
SHADERS

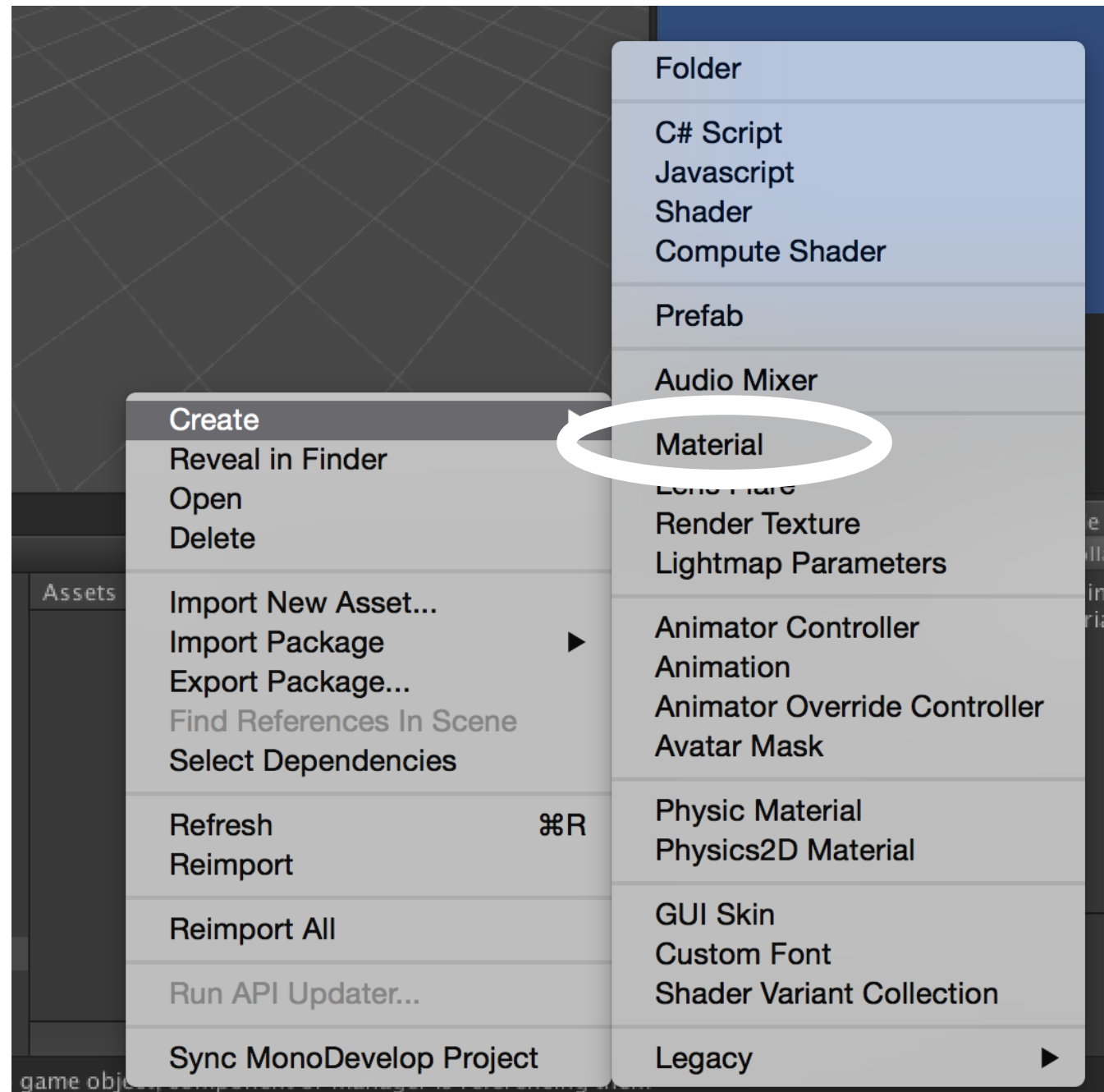
QUÉ SON LOS SHADERS?

- Son pequeños programas que corren en la placa de video (GPU).
- Hay distintos tipos y se utilizan para cosas específicas (vertex, fragment, geometry, tessellation).
- Tienen un set de instrucciones mucho más pequeño que los programas que corren en CPU...
- ... pero corren en paralelo en forma masiva.

