

Informática Gráfica II
Examen de la convocatoria extraordinaria (junio/25)
Curso 24/25

Una empresa dedicada a la grabación de películas de acción nos ha encargado el diseño de una escena para su última película. En la escena pedida, debemos diseñar un avión con distintos efectos.

Para el desarrollo del examen se proporciona la clase **DataSizes**, la cual contiene constantes que pueden utilizarse para implementar las clases pedidas. La escena se configura en la clase **IG2App**. Se tendrá en cuenta la calidad de código, evaluando su claridad, usabilidad y estructura.

La carpeta **media/IG2App** debe contener las texturas que se utilicen para el desarrollo del examen. Además, en esta carpeta deben incluirse los ficheros de materiales, sistemas de partículas y shaders.

Apartado 1. Diseño del avión [2 puntos]

En este apartado se realizará el diseño del avión, que está formado por varias partes: el cuerpo, dos alas, un timón (*rudder*) y dos motores. El código del avión estará en una clase llamada **Airplane**, mientras que el código del motor estará en la clase **Engine**.

Para el cuerpo usaremos una esfera (malla `uv_sphere.mesh`). Las alas y el timón serán cubos (malla `cube.mesh`). Estos elementos deberán escalarse adecuadamente utilizando las medidas indicadas en las constantes **AIRPLANE_BODY_SIZE**, **AIRPLANE_WING_SIZE** y **AIRPLANE_RUDDER_SIZE**, respectivamente.

Cada motor del avión lo forman dos partes: la base y los cohetes (*rockets*). La base del motor se puede representar con un cilindro (malla `Barrel.mesh`), cuyas medidas se indican en la constante **ENGINE_BASE_SIZE**. Los cohetes los representaremos con esferas (también con la malla `uv_sphere.mesh`) de forma que cada uno sea adyacente a la base del motor. La constante **ROCKET_SIZE** define el tamaño de cada cohete.

Los cohetes deben posicionarse de forma equidistante en la parte exterior del círculo que forma la base del motor. El número de cohetes es configurable y se indica en la constante **AIRPLANE_NUM_ROCKETS**, es decir, al cambiar su valor, se debe generar la configuración correspondiente para el motor cumpliendo con los requisitos del enunciado.

Las siguientes figuras muestran distintas vistas del avión completo. **No serán consideradas válidas** las soluciones que no tengan en cuenta la constante **NUM_ROCKETS**, o la colocación de los cohetes según se describe en el enunciado.

Nota: En este apartado no es necesario aplicar color o texturas a los objetos. Las imágenes se han realizado así para poder distinguir mejor las partes del avión.

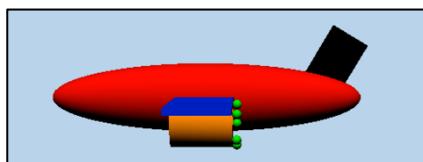


Figura 1: Vista lateral

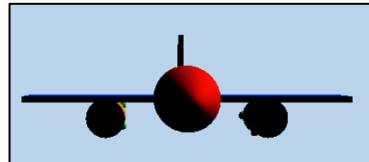


Figura 2: Vista frontal

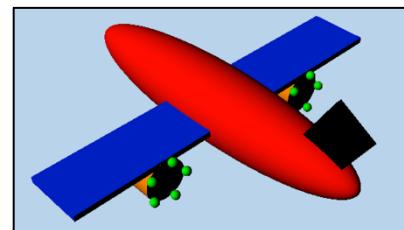


Figura 3: Vista superior

La Figura 1 muestra el avión desde una vista lateral. La Figura 2 muestra el avión desde una vista frontal. Finalmente, la Figura 3 muestra una vista superior. En las imágenes se aprecia el cuerpo del avión en rojo, las alas en azul, el timón en negro, la base del motor en naranja y los cohetes en verde. Tened en cuenta que, en estos ejemplos, se ha renderizado el avión con la constante `NUM_ROCKETS=5`.

Apartado 2. Texturas del avión (0.75 puntos)

En este apartado aplicaremos texturas al avión. Para ello, incluiremos en el fichero `examen.material` los materiales correspondientes. Los ficheros que utilizaremos para aplicar los materiales serán los siguientes:

- Cuerpo del avión: `corrosion.jpg`
- Alas y timón: `RustySteel.jpg`
- Base del motor: `MtlPlat2.jpg` (Utilizad `lighting off` en el pase)
- Cohetes: `BumpyMetal.jpg`

La Figura 4 muestra el avión con las texturas aplicadas.

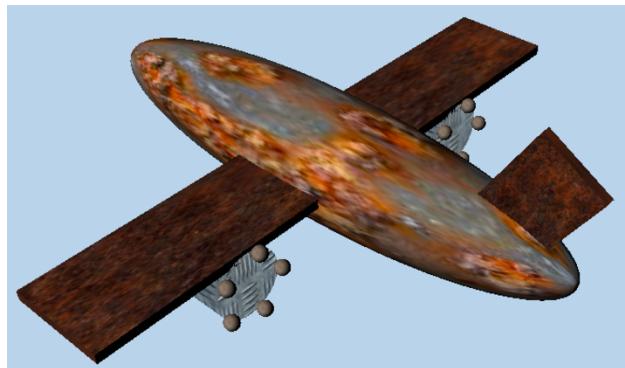


Figura 4: Avión con texturas

Apartado 3. Suelo y cielo (1.25 puntos)

Una vez finalizado el avión, crearemos el suelo y el cielo.

