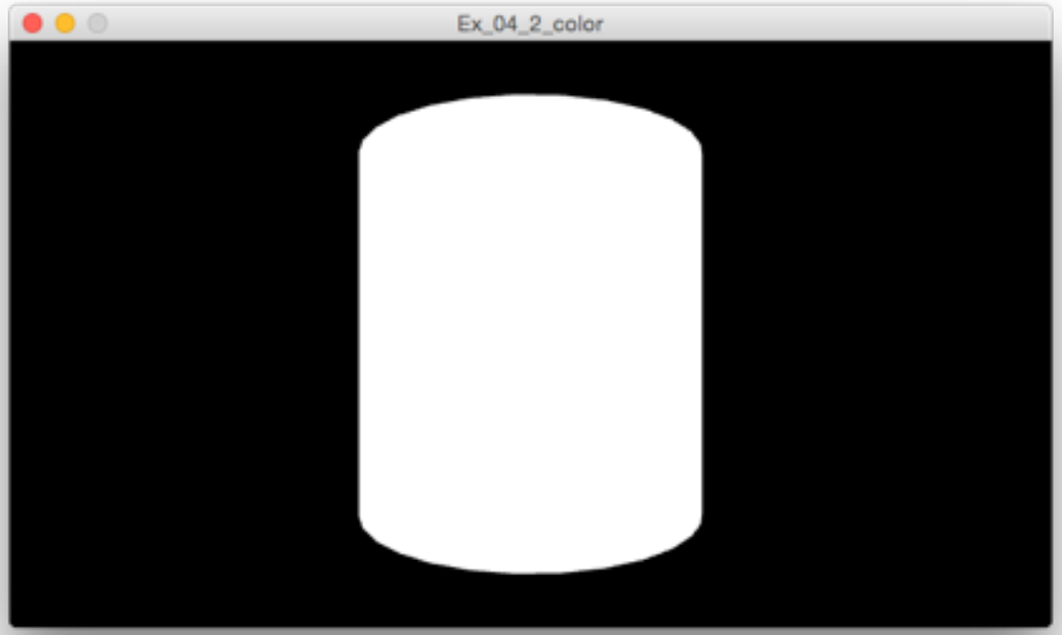
Shaders de color

colorvert.glsl:

```
#define PROCESSING_COLOR_SHADER

uniform mat4 transform;

attribute vec4 vertex;
attribute vec4 color;

varying vec4 vertColor;

void main() {
  gl_Position = transform * vertex;
  vertColor = color;
}
```

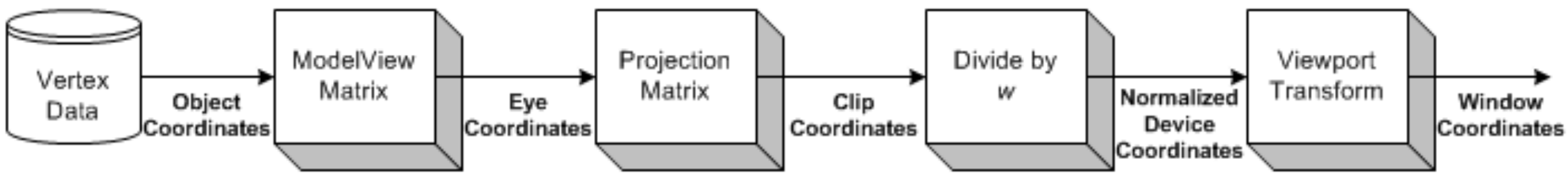
colorfrag.glsl:

```
#ifdef GL_ES
precision mediump float;
precision mediump int;
#endif

varying vec4 vertColor;

void main() {
  gl_FragColor = vertColor;
}
```



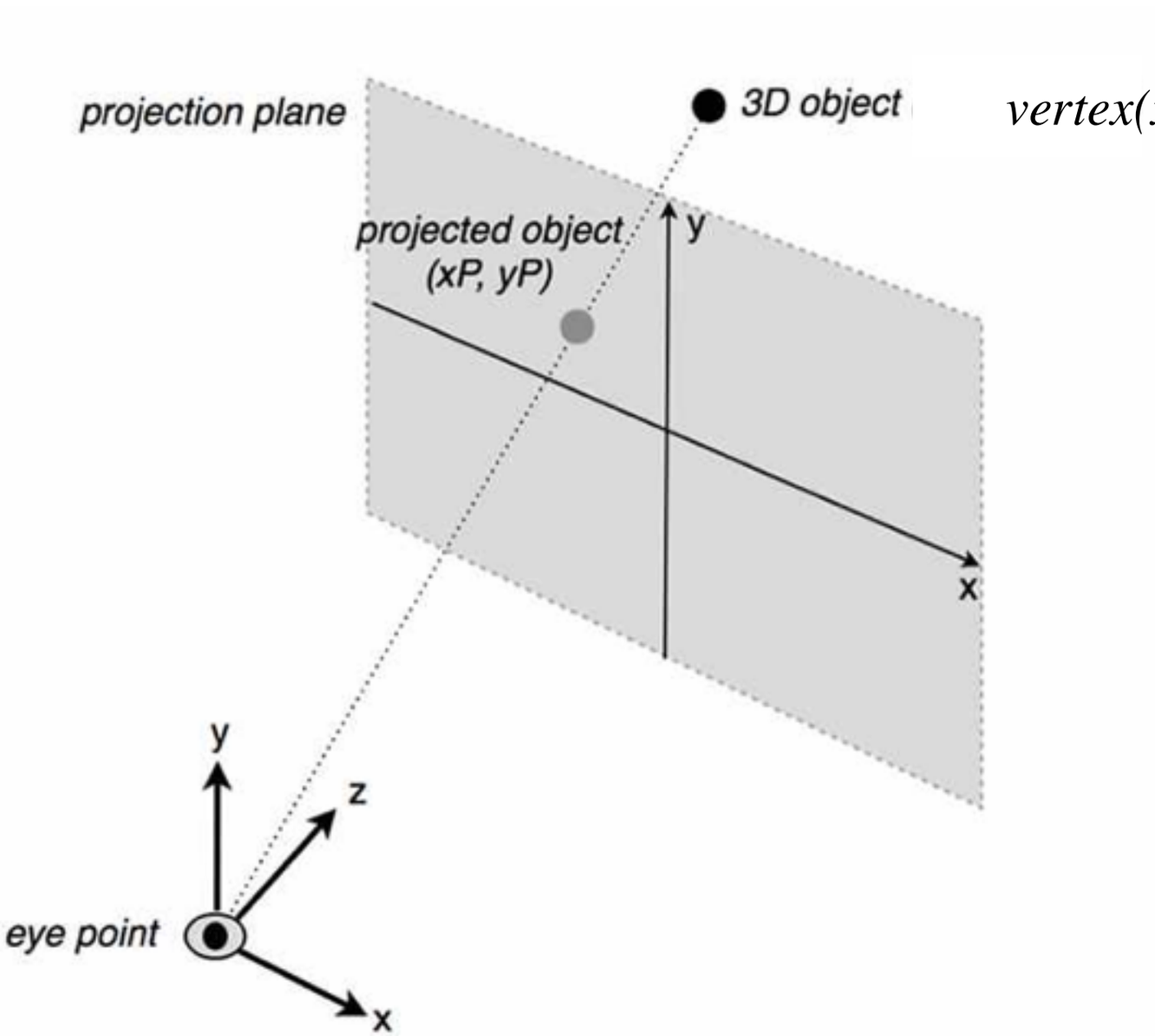
Algo de matemáticas

```
gl_Position = transform * position
```