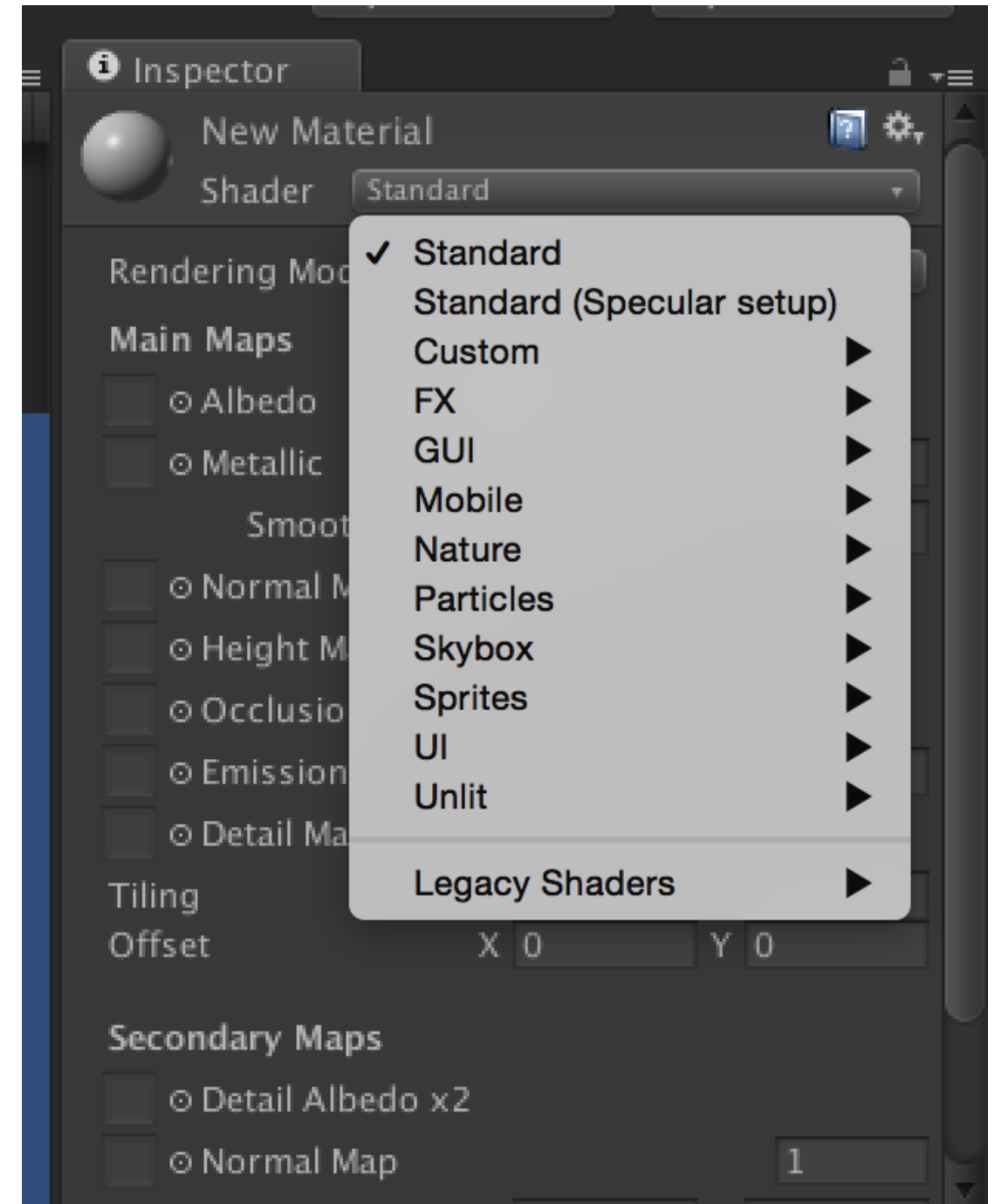
