

**INFORMÁTICA GRÁFICA 2**  
**Grado en desarrollo de videojuegos**  
**Curso 2020-21**  
**Práctica 2 (GLSL)**

Todos los scripts de los nuevos materiales deben estar en el archivo [practicaGLSL.material](#).

**1- Mezcla de texturas y zoom.**

Define para el SkyPlane el material [IG2/spaceGLSL](#) de forma que module dos texturas (producto de colores) con las mismas coordenadas de textura. Define los shaders en los archivos [SpaceVS.glsl](#) y [SpaceFS.glsl](#). En el material utiliza las texturas [lightMap.jpg](#) y [numbers.jpg](#) (en el campus) para las unidades de textura 0 y 1 respectivamente.

Parte del [Ejemplo2](#): en lugar de mezclar las texturas haciendo blending hay que multiplicarlas y no se utiliza luz ambiente.

- **Zoom fijo.** Modifica los shaders de forma que el vertex shader pase al fragment shader dos coordenadas de texturas, las de la malla (vUv0) y otras (vUv1) para un zoom centrado de **ZF = 0.5** en las coordenadas s y t. El fragment shader utiliza las primeras (vUv0) para la primera textura ([lightMap](#)) y las otras (vUv1) para la segunda.

Como el intervalo de las coordenadas de textura es [0, 1], para aplicarles una escala primero hay que centrarlas trasladándolas al intervalo [-0.5, 0.5], luego se escalan y por último se deshace la traslación ( $c1 = (c0 - 0.5) * \text{ZF} + 0.5$ )

- **Zoom animado.** Añade al vertex shader una variable [uniform float SinTiempo](#) y utilízala para el factor de escala en lugar del **0.5** fijo (dará lugar a una animación de la textura). Los valores para **ZF** deben estar entre 0.5 y 1 variando en función del tiempo. Utiliza [sintime\\_0\\_2pi](#) con intervalo [32](#) como valor por defecto para [SinTiempo](#)  
[param\\_named\\_auto SinTiempo sintime\\_0\\_2pi 32](#)

Y en el shader, adecua los valores del intervalo [-1, 1] al intervalo [0.5, 1]. Como el intervalo [-1, 1] está centrado, se puede aplicar una escala para ajustar la amplitud del intervalo a 0.5 y luego trasladarlo.

Cuando funcione correctamente cambia en el material la textura [numbers.jpg](#) por [spaceSky.jpg](#). La textura con la animación debe ser [spaceSky.jpg](#).

**2- Material con agujeros e iluminación difusa.**

Cambia la malla de la boya por [uv\\_sphere.mesh](#), una esfera con normales y coordenadas de textura. Implementa para la boya un material con agujeros. El material debe implementar en el fragment shader la iluminación difusa y utilizar un color para el exterior y otro para el interior.

- **Material con agujeros** (de momento sin iluminación)

En el vertex shader tendrás que pasar `gl_Position` en coordenadas de corte y las coordenadas de textura.

En el fragment shader Utiliza dos parámetros para los colores:

```
uniform vec4 InColor;  
uniform vec4 OutColor;
```

Para decidir si un fragmento se corresponde con un punto que está roto utiliza la textura `corrosion.jpg` (en el campus) y comprueba si la componente roja del t xel es mayor que 0.5

Para saber si el fragmento corresponde a la cara front o back utiliza la variable booleana predefinida: `gl_FrontFacing` (-> p gina 22 de Introducci n a GLSL)

Y para descartar fragmentos utiliza el comando GLSL: `discard`

En el script del material, para indicar que se rendericen ambas caras (front y back):

```
cull_hardware none  
cull_software none
```

Y para especificar los colores, en `default_params`

```
param_named InColor float4 r g b a // por ejemplo 0.5 0.5 0.5 1.0  
param_named OutColor float4 r g b a
```

- **Iluminaci n difusa por fragmento** en sistema mundial o de la c mara (front and back faces).

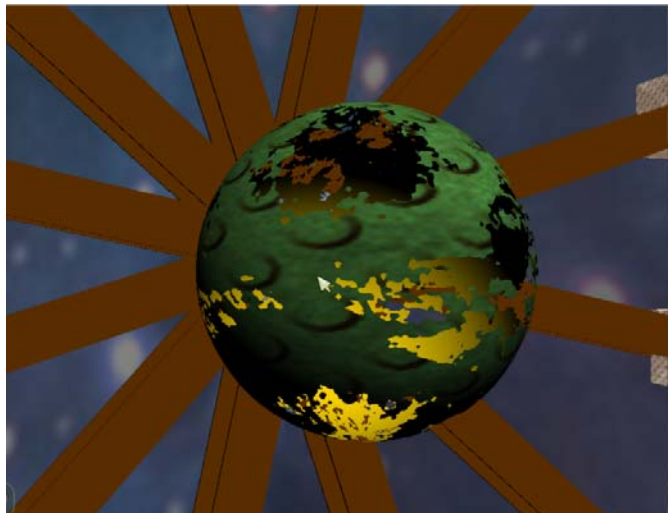
El vertex shader tendr  que pasar al fragment shader, adem s de `gl_Position`, el v rtice y el vector normal en coordenadas mundiales o de la c mara, para lo que har  falta las correspondientes matrices.

El fragment shader calcular  el color correspondiente a la iluminaci n difusa, para lo que har  falta la intensidad de la luz difusa y la direcci n de la luz en coordenadas mundiales o de la c mara. Como coeficiente de reflexi n difusa del material utiliza el color correspondiente a la cara.

En el script del material tendr s que pasar los valores para las variables `uniform`.