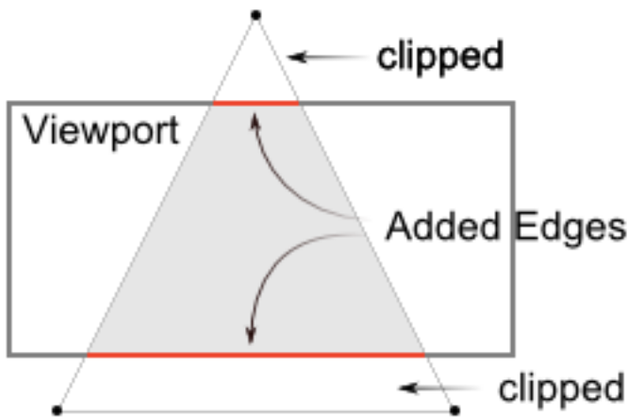



```
modelview   projection
 ______________________

    transform  = projection * modelview
                              ___________
                              view * model
```

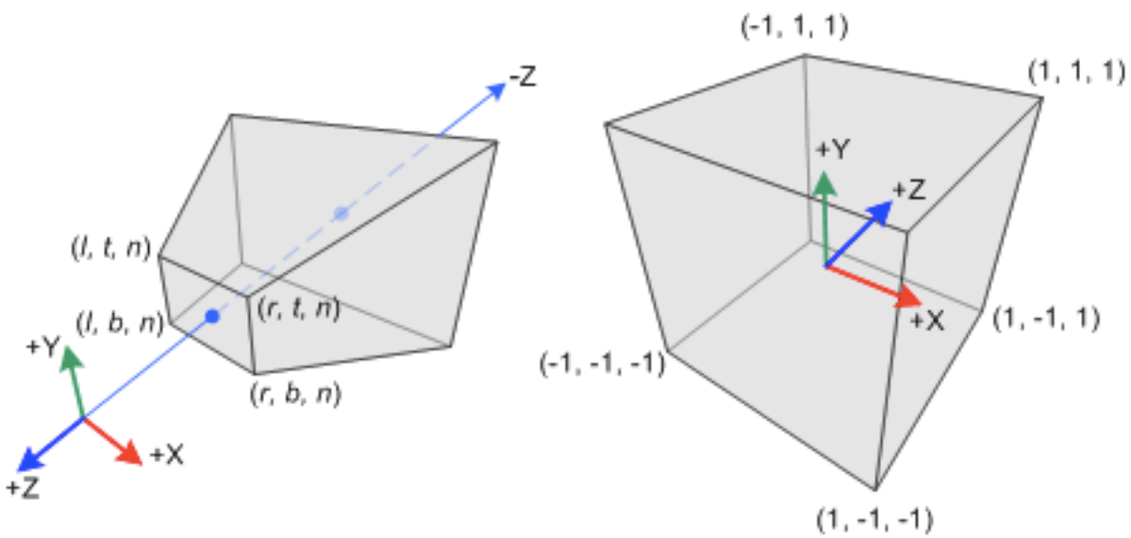
$$\begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix} = M_{modelView} \cdot \begin{pmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{pmatrix} = M_{view} \cdot M_{model} \cdot \begin{pmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ w_{obj} \end{pmatrix}$$



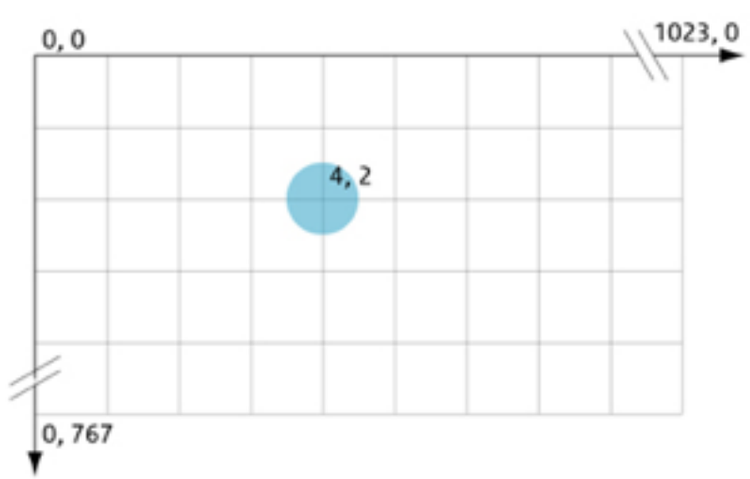
$vertex(x_{obj}, y_{obj}, z_{obj}, w_{obj}) \longrightarrow (x_{eye}, y_{eye}, z_{eye}, w_{eye})$



