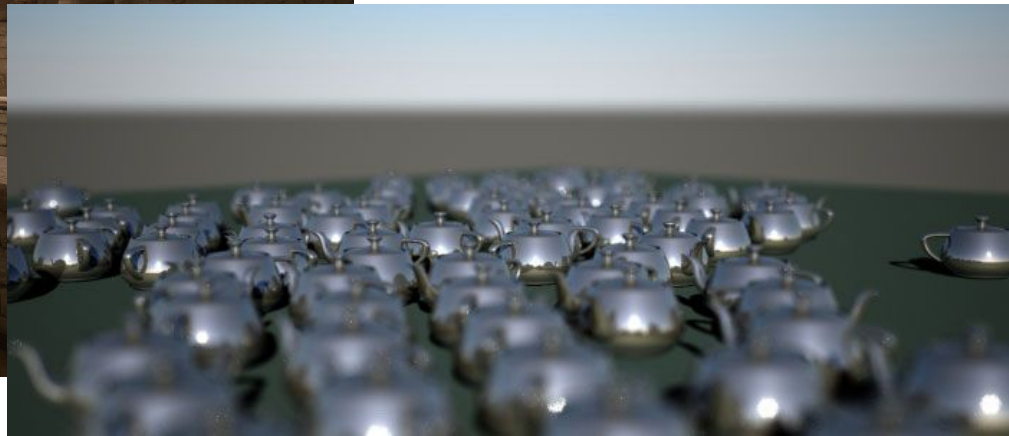
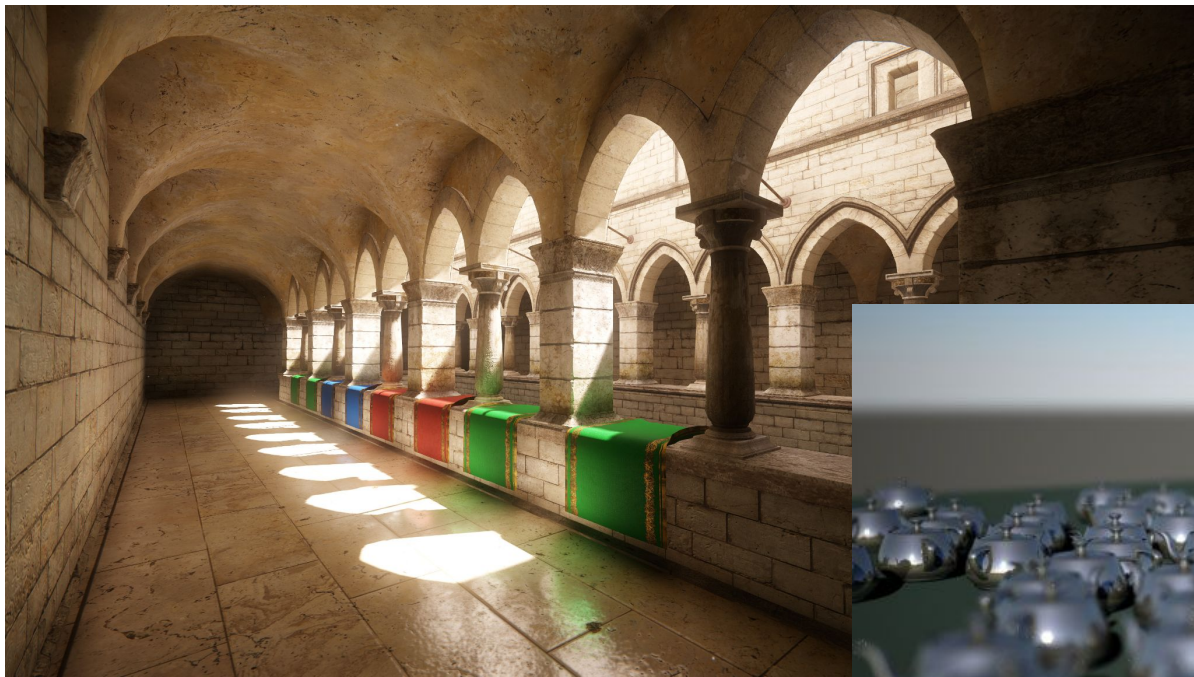
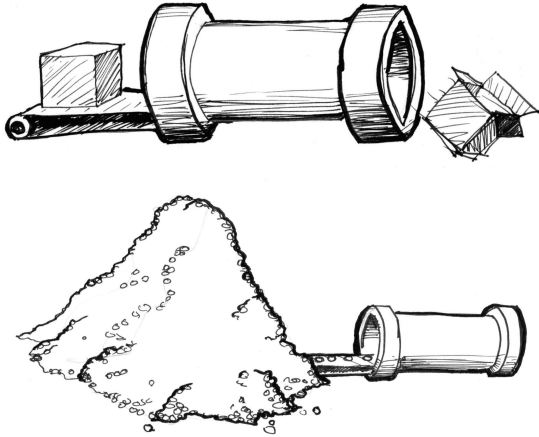