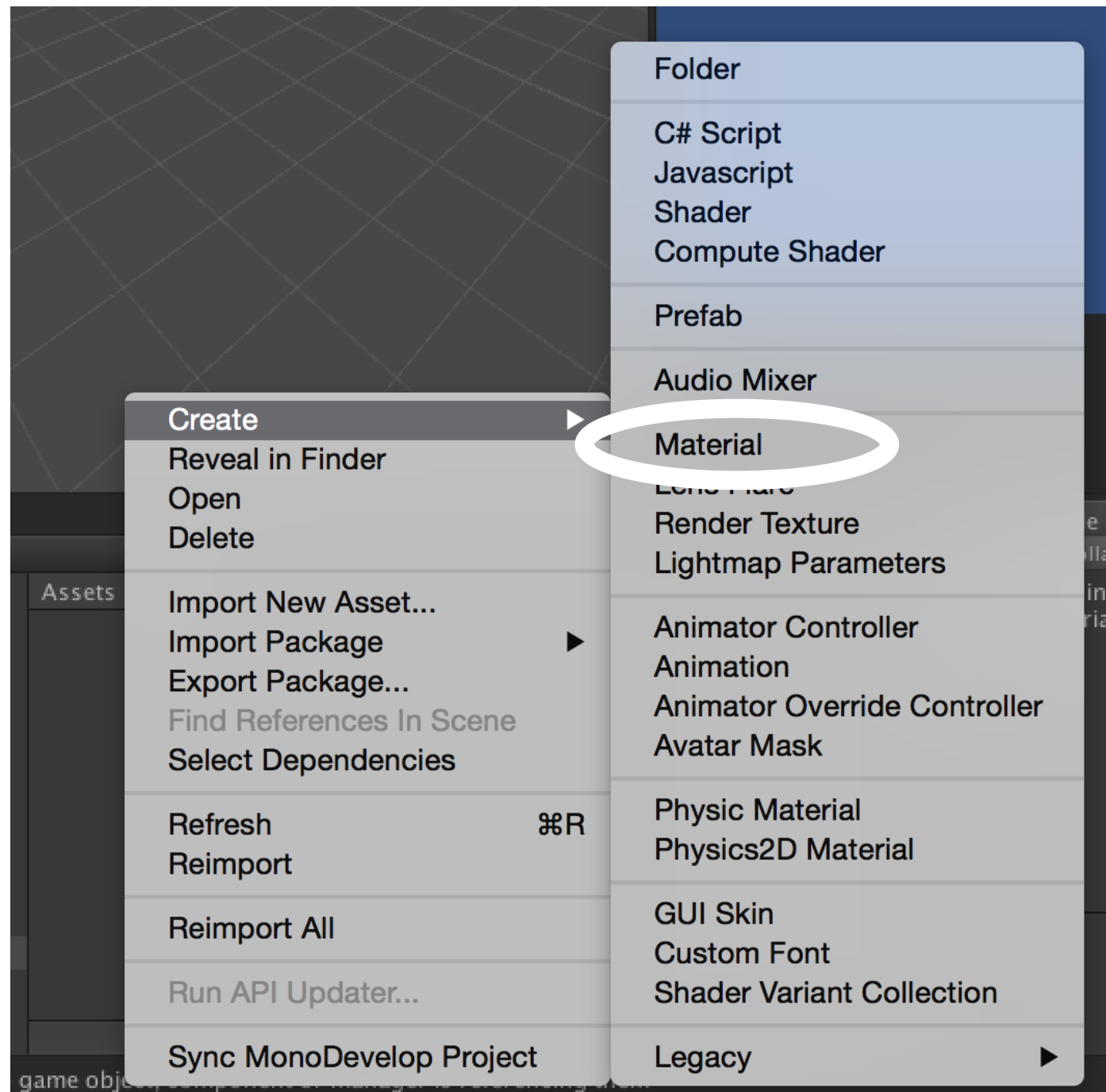
PARA QUÉ SE USAN?