$$\begin{pmatrix} x_{clip} \\ y_{clip} \\ z_{clip} \\ w_{clip} \end{pmatrix} = M_{projection} \cdot \begin{pmatrix} x_{eye} \\ y_{eye} \\ z_{eye} \\ w_{eye} \end{pmatrix}$$



$$\begin{pmatrix} x_{ndc} \\ y_{ndc} \\ z_{ndc} \end{pmatrix} = \begin{pmatrix} x_{clip}/w_{clip} \\ y_{clip}/w_{clip} \\ z_{clip}/w_{clip} \end{pmatrix}$$



$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \begin{pmatrix} \frac{\mathrm{w}}{2} x_{ndc} + (\mathrm{x} + \frac{\mathrm{w}}{2}) \\ \frac{\mathrm{h}}{2} y_{ndc} + (\mathrm{y} + \frac{\mathrm{h}}{2}) \\ \frac{\mathrm{f}-\mathrm{n}}{2} z_{ndc} + \frac{\mathrm{f}+\mathrm{n}}{2} \end{pmatrix}$$

Shaders de textura

```
#define PROCESSING_TEXTURE_SHADER

uniform mat4 transform;
uniform mat4 texMatrix;

attribute vec4 vertex;
attribute vec4 color;
attribute vec2 texCoord;

varying vec4 vertColor;
varying vec4 vertTexCoord;

void main() {
  gl_Position = transform * vertex;

  vertColor = color;
  vertTexCoord = texMatrix * vec4(texCoord, 1.0, 1.0);
}
```

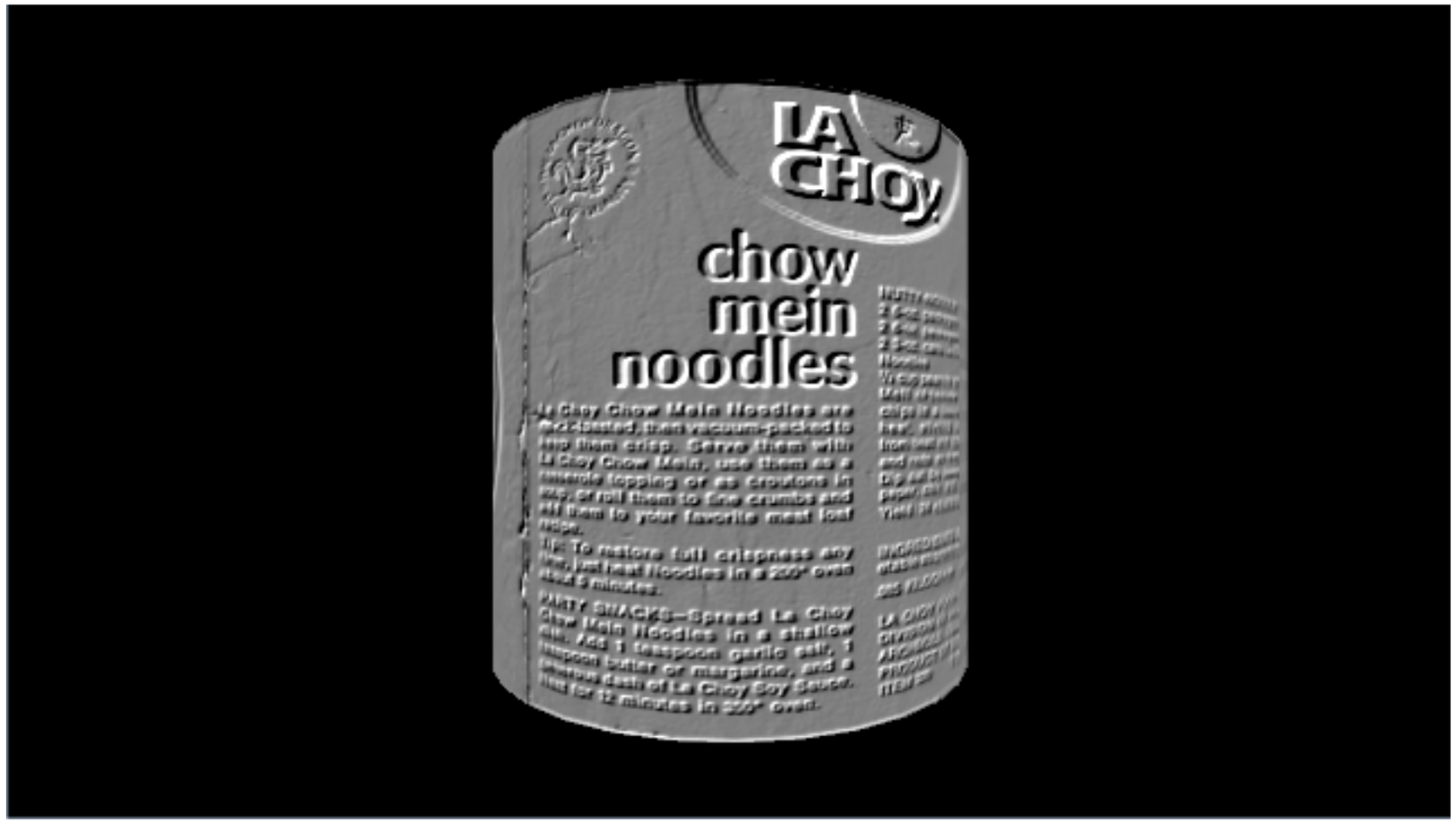
vertex(x, y, z, u, v)

texfrag.glsl:

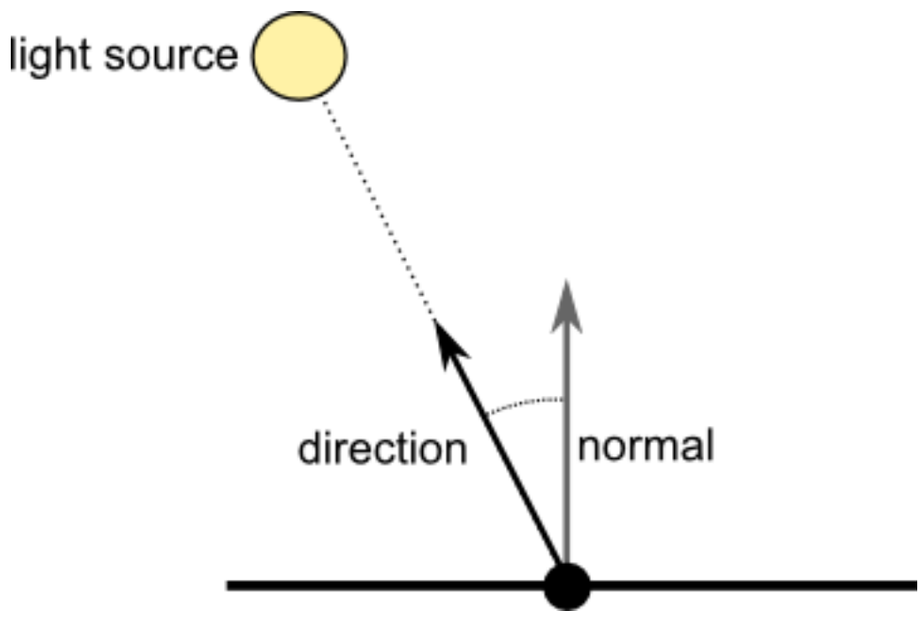
```
#ifdef GL_ES
precision mediump float;
precision mediump int;
#endif

uniform sampler2D texture;

varying vec4 vertColor;
varying vec4 vertTexCoord;

void main() {
  gl_FragColor = texture2D(texture, vertTexCoord.st) * vertColor;
}
```

image(foto, ...)

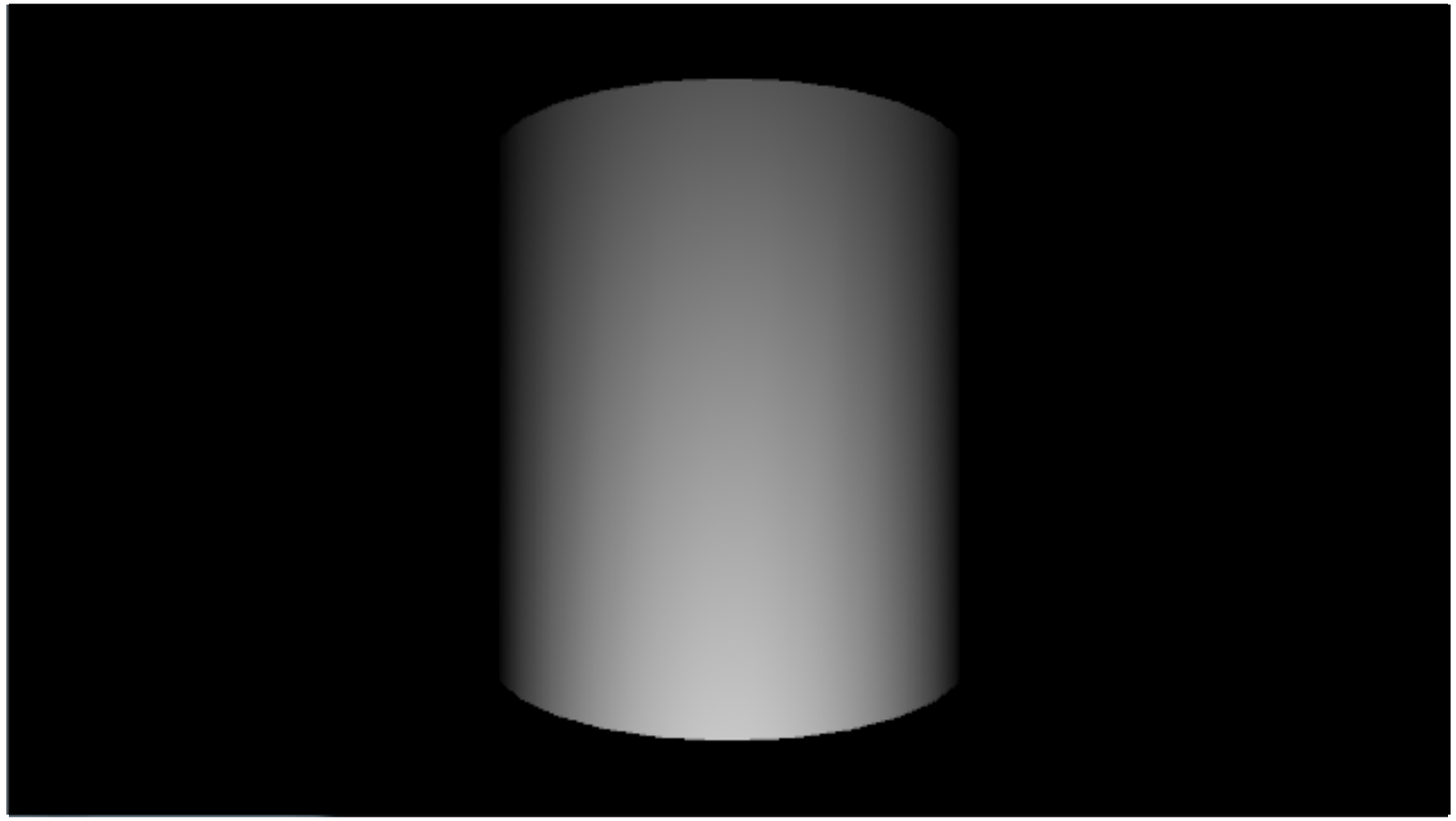
Procesamiento de imágenes en el shader de fragmentos