A ade una textura a una de las caras de la boya (en la imagen `BumpyMetal.jpg` en el exterior).

Fecha de entrega: 17 de diciembre



### 3- Postprocesado: `Ogre::Compositor`

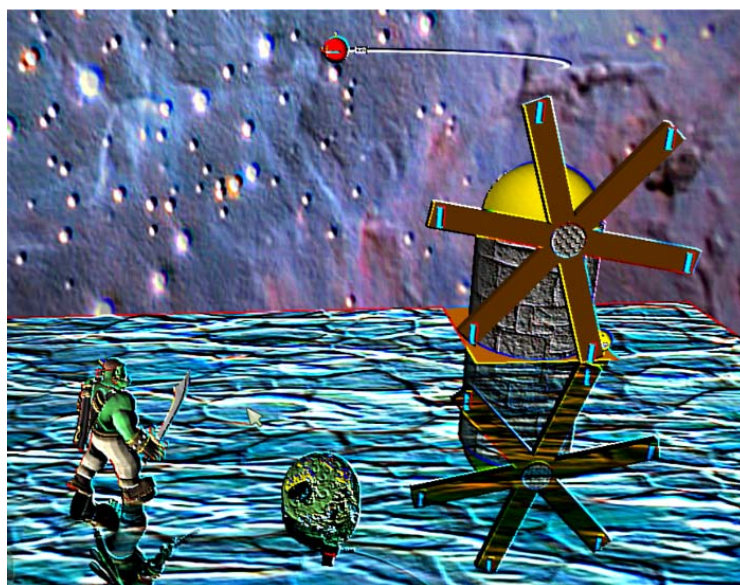
- Define el compositor `IG2/Luminance` para realizar un postprocesado a la imagen de la escena con el material `IG2/LuminancePS`, que transforma los colores de la imagen en escala de grises. Define el `vertex_program RenderQuadVS` e implementa el fragment shader en el archivo `LuminancePS.glsl`.

Añade el compositor al puerto de vista de la cámara en el método `setupScene`.

Utiliza la tecla `SDLK_I` para activar/desactivar el compositor `IG2/Luminance`.



- Define el compositor `IG2/EdgeEmboss` para realizar un postprocesado a la imagen de la escena con el material `IG2/EdgeEmbossPS`, que transforma los colores de la imagen aplicando un filtro para resaltar los borde con la matriz  $\begin{Bmatrix} -4 & -2 & 0 & -2 & 1 & 2 & 0 & 2 & 4 \end{Bmatrix}$ . Utiliza el `vertex_program RenderQuadVS` e implementa el fragment shader en `Kernel3x3PS.glsl`.



El fragment shader recibe como parámetro, además de la textura de la escena, los pesos del kernel3x3 que se quiere aplicar a la imagen: `uniform float kernel[9];`

Define el `fragment_program` Kernel3x3PS que utiliza el fragment shader del archivo `Kernel3x3PS.glsl` y le pasa por defecto la unidad de textura 0, para la textura de la escena, y al parámetro `kernel` los pesos `{ 0, 0, 0, 0, 1, 0, 0, 0, 0 }` con la sintaxis

```
param_named kernel float9 0 0 0 0 1 0 0 0 0
```

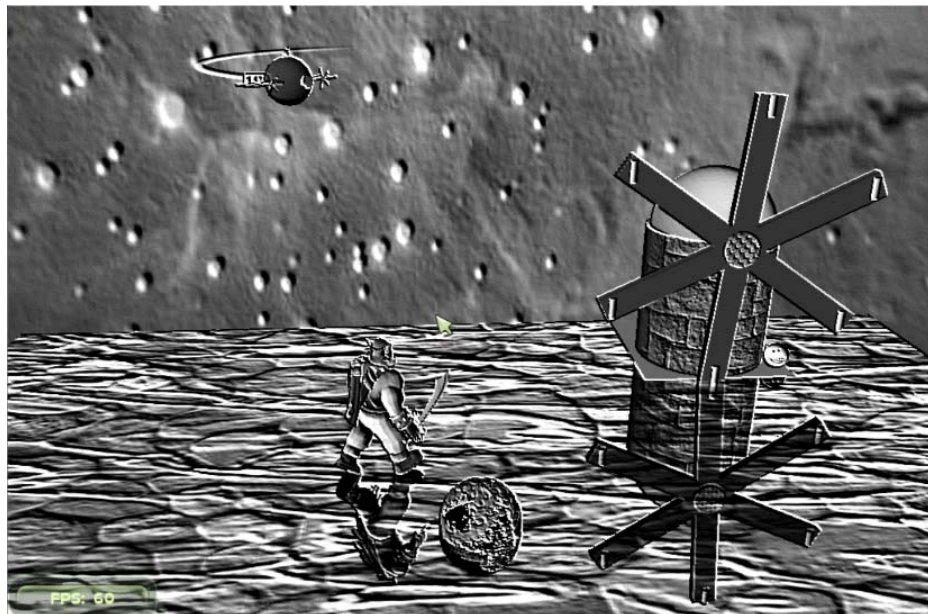
Observa que aplicar este kernel no tiene ningún efecto sobre la imagen.

En el material `IG2/EdgeEmbossPS`, cuando se indica el `fragment_program` que se utiliza, pasa el kernel del ejercicio.

Añade el compositor al puerto de vista de la cámara en el método `setupScene` y comenta el anterior.

Utiliza la tecla `SDLK_k` para activar/desactivar el nuevo compositor `IG2/EdgeEmboss`.

- Encadena ambos compositors. Las teclas `l` y `k` siguen funcionando para cada compositor.



Fecha de entrega: miércoles 13 de enero