

INFORMÁTICA GRÁFICA 2

Grado en desarrollo de videojuegos

Curso 2020-21

Práctica 2.0

Billboards and Particle Systems

- Cartel

Añade un cartel (**BillboardSet**) a la cola del avión de forma que siempre mire a la cámara y siga al avión. Crearlo en la clase **Avion**.

<https://ogrecave.github.io/ogre/api/latest/index.html> -> **BillboardSet**

- Explosión

- Añade una explosión (**ParticleSystem**) al avión. Define en el archivo **IG2App.particle** un sistema de partículas **IG2App/Explosion** para el humo de la explosión de forma que el humo se extienda en todas las direcciones. Utiliza el material del ejemplo (**IG2App/Smoke**). El color (atributo **colour**) por defecto de las partículas es blanco opaco, cámbialo por gris oscuro traslúcido. En lugar del **affector colourImage** utiliza **colourFader** para incrementar la transparencia del color y aclararlo hasta llegar a blanco transparente. Crea el sistema de partículas en la clase **Avion**.

<https://ogrecave.github.io/ogre/api/latest/particle-scripts.html>

Observa que los incrementos (decrementos) de las componentes del color en **colourFader** son por segundo. Hay que coordinar los atributos **time_to_live**, **colour** y **colourFader** para llegar al color final deseado.

Los atributos **quota**, **emission_rate**, y **time_to_live** también deben coordinarse.

El ángulo de emisión se establece entre 0 (todas las partículas se emiten en la misma dirección) y 180 (las partículas se emiten en cualquier dirección).

- Modifica la respuesta del avión al aviso del ejercicio 31 para que explote y desaparezca. Utiliza el método **setVisible** de **MovableObject** o de **SceneNode**.

<https://ogrecave.github.io/ogre/api/latest/index.html> -> **setVisible**

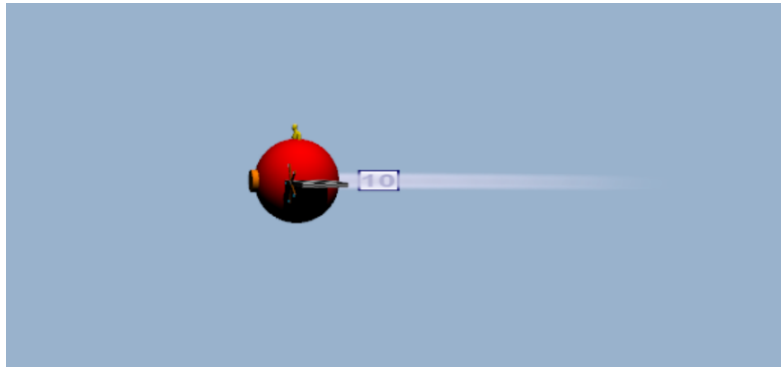
Además, Sinbad responde al aviso de forma que deja de correr, cae tumbado bocarriba y queda en estado de animación **IdleTop**.



- Estela

- Añade una estela ([ParticleSystem](#)) al avión. Define en el archivo [IG2App.particle](#) un sistema de partículas [IG2App/SmokeTrail](#) para la estela de humo blanco del avión. Utiliza el material del ejemplo ([IG2App/Smoke](#)). Establece el color (atributo [colour](#)) a gris, casi blanco, traslúcido. Utiliza [colourFader](#) para incrementar la transparencia del color hasta terminar en transparente. Adecua el tamaño de las partículas. El humo casi no se mueve, lo hace el avión. Crea el sistema de partículas en la clase [Avion](#).

https://ogrecave.github.io/ogre/api/latest/_particle_scripts.html



Multitexturing and RenderTextures

Limpia el código, deja solo la escena de la práctica 1.5, incluida la boya y los últimos ejercicios, pero quita el foco del avión y las sombras (también puedes quitar el suelo rojo), y deja la luz direccional encendida con la dirección (0, -1, -1) y componente difusa (1.0, 1.0, 1.0).

Configura Ogre seleccionando OpenGL 3+ Rendering Subsystem

- [SkyPlane](#)

Define el material [IG2/space](#) de forma que no refleje luz y module dos texturas (producto de colores) con las mismas coordenadas de textura. Utiliza las texturas [lightMap.jpg](#) y [spaceSky.jpg](#) (ambas en el campus) para las unidades de textura 0 y 1 respectivamente. Configura una animación de rotación para spaceSky.

https://ogrecave.github.io/ogre/api/latest/_material_scripts.html -> Texture Units -> colour_op

Recuerda que la operación de mezcla de una unidad de textura se realiza sobre el resultado de las anteriores. En el caso de la primera unidad, la operación se realiza sobre el color de la iluminación. Experimenta con las opciones de textura en el material. Utiliza numbers.jpg en lugar de [spaceSky.jpg](#) para las pruebas.

Añade a la escena un **SkyPlane** vertical con el material **IG2/space** (puedes hacerlo en el método **setupScene** de la clase **IG2App**). Experimenta con los parámetros del método **setSkyPlane** (normal y distancia del plano, curvatura). Deja comentado en una línea la generación del Skyplane con parámetros que lo configuran para que sea plano (sin curvatura) y esté a una distancia de 200.

<https://ogrecave.github.io/ogre/api/latest/index.html> -> **setSkyPlane**

Establece el color de fondo a negro.

- Reflejo

Define el material **IG2/reflejo** para el rio de forma que tenga las texturas piedras y agua mezcladas, además de los coeficientes de reflexión. Recuerda que el agua se mueve y que, como respuesta al aviso del ejercicio 31 el agua desaparece.



Añade a la clase del rio un método **setReflejo** con un argumento para la cámara (**Camera***) que añada una textura dinámica en la que renderizar el **reflejo** en el agua. Añade un atributo (**MovablePlane***) para configurar el plano del reflejo de la nueva cámara (recuerda eliminarlo en la destructora).

Opcional: soluciona el efecto de la cabeza del muñeco en el reflejo y el problema que se produce al ampliar la ventana.

Fecha de entrega: 2 de diciembre