Shaders de iluminación

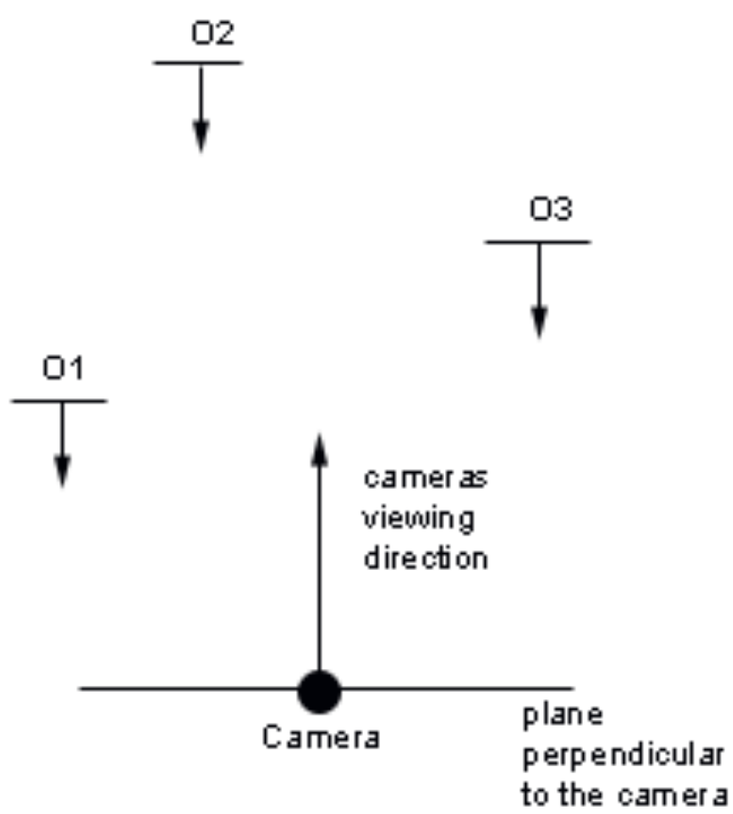


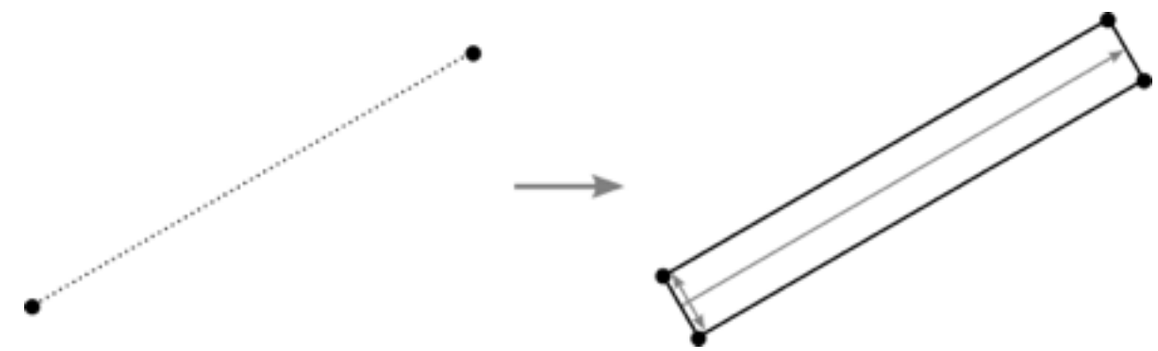
Iluminación por-vértice vs por-pixel

