



**UNIVERSIDAD  
DE GRANADA**

# CONVOCATORIA EXTRAORDINARIA

MAURICIO LUQUE JIMÉNEZ

10 DE FEBRERO DE 2025

INFORMÁTICA GRÁFICA

## MODELO JERÁRQUICO

### TRANSFORMACIONES GEOMÉTRICAS Y ANIMACIONES

Además del modelo jerárquico desarrollado en el código relativo a la P3 (explicado en su respectivo grafo de escena), en el proyecto final se incluyen tres objetos más que hacen uso de transformaciones geométricas.

Primero, un cubo uniforme (1.0, 1.0) al que se le aplica un escalado y una transformación para que sirva como carretera para que los dos objetos principales de modelo jerárquico, es decir, el coche y el objeto *sims* puedan simular una carrera de ida y vuelta.

En segundo lugar, un coche representado por una malla de triángulos a la que se aplican transformaciones de escalado y traslación.

En tercer lugar, un objeto *sims* que supone la combinación de otros tantos objetos más simples. Para empezar, está compuesto de un objeto *baymax* y de un *cursor* idéntico al que aparece en el popular juego Sims sobre el personaje que controla el jugador. Este cursor, a su vez, es la combinación de dos pirámides, estando una de ellas invertida y trasladada para que su cara encaja exactamente con la de la otra pirámide. Cuando se entra en modo animación, este cursor, que siempre acompaña el movimiento del objeto *baymax*, realiza un movimiento vertical en el eje Y para animar el icono de selección, como podría ocurrir en el videojuego original. Por otra parte, el objeto *baymax* está compuesto de cinco partes independientes (todas ellas representadas por una malla *ply*): el cuerpo y cada una de las extremidades. De esta manera, se aplican transformaciones individuales a cada una de las extremidades para simular animaciones de movimiento. Así pues, cuando se entra en el modo animación, el objeto *sims* realiza tres movimientos diferentes simultáneamente. Primero, el movimiento que traslada tanto a *baymax* como al cursor; por otra parte, la traslación vertical que realiza el propio cursor; y por último, las rotaciones individuales que realizan las extremidades de *baymax*. Esta estructura permite que los elementos dentro de *sims* se muevan de manera sincronizada y permite manipular cada objeto individualmente sin afectar al resto de la escena.

Además, en un extremo de la carretera se sitúa una mesa que sostiene un trofeo, concretamente el trofeo del Gran Premio de Mónaco, el más aclamado de la Fórmula 1. El trofeo está trasladado con la mesa, de manera que siempre está encima de ésta. Además, se le incluye una animación en la que gira sobre el eje Y. Esta animación siempre está activa cuando no es el objeto seleccionado: tanto si no se selecciona ningún objeto, como si se selecciona cualquier otro, siempre va a estar rotando sobre sí mismo.

## TEXTURAS

### APLICACIÓN DE TEXTURA DE CARRETERA AL CUBO *ROAD*

Para la aplicación de texturas, se ha aplicado el siguiente procedimiento. En primer lugar, se carga la textura desde un archivo JPEG. En este caso, se carga la imagen *road.jpeg* utilizando *LeerArchivoJPEG()*, que devuelve un puntero a un array de píxeles en formato RGB.

Una vez se ha cargado la textura, se configura en el objeto *road*. Para ello, se pasa la imagen a la función *setTextura()*, que se encarga de:

- Generar una textura con *glGenTextures(1, &textid)*.
- Configurar los parámetros de repetición y filtrado con *glTexParameterf()*.
- Asociar la imagen a la textura con *glTexImage2D()*.

En este punto, es interesante destacar que, si se quieren cargar varias texturas, podría ser útil realizar esta lectura y carga de imágenes dentro de la función *InitModel()*, para que la textura fuera leída y cargada una sola vez, permitiendo liberar memoria después de leerla. Además, no sólo es útil en el tema de memoria, sino que evita sobreescrituras a la hora de asignar las texturas.

Una vez tenemos la textura cargada, podemos proceder al renderizado con la textura activada. Para ello, se transforma el cubo *road* para posicionarlo en la escena. A continuación, la función *drawTextura()* habilita la textura con *glEnable(GL\_TEXTURE\_2D)*, la vincula con *glBindTexture(GL\_TEXTURE\_2D, textid)*, y dibuja la cara superior del cubo con coordenadas de textura.

## SELECCIÓN

### FUNCIÓN *PICK* Y SU EFECTO EN LAS ANIMACIONES

Para la selección, se crea una variable booleana *seleccion* que, cuando se activa, desactiva el cálculo de iluminación y texturas para dibujar la escena en un modo especial. Cada objeto se dibuja con un único color, correspondiente a su *idSeleccion* asignado en el momento de crear cada objeto. Este dibujado con un único color se realiza con la función *colorSeleccion()*.

Para identificar el objeto seleccionado, el programa lee el color del pixel en el momento en el que se hace clic izquierdo. Se obtiene el ID del objeto a partir del color del pixel y se almacena en *idSeleccionado*.

Esta selección influye altamente en las animaciones establecidas en la P3, ya que hay diferentes comportamientos según el objeto que esté seleccionado.

- Si no hay selección (*idSeleccionado == 0*), todos los objetos animables se mueven.
- Si se selecciona el coche (*idSeleccionado == 1*), solo el coche y el trofeo se animan.
- Si se selecciona *sims* (*idSeleccionado == 2*), solo Baymax y el trofeo se animan.
- Si se selecciona la mesa (*idSeleccionado == 3*), solo el trofeo se anima.
- Si se selecciona el trofeo (*idSeleccionado == 4*), no se anima ningún otro objeto.

Dentro de la función *Dibuja*, para remarcar el objeto seleccionado de una forma visible, se utiliza el siguiente código:

```
if (idSeleccionado == sims.getIdSeleccion())  
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, selectedRed);  
else  
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, red);
```

De manera que se puede identificar a simple vista qué objeto es el que está seleccionado.