- Son la herramienta que tenemos para lograr mostrar cosas en pantalla
- Texturizar e iluminar un modelo 3D
- Mostrar un **sprite** 2D
- Efectos especiales
- Deformaciones de los modelos
- Etc.

CÓMO SE USAN EN UNITY?



CÓMO SE USAN EN UNITY?



PARA EXPERIMENTAR: BUILT-IN SHADERS

<https://unity3d.com/get-unity/download/archive>

SHADERS

```
1 Shader "Folder/Shader" {
2     Properties
3     {
4         // ...
5     }
6
7     SubShader
8     {
9         Pass
10        {
11            CGPROGRAM
12            // ...
13            ENDCG
14        }
15    }
16 }
17
```

ANTES QUE NADA: CÓMO SE REPRESENTA UN COLOR?

R**G****B**A

(0,0,0,1) Negro

(1,1,1,1) Blanco

(.5,.5,.5,1) Gris medio

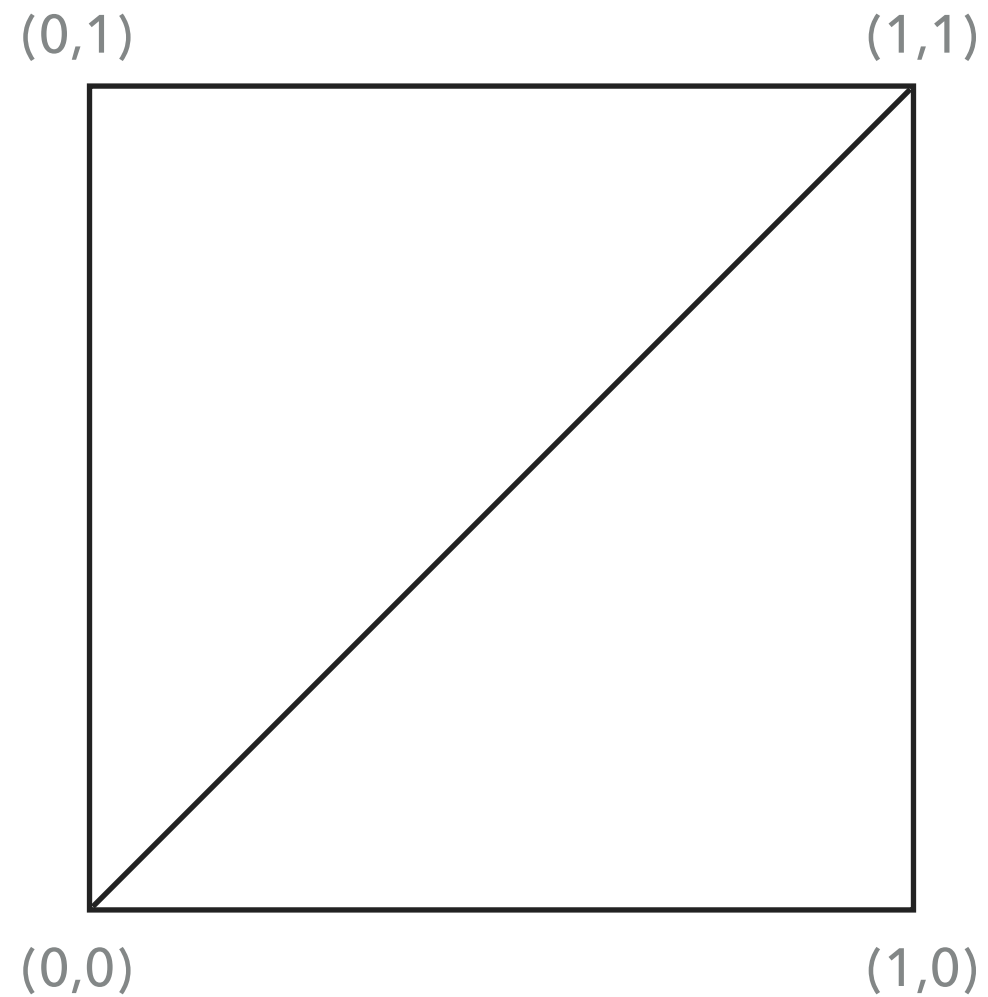
(1,0,1,1) Magenta (Errores)

(1,1,0,1) Amarillo

(0,1,1,1) Cyan

CREEMOS UN SHADER DESDE CERO

UV MAPPING (QUAD)



TIME

_Time	float4	Time (t/20, t, t*2, t*3), use to animate things inside the shaders.
_SinTime	float4	Sine of time: (t/8, t/4, t/2, t).
_CosTime	float4	Cosine of time: (t/8, t/4, t/2, t).
unity_DeltaTime	float4	Delta time: (dt, 1/dt, smoothDt, 1/smoothDt).