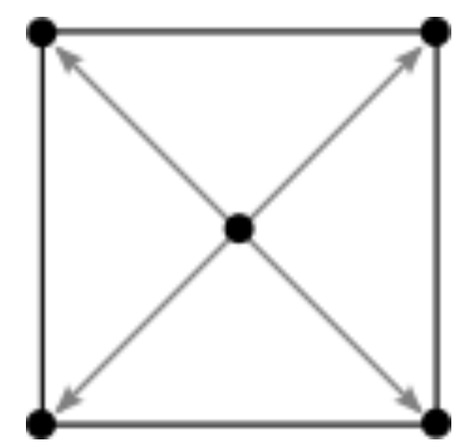


Combinando iluminación y texturado

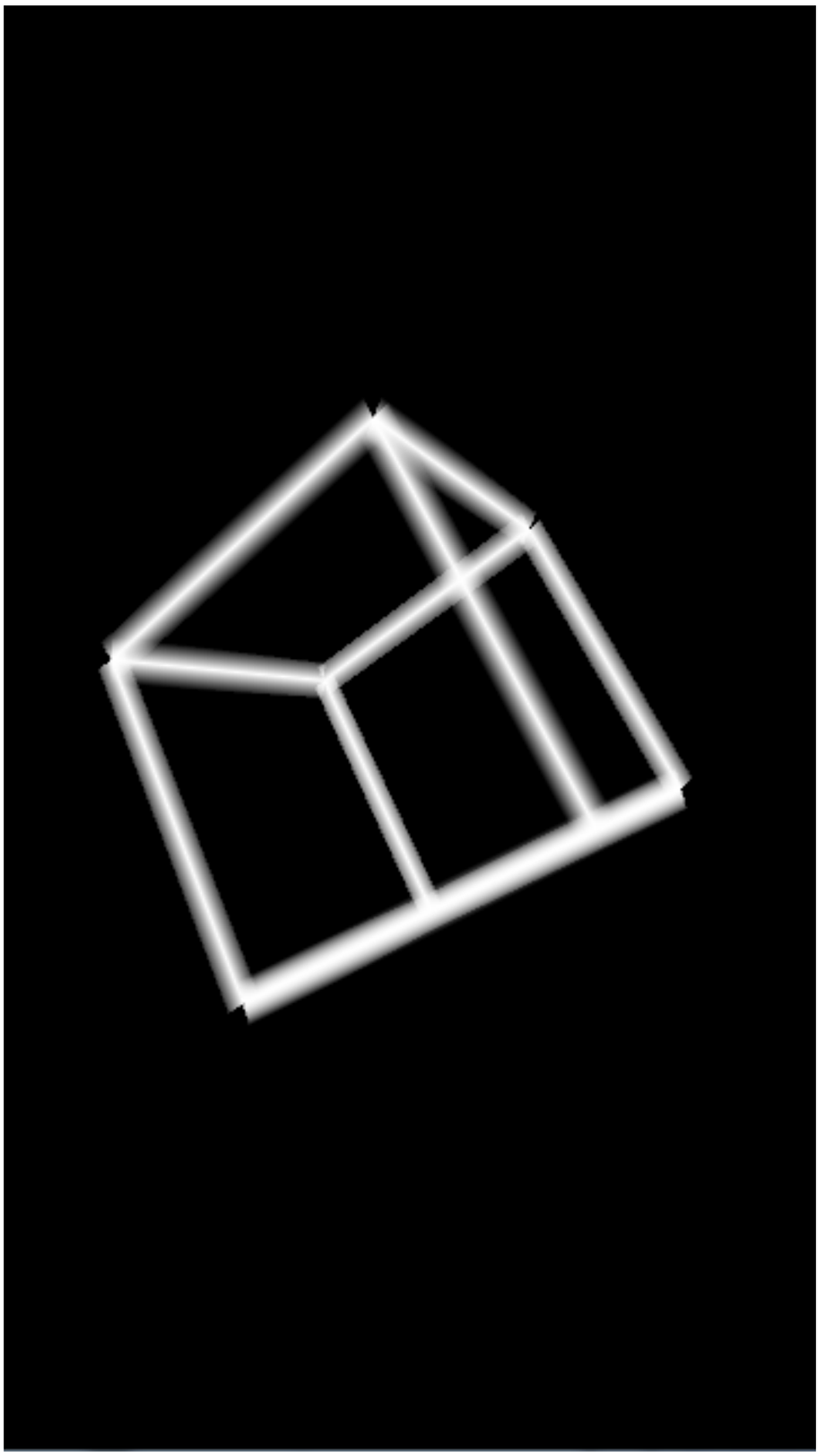
Shaders de puntos y líneas