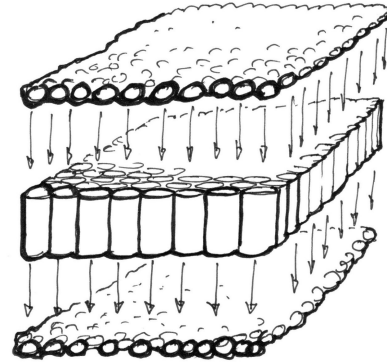
Shaders



Qué son los shaders?

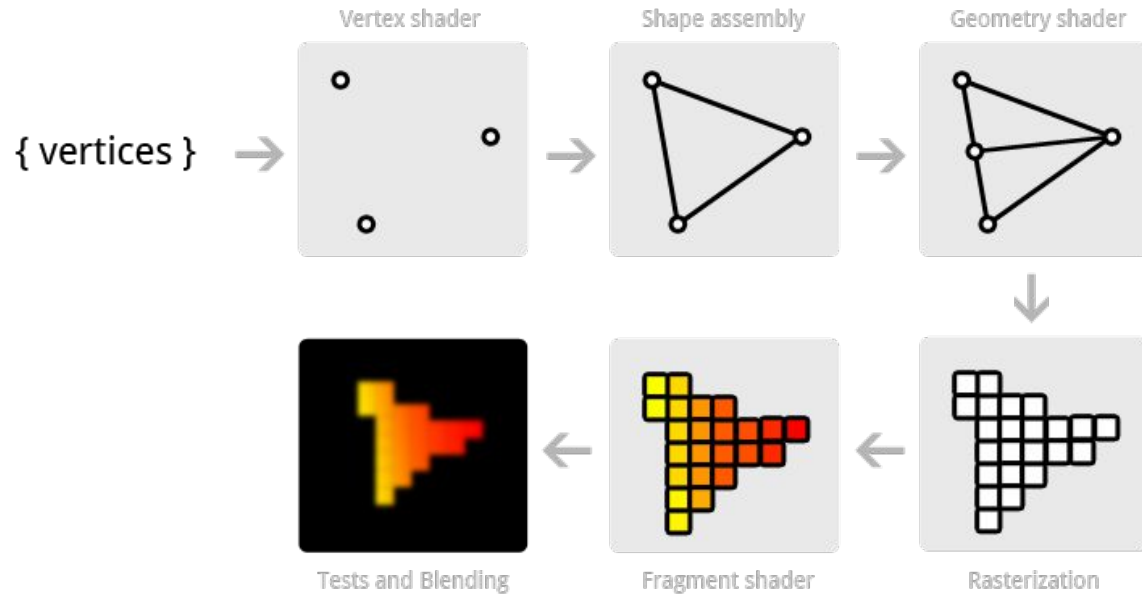


CPU



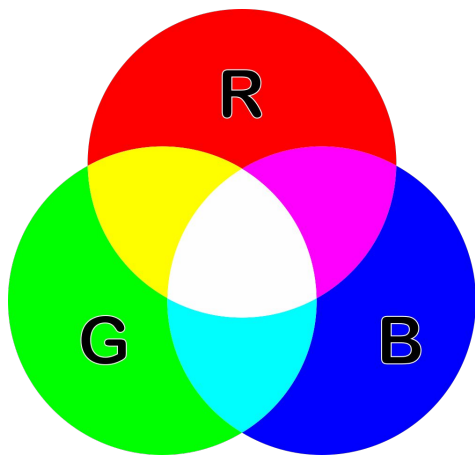
Múltiples threads en paralelo

Tipos de Shaders

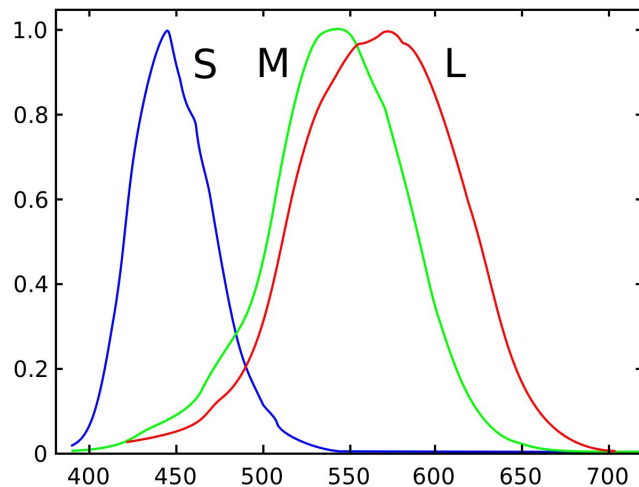


Pipeline de procesamiento gráfico

Espacios de Colores



RGB color space



LMS color space

Programando shaders

Características del lenguaje:

- Lenguaje similar a C
- Tienen una función principal que recibe como entrada una posición de pixeles fragCoord y devuelve un color en fragColor
- Utiliza espacio de colores RGBA

Como representar colores

```
vec4 vector;  
vector[0] = vector.r = vector.x = vector.s;  
vector[1] = vector.g = vector.y = vector.t;  
vector[2] = vector.b = vector.z = vector.p;  
vector[3] = vector.a = vector.w = vector.q;
```