EJERCICIO

- 1) Abrir escena en "Ejercicios/1 - Endless Scroller/"
- 2) Editar shader "Endless Scroller" para que el fondo haga scroll indefinidamente

SOLUCIÓN

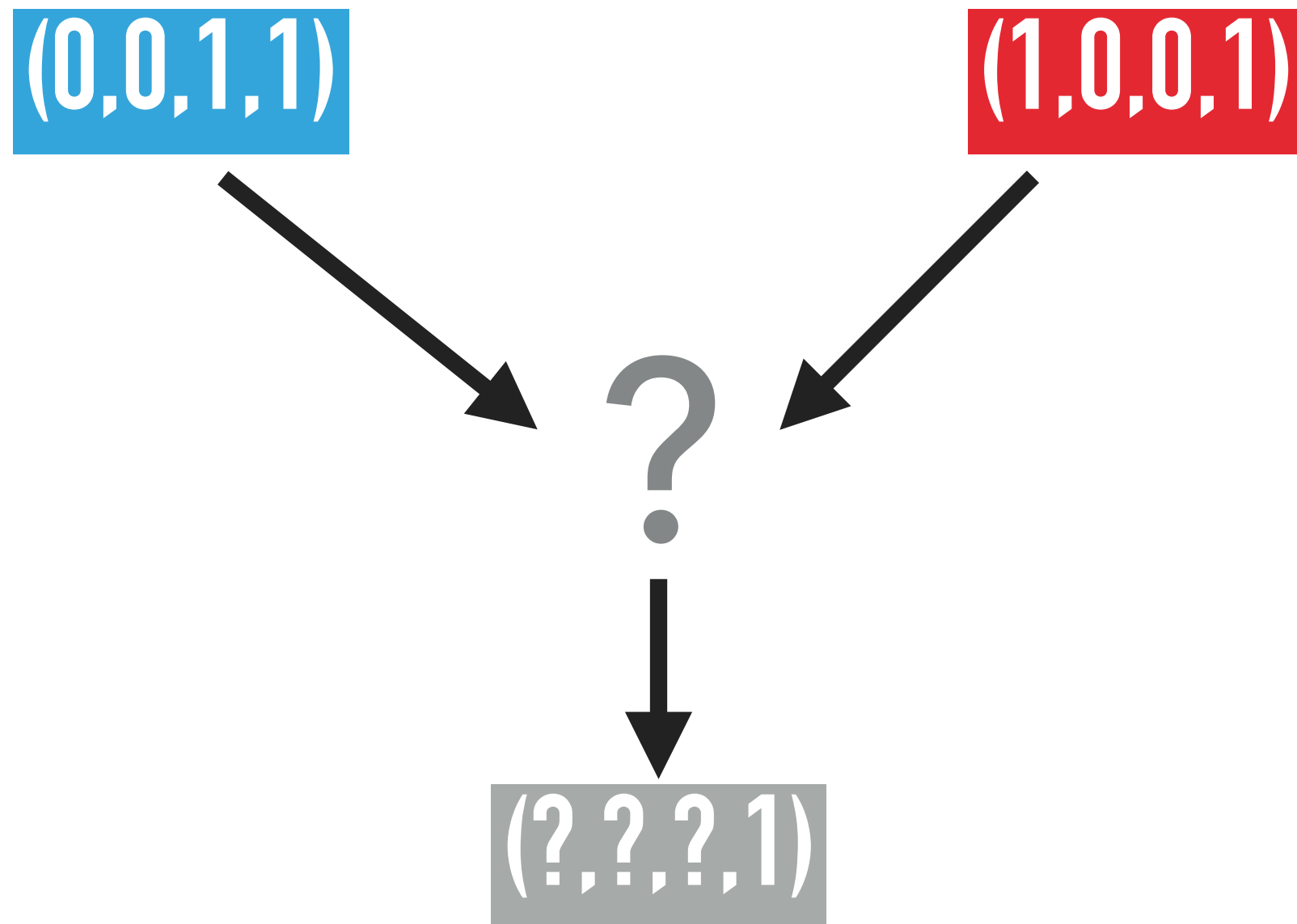
SOLUCIÓN

```
fixed4 frag (v2f i) : COLOR
{
    fixed4 col = tex2D(_MainTex, i.texcoord + fixed2(_Time.x,0));
    return col;
}
```