El suelo tendrá unas dimensiones de 5000x5000. Además, se aplicará un material para recrear el efecto que se muestra en el vídeo `examenVideo1.mov`. Para conseguir este efecto se han utilizado **dos ficheros de imagen**, que pueden encontrarse en la carpeta `ficherosImagen`, y deben copiarse en el directorio `media/IG2App`.

El cielo deberá mostrarse de forma similar a como se aprecia en el vídeo `examenVideo1.mov`. El fichero de imagen necesario para mostrar el cielo se encuentra también en la carpeta `ficherosImagen`. En este caso, no se han utilizado shaders, únicamente se ha configurado un material.

La Figura 5 muestra el efecto descrito en el suelo y el cielo.



Figura 5: Escena con el suelo y el cielo

Apartado 4. Movimiento (2 puntos)

En este apartado añadiremos varios movimientos al avión.

Para ello, colocaremos el avión en el extremo superior del plano (suelo) tal y como se ve en la figura.



Figura 6: Posición inicial del avión para realizar el movimiento

El avión debe realizar un movimiento en círculo, siendo el centro del círculo, el centro del plano. El avión girará en el círculo en sentido antihorario (contrario a las agujas del reloj) a una velocidad de `AIRPLANE_SPEED` grados por segundo, teniendo en cuenta que los grados que avanza lo forman los segmentos correspondientes entre el centro del círculo y su posición actual, y el centro del círculo y su posición final. Tened en cuenta que en el método `frameRendered` podemos conocer el tiempo que ha transcurrido entre el *frame* actual y el anterior.

Además, el avión podrá realizar giros. Para ello utilizaremos las teclas. Al pulsar la tecla ‘a’ indicaremos que el avión gire, sobre el eje X, **AIRPLANE_ROTATION** grados por segundo, mientras que si pulsamos la tecla ‘d’ indicaremos que avión gire, sobre el eje X, **-AIRPLANE_ROTATION** grados por segundo.

Para realizar los giros NO es necesario dejar las teclas presionadas. Es decir, al presionar la tecla ‘a’, o la tecla ‘d’, **una vez**, el avión permanecerá rotando hasta que se presione la tecla que indique un giro opuesto, o bien la tecla ‘s’, que indica que el avión ya no gira, manteniendo su posición mientras realiza el movimiento en círculo.

Se penalizarán las soluciones que no realicen el movimiento en función del tiempo transcurrido. Podéis ver ambos movimientos del avión en el vídeo **examenVideo1.mov**

Apartado 5. Sistemas de partículas (2 puntos)

Para darle más realismo a la escena vamos a desarrollar dos sistemas de partículas. El vídeo **examenVideo1.mov** muestra un ejemplo de estos sistemas de partículas.

El primer sistema representará el humo de los cohetes. Para ello, utilizaremos dos colores: humo blanco para los cohetes en posiciones pares, y humo gris oscuro para los que están en posiciones impares. Estos sistemas de partículas permanecerán activos – ininterrumpidamente – desde el comienzo de la escena. Algunos de los valores que se han utilizado para estos sistemas de partículas son:

- quota: 1000
- time_to_live: 10
- velocity_min: 150 y velocity_max: 200
- tamaño de cada partícula: 10x10
- Dos modificadores para el color y la escala de las partículas.

El segundo sistema de partículas representará el fuego que sale del sistema de propulsión (ubicado en la parte trasera del cuerpo del avión). Inicialmente permanecerá desactivado. Para activar o desactivar este sistema de propulsión, se debe pulsar la tecla tecla ‘w’ (en minúscula). Cuando esté activo, el avión se moverá al doble de velocidad cuando realice el movimiento en círculo. Sin embargo, los giros (teclas ‘a’ y ‘d’) siempre se realizan a la misma velocidad, esté activo el sistema de propulsión, o no.

Algunos de los valores que se han utilizado para este sistema de partículas son:

- quota: 500
- time_to_live: 6
- velocity_min: 150 y velocity_max: 180
- tamaño de cada partícula: 35x35

Los sistemas de partículas se definirán en el fichero **examen.particle**. La Figura 7 muestra los dos sistemas de partículas activos.



Figura 7: Sistemas de partículas en los motores y en el sistema de propulsión

Apartado 6. Shader (2 puntos)

En este último apartado vamos a desarrollar un material nuevo que aplicará un shader a los cohetes de los motores. Este material contendrá dos shaders: uno de vértices y otro de fragmentos. Dejad en el código la línea que asigna el material de los cohetes -- relativa al apartado 2-- comentada.

En particular, vamos a desarrollar un material – llamado `exam/coloredRocket` – que, al aplicar los shaders, dará color a la esfera de los cohetes. El efecto puede verse en el vídeo `examenVideo2.mov`, donde la esfera va tomando distintos colores a modo de gradiente según avanza el tiempo. El objetivo es que las esferas vayan modificando su color gradualmente, de forma similar a como se ve en la escena. Cabe destacar que en la parte derecha de la esfera (según se aprecia en la imagen y en el vídeo) siempre se asignan colores claros, mientras que en la parte izquierda, son oscuros.

La siguiente imagen muestra un *frame* de la escena donde se aplican los shaders. No es necesario que los colores sean exactamente los mismos, el objetivo es conseguir el efecto descrito.



Figura 8: Imagen de la escena aplicando los shaders correspondientes