LÖVE®

An *awesome* framework you can use to make 2D games in Lua.

Jordi Borraz
Nil Muns
Ricard Ruiz
Héctor Vergel

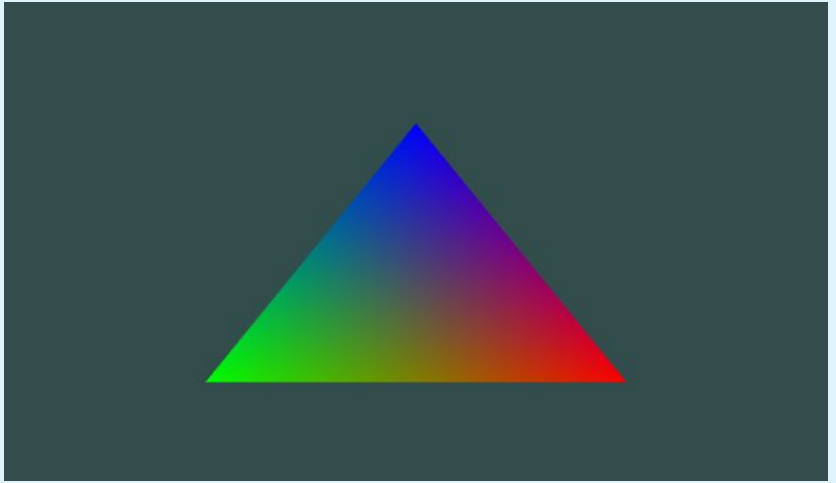
Introducción

- Postprocesamiento
- Librería Externa
- Moonshine

Implementación

- Shaders en LÖVE

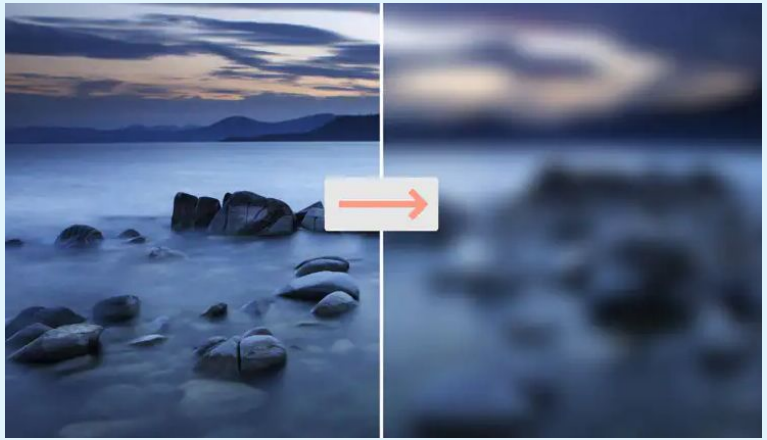
- GLSL



Ejemplos

- Shaders en LÖVE

- BoxBlur



List of effects

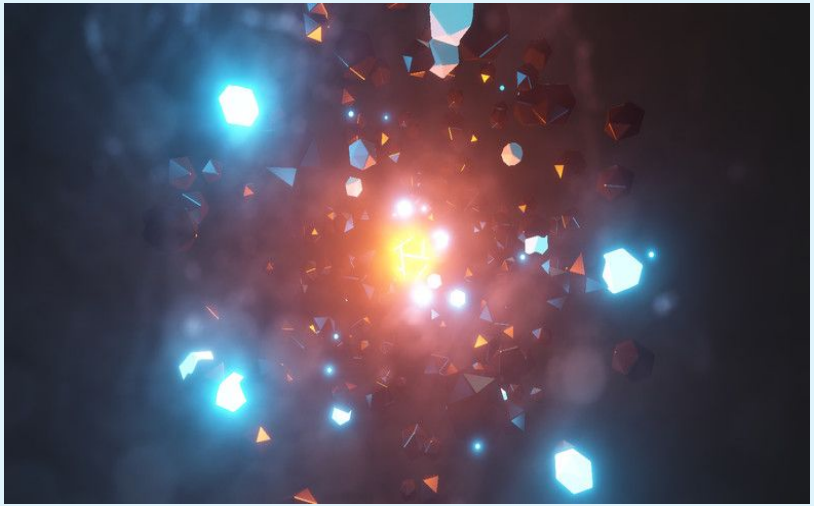
Currently, moonshine contains the following effects (in alphabetical order):