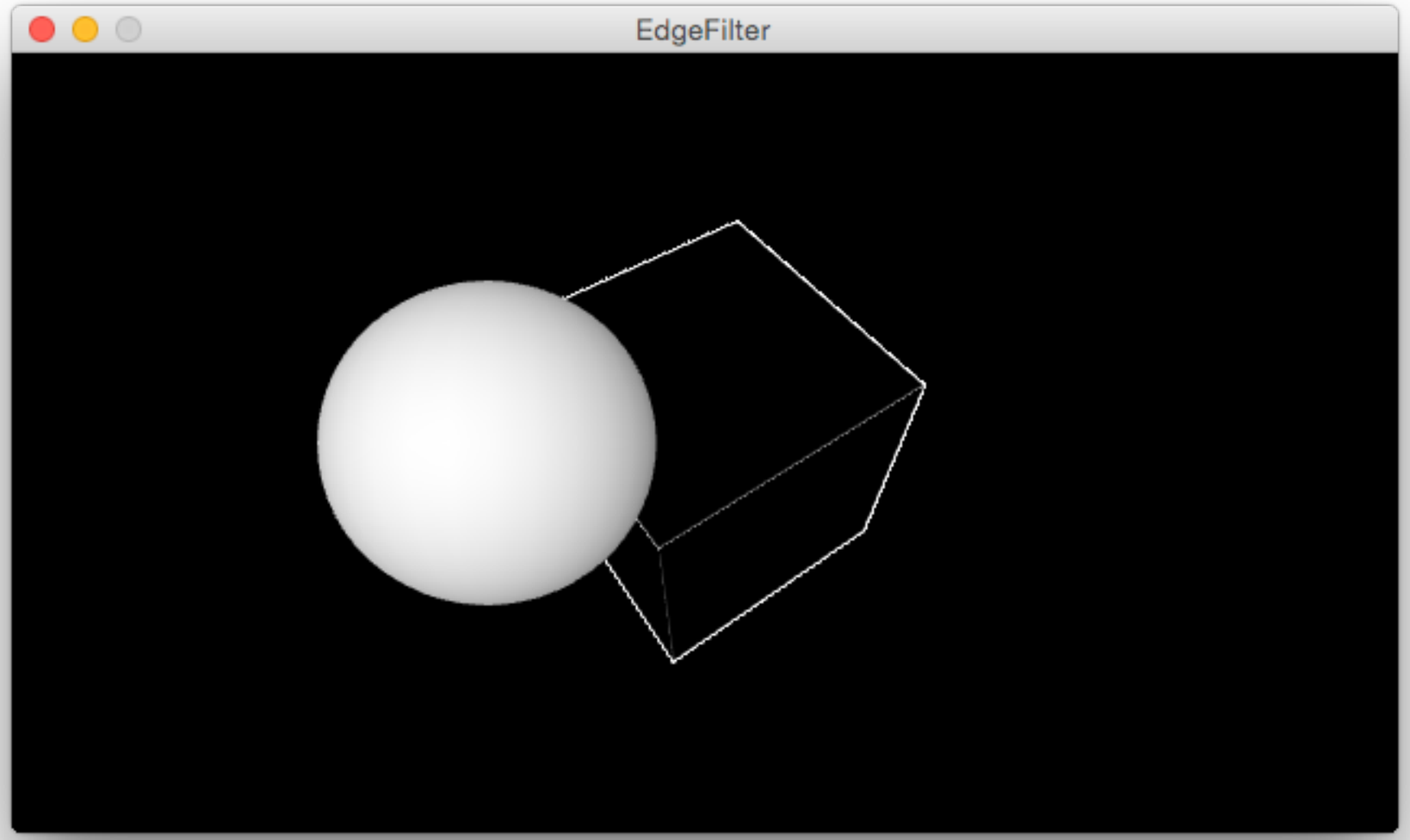


Líneas y puntos customizados



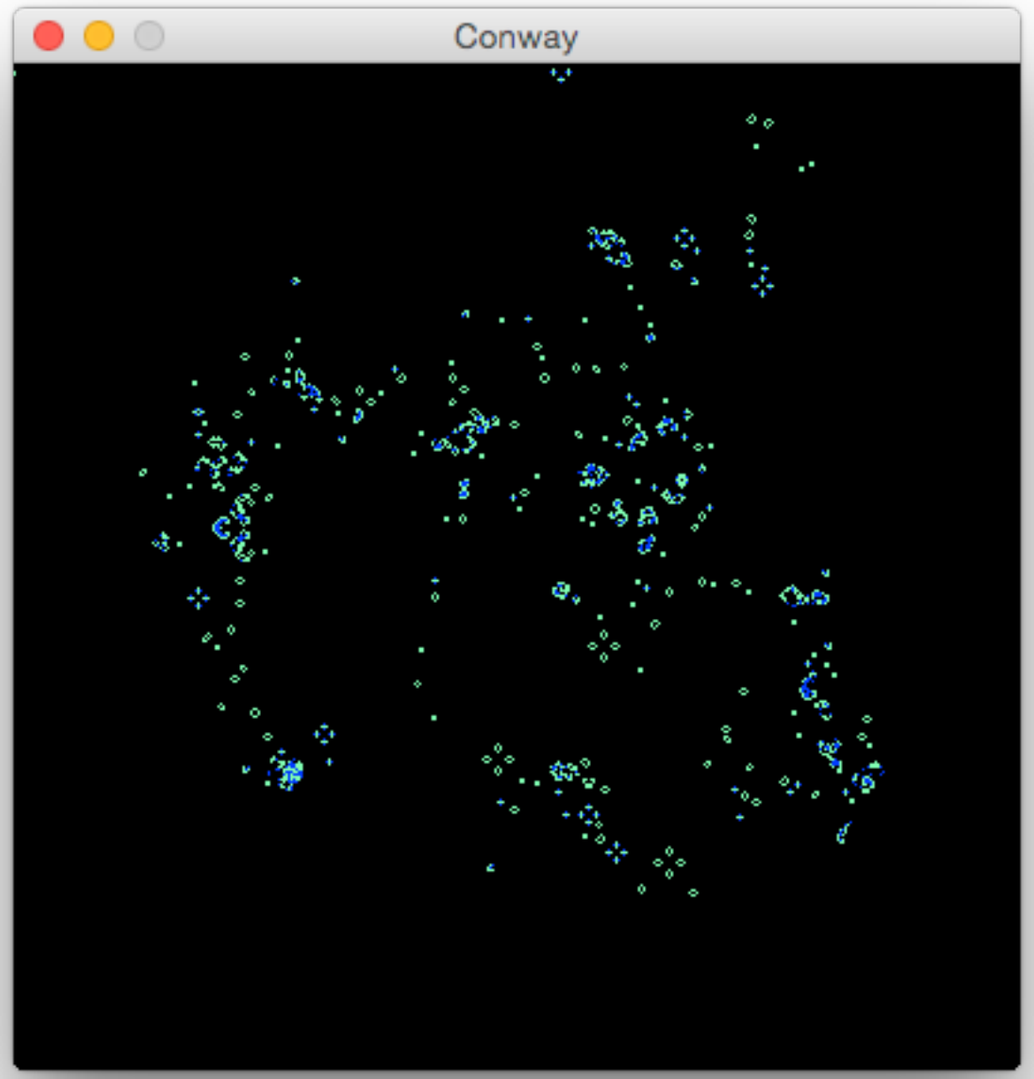
Algunos “trucos” con shaders en Processing