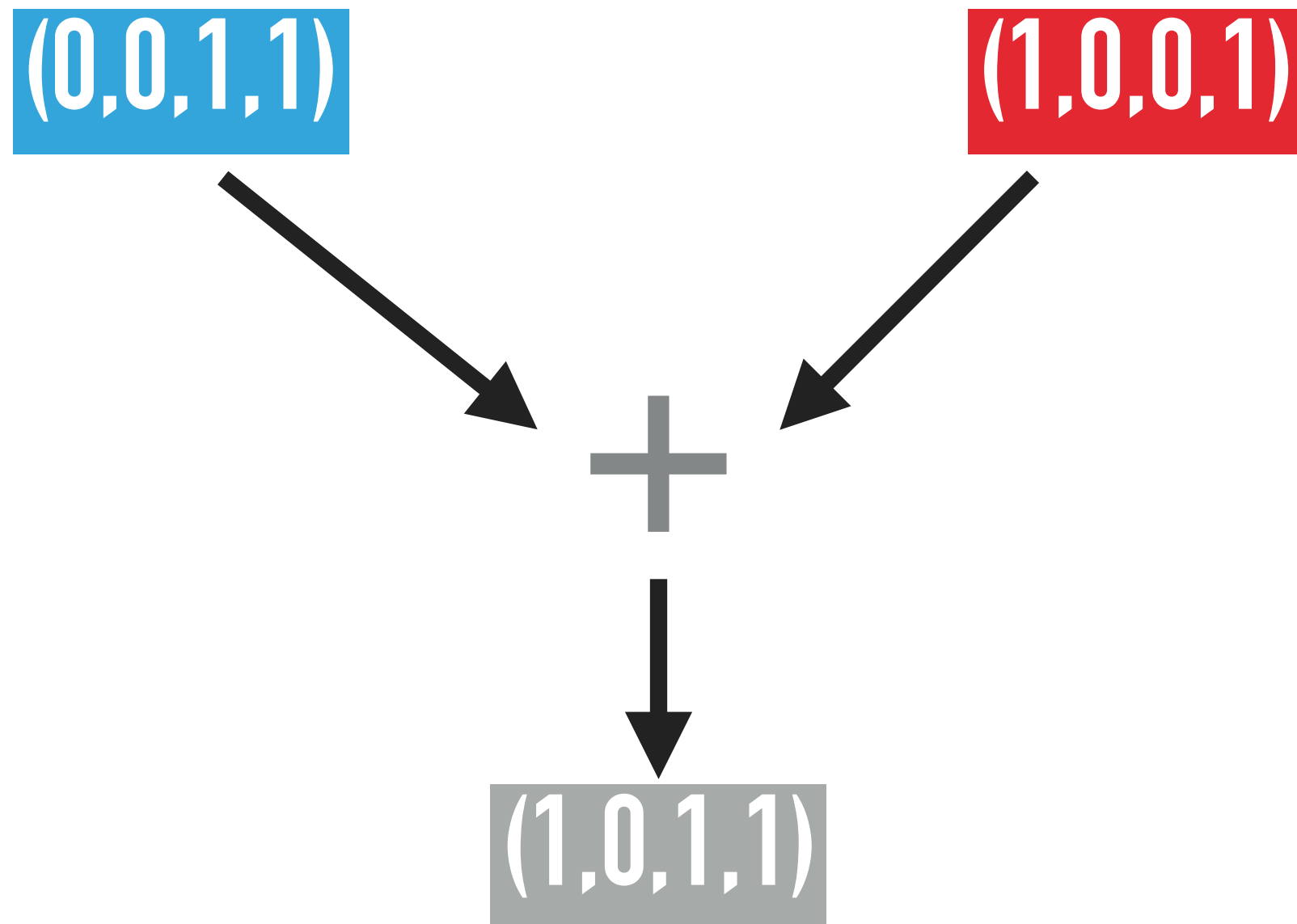
(Y QUE EL WRAP MODE DE LA TEXTURA SEA “REPEAT” Y NO “CLAMP”)