- boxblur: simple blurring
- chromasep: cheap/fake chromatic aberration
- colorgradesimple: weighting of color channels
- crt: crt/barrel distortion
- desaturate: desaturation and tinting
- dmg: Gameboy and other four color palettes
- fastgaussianblur: faster Gaussian blurring
- filmgrain: image noise
- gaussianblur: Gaussian blurring
- glow: aka (light bloom
- godsray: aka light scattering
- pixelate: sub-sampling (for that indie look)
- posterize: restrict number of colors
- scanlines: horizontal lines
- sketch: simulate pencil drawings
- vignette: shadow in the corners

Ejemplos

- DMG
  - Efecto GameBoy

- Glow
  - Luminosidad



Ejemplos

- DMG
  - Efecto GameBoy

```lua
local lookup_palette = function(name)
  for _,palette in pairs(palettes) do
    if palette.name == name then
      return palette
    end
  end
end

local is_valid_palette = function(v)
  -- Needs to match: {{R,G,B},{R,G,B},{R,G,B},{R,G,B}}
  if #v ~= 4 then return false end

  for i = 1,4 do
    if type(v[i]) ~= "table" or #v[i] ~= 3 then return false end
    for c = 1,3 do
      if type(v[i][c]) ~= "number" then return false end
      local x = v[i][c]
      if x > 1 then x = x / 255 end
      if x < 0 or x > 1 then return false end
      v[i][c] = x
    end
  end
  return true
end
```

```lua
name = "green",
colors = {
  {8/255,56/255,8/255},
  {48/255,96/255,48/255},
  {136/255,168/255,8/255},
  {183/255,220/255,17/255}
}
```

```lua
name = "greyscale",
colors = {
  {56/255,56/255,56/255},
  {117/255,117/255,117/255},
  {178/255,178/255,178/255},
  {239/255,239/255,239/255}
}
```

```lua
name = "dark_yellow",
colors = {
  {33/255,32/255,16/255},
  {107/255,105/255,49/255},
  {181/255,174/255,74/255},
  {255/255,247/255,123/255}
}
```

```lua
name = "stark_bw",
colors = {
  {0/255,0/255,0/255},
  {117/255,117/255,117/255},
  {178/255,178/255,178/255},
  {255/255,255/255,255/255}
}
```

```lua
name = "light_yellow",
colors = {
  {102/255,102/255,37/255},
  {148/255,148/255,64/255},
  {208/255,208/255,102/255},
  {255/255,255/255,148/255}
}
```

```lua
name = "pocket",
colors = {
  {108/255,108/255,78/255},
  {142/255,139/255,87/255},
  {195/255,196/255,165/255},
  {227/255,230/255,201/255}
}
```

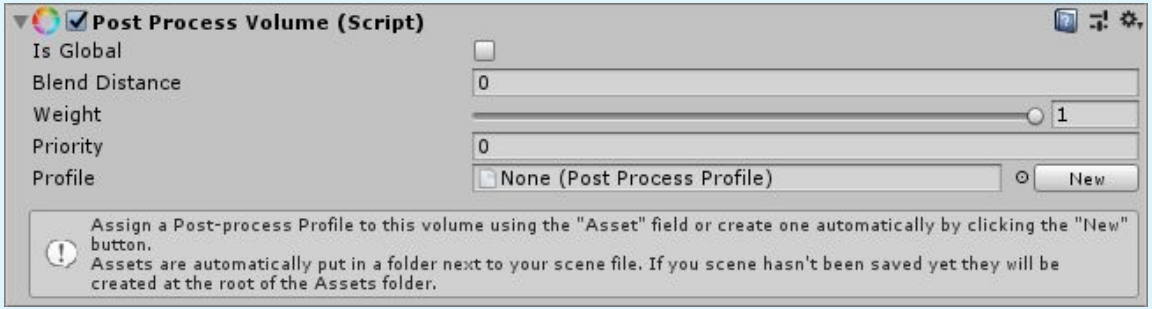
- Glow
  - Luminosidad

```lua
-- 1st pass: draw scene with brightness threshold
love.graphics.setCanvas(front)
love.graphics.clear()
love.graphics.setShader(threshold)
love.graphics.draw(scene)

-- 2nd pass: apply blur shader in x
blurshader:send('direction', {1 / love.graphics.getWidth(), 0})
love.graphics.setCanvas(back)
love.graphics.clear()
love.graphics.setShader(blurshader)
love.graphics.draw(front)

-- 3nd pass: apply blur shader in y and draw original and blurred scene
love.graphics.setCanvas(front)
love.graphics.clear()

-- original scene without blur shader
love.graphics.setShader()
love.graphics.setBlendMode("add", "premultiplied")
love.graphics.draw(scene) -- original scene

-- second pass of light blurring
blurshader:send('direction', {0, 1 / love.graphics.getHeight()})
love.graphics.setShader(blurshader)
love.graphics.draw(back)

-- restore things as they were before entering draw()
love.graphics.setBlendMode("alpha", "premultiplied")
scene = back
end
```