Aplicar un filtro sobre toda la pantalla



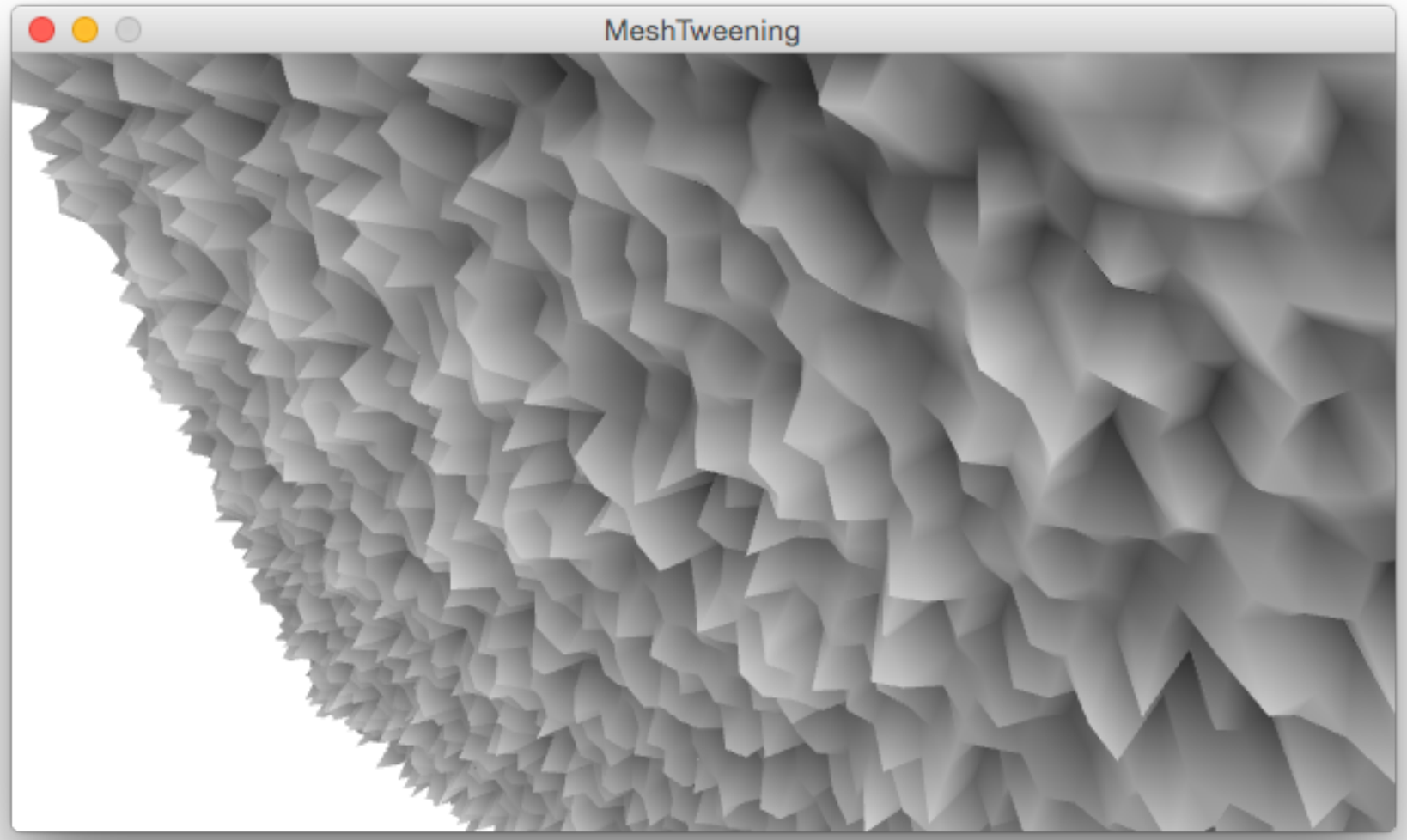
Ejemplo: Topics|Shaders|EdgeFilter

Leer los píxeles del cuadro anterior



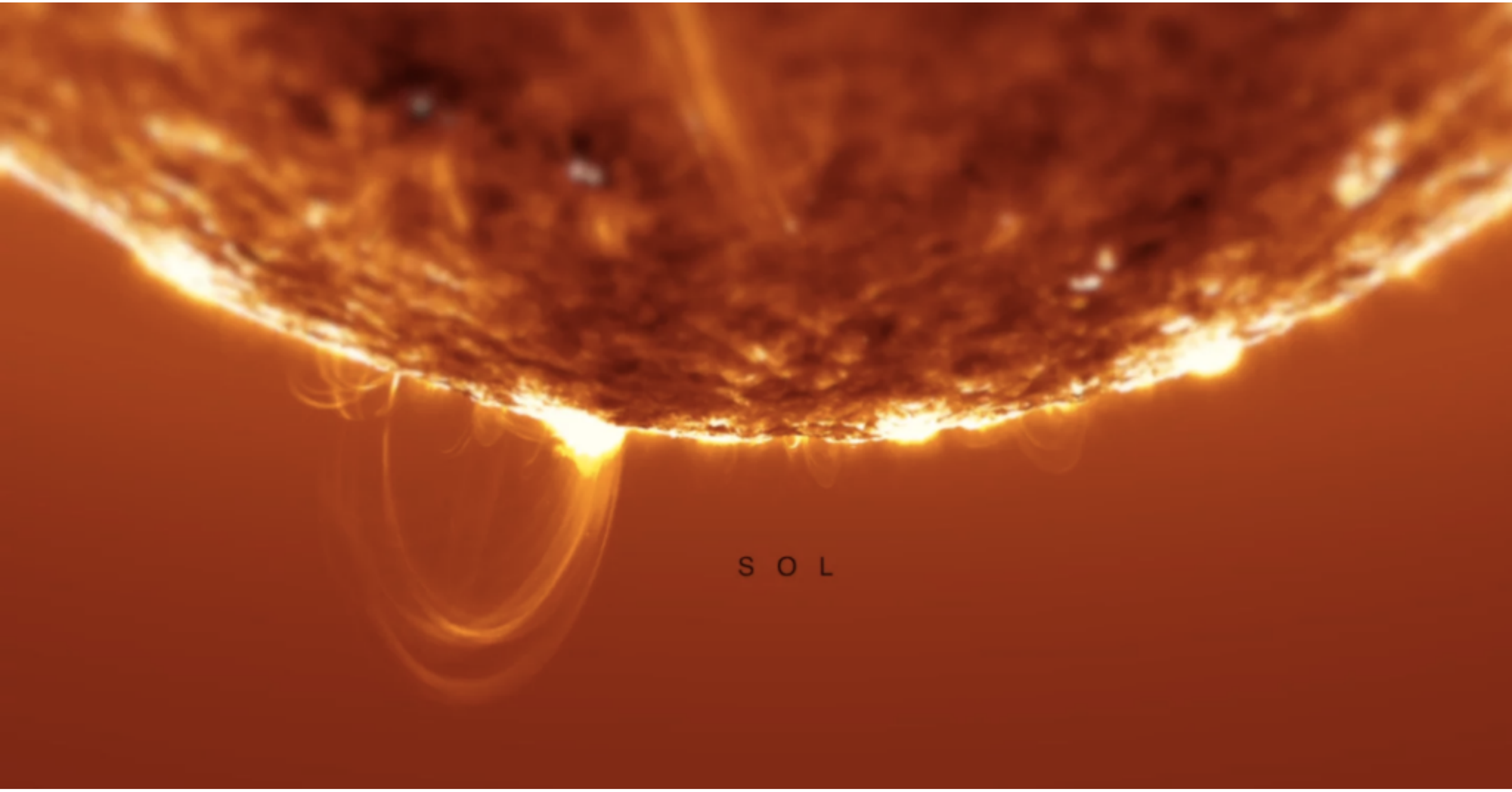
Ejemplo: Topics|Shaders|Conway

Crear nuevos atributos de vértice



Ejemplo: Demos|Graphics|MeshTweening

flight404 (Robert Hodgins)



https://vimeo.com/146875858