Funciones Aceleradas por hardware

length()

distance()

dot()

cross()

normalize()

faceforward()

reflect()

refract()

sin()

cos()

tan()

asin()

acos()

atan()

pow()

exp()

log()

sqrt()

abs()

sign()

floor()

ceil()

fract()

mod()

min()

max()

clamp()

Debugging

Debugging

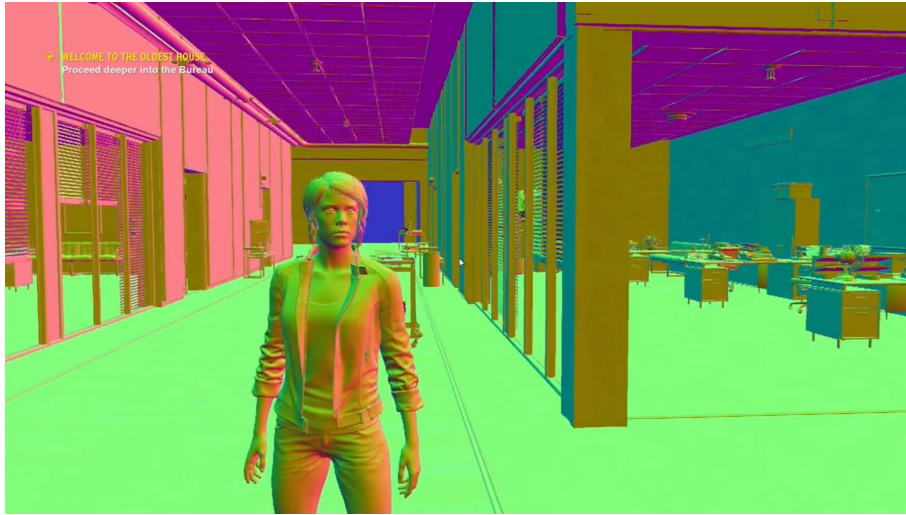
Debugging shaders:



Debugging code:



Debugging



Normal Map



Ambient Occlusion Map

Que se pueden hacer con Shaders?

- Crear formas procedurales, patrones, texturas y animaciones.
- Procesamiento de imágenes
- Simulaciones
- Y más.

Como correr shaders

- Shadertoy
- glslViewer
- glslCanvas
- Processing
- WebGL
- openFrameworks

Para continuar leyendo

- The book of shaders
- “Passes”
- Mezclado de colores



The Book of Shaders

by Patricio Gonzalez Vivo

Muchas gracias