LÖVE vs Unity

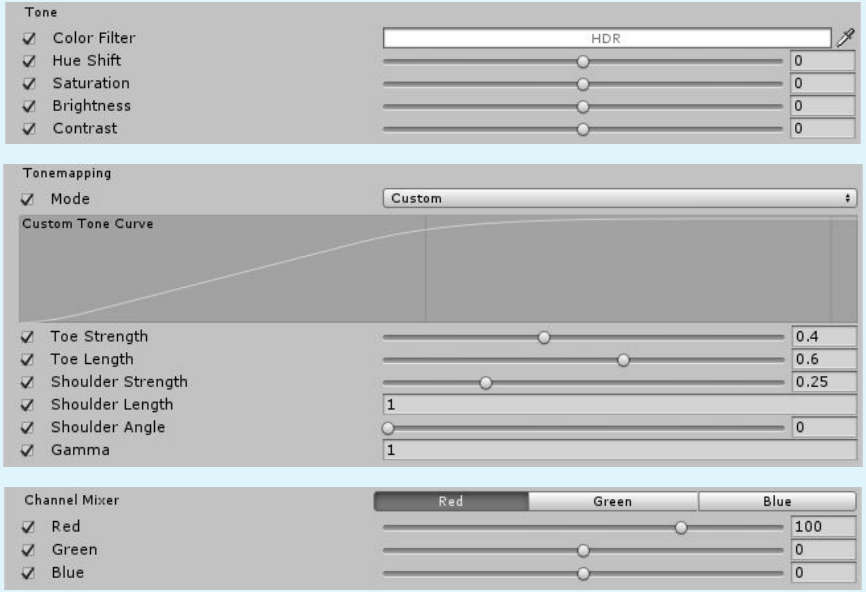
- Sistema de Componente “Volume”

- Más completo y versátil

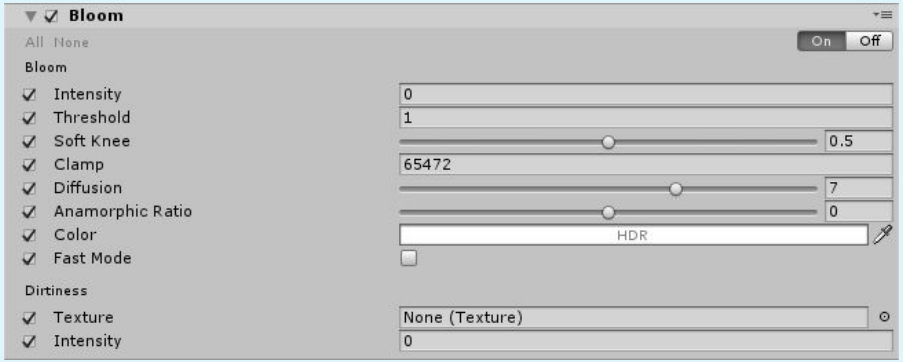


LÖVE vs Unity

- Color-Grading
  - Efecto GameBoy y más



- Bloom
  - Luminosidad



Conclusiones

- LÖVE
  - Todo por código
  - Menos opciones
  - Librerías Externas
  - Más complicado

- Unity
  - Facilidad por Editor y código
  - Más opciones
  - Integrado en el Motor
  - Más sencillo

Gracias por vuestra atención

¿Alguna pregunta o sugerencia?