BLENDING MODES

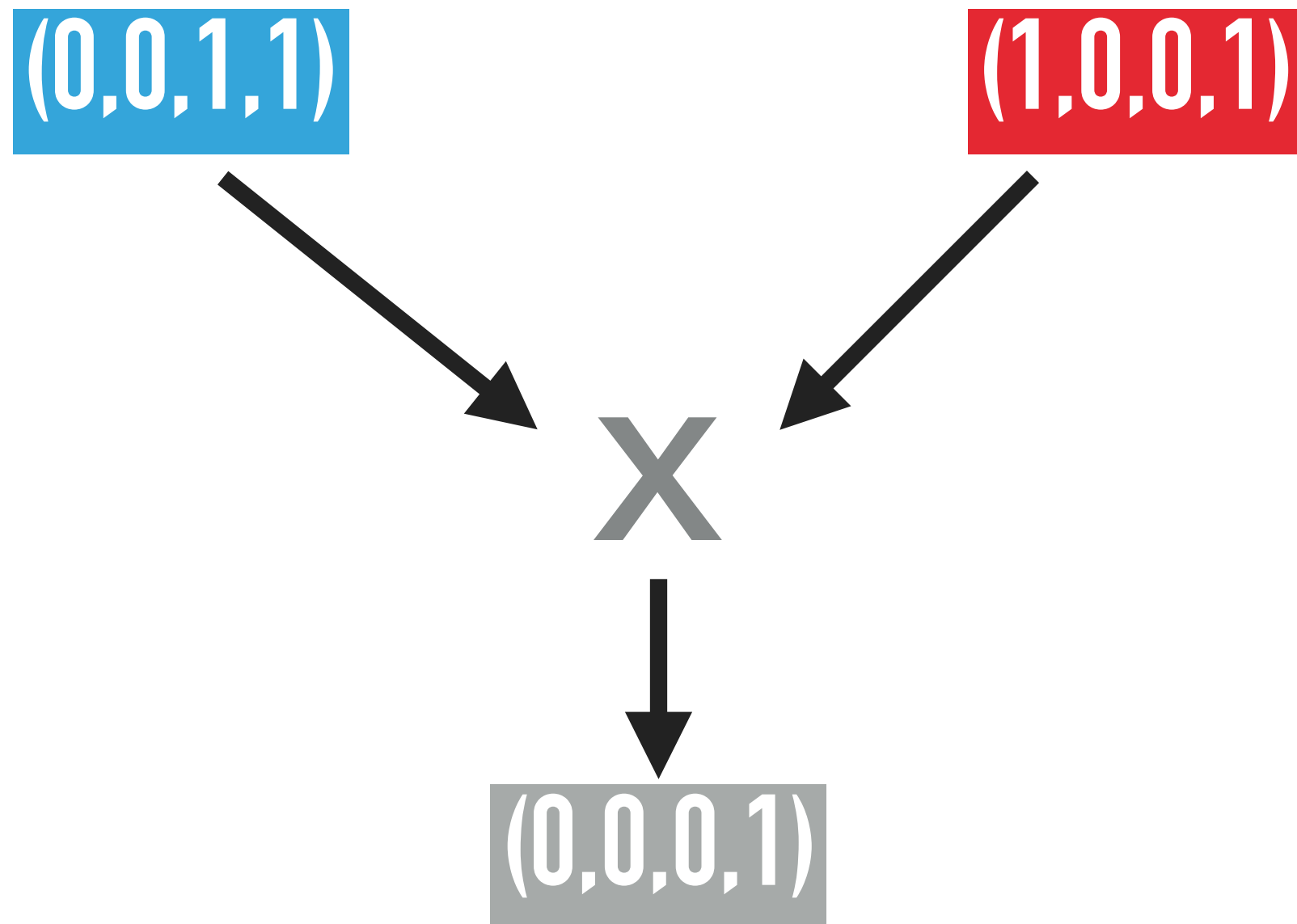
BLENDING MODES



ADDITIVE (ÚTIL PARA ILUMINAR)



MULTIPLY (ÚTIL PARA SOMBREAR)



OTROS BLENDING MODES ÚTILES

Si trabajamos con artistas que usan Photoshop, conviene usar la misma nomenclatura:

MULTIPLY	$A * B$
COLOR BURN	$1 - (1 - B) / A$
LINEAR BURN	$A + B - 1$
SCREEN	$1 - (1 - A) * (1 - B)$
COLOR DODGE	$B / (1 - A)$
LINEAR DODGE	$A + B$
SUBTRACT	$B - A$
DIVIDE	B / A

BLENDING MODES EN UNA SCENE

<http://docs.unity3d.com/Manual/SL-Blend.html>

SPRITES CON ALPHA

```
Blend SrcAlpha OneMinusSrcAlpha
```

TEXT

EFFECTOS FULL SCREEN

EFFECTOS FULL SCREEN

```
using UnityEngine;

public class CameraEffect : MonoBehaviour
{
    public Material material;

    void OnRenderImage ( RenderTexture src, RenderTexture dst)
    {
        Graphics.Blit(src, dst, material);
    }
}
```

EJERCICIO

- 1) Abrir escena en "Ejercicios/2 - Vignetting/"
- 2) Editar shader "Vignetting" para lograr un fade radial desde afuera al centro
- 3) Usar una propiedad para modificar el radio del vignette



HINT: USAR DISTANCIA (LENGTH) Y PROPIEDAD RANGE()

SOLUCIÓN

SOLUCIÓN

```
Properties
{
    _Radius  ("Radius", Range(0,1)) = 0.9
}

[...]

float _Radius;

[...]

fixed4 frag (v2f i) : COLOR
{
    fixed4 col = tex2D(_MainTex, i.texcoord );
    fixed vignetting = _Radius - length(i.texcoord - 0.5);
    return col * vignetting;
}
```

“ILUMINACIÓN” 2D

“ILUMINACIÓN” 2D

Utilizando shaders con **blending mode aditivo**, podemos crear luces para darle más vida a nuestro juego.

“ILUMINACIÓN” 2D

Utilizando shaders con **blending mode aditivo** podemos crear luces para darle más variedad a la escena.

Utilizando shaders con **blending mode multiplicativo** podemos crear sombras que dan profundidad a la escena.

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

Podemos utilizar funciones matemáticas para hacer cosas interesantes.

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

$\sin(x)$

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

$$-1 < \sin(x) < 1$$

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

$$-1 < \sin(x) < 1$$

$$a * \sin(f)$$

$$a = \text{amplitud}$$

$$f = \text{frecuencia}$$

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

`smoothstep(a,b,v)`

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

`smoothstep(a,b,v)`

hace interpolación lineal

entre 0 y 1, comparando a y b con v.

Sirve para hacer transiciones suaves.

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

`smoothstep(0.2,0.3,0.1) = 0`

`smoothstep(0.2,0.3,0.2) = 0`

`smoothstep(0.2,0.3,0.25) = 0.5`

`smoothstep(0.2,0.3,0.3) = 1`

`smoothstep(0.2,0.3,0.5) = 1`

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

lookup tables (LUT)

Efecto muy usado en la demoscene durante los 90s y 2000s por ser poco costoso de ejecutar y lograr muy buenos resultados visuales.

Un poco drástico para usar en videojuegos pero siempre algo se puede hacer.

MODIFICAR LAS TEXTURAS PROCEDURALMENTE

Manejo de canales RGB

Podemos modificar individualmente los canales RGB para hacer correcciones de color o efectos de offsetting.

EJEMPLOS EN NUESTROS JUEGOS

SHADERS

PREGUNTAS?

PARA APRENDER MÁS

<http://docs.unity3d.com/460/Documentation/Manual/SL-Reference.html>

Iñigo Quillez: <http://www.iquilezles.org/www/index.htm>

Libros:

Graphics Shaders: Theory and practice Second edition

Real-Time Rendering Third Edition <http://www.realtimerendering.com/>

Para divertirse y asombrarse:

<https://www.shadertoy.com/> (Proyecto de Iñigo Quillez!)

SHADERS

SANDBOX HASTA LAS 17:30 !!!

A VER QUÉ HICIERON?

MUCHAS GRACIAS!

MUCHAS GRACIAS!

AHORA A ESCRIBIR SHADERS PARA MOSTRAR EN EVA15

info@franciscotufro.com

@franciscotufro