

**GRADO EN DISEÑO**  
**Y**  
**DESARROLLO DE VIDEOJUEGOS (MÓSTOLES) (2020-21)**

**Informática Gráfica**

**Práctica 3: Introducción a OpenGL**

**Participantes del grupo:**

Pablo Burgaleta de la Peña

Raquel Mijarra Benítez

Adrián Mira García

Iván Pérez Ciruelos

## **Parte obligatoria**

### **1. Modifica las propiedades (intensidad y posición) de la luz a través del teclado.**

Para la implementación de este apartado y poder controlar la intensidad y la posición de la luz por teclado hemos creado unas variables uniform que son las que vamos a modificar por el teclado y mandamos a los shaders de fragmentos.

Las funciones de teclado para modificar las propiedades de la luz son:

- 'f': mover la luz a la izquierda
- 'h': mover la luz a la derecha
- 't': mover la luz hacia arriba
- 'g': mover la luz hacia abajo
- 'u': mover la luz hacia delante
- 'j': mover la luz hacia detrás
- 'i': aumentar la intensidad de la luz
- 'k': disminuir la intensidad de la luz

### **2. Define una matriz de proyección que conserve el aspect ratio cuando cambiamos el tamaño de la ventana.**

Para conservar el aspect ratio recibimos el ancho y el alto de la ventana en la función `resizeFunc` para crear una variable `aspectratio` que conserve las proporciones y se lo aplicamos a la matriz de proyección.

### **3. Añade un nuevo cubo a la escena. El segundo cubo deberá orbitar alrededor del primero describiendo una circunferencia a la vez que rota sobre su eje Y. NOTA: No tienes por qué crear otro objeto. Pinta el mismo VAO con otra matriz model.**

Para este apartado hemos renderizado el mismo otro cubo similar al cubo principal (mismo vao) pero cambiando la matriz model (`model2`) para poder cambiarlo de posición.

### **4. Utiliza un programa diferente para cada cubo. Deberéis usar 2 parejas de shaders diferentes para cada cubo. NOTA: Tendréis que pensar cómo manejar todas las variables que almacenan identificadores de cada programa.**

Para este apartado hemos replicado la creación del `shader1` y del `programa1`, teniendo así un `shader2` y un `programa2`. Además de replicar la creación del shader y del programa hemos mirado las variables que se utilizan en ambos casos para duplicarlas y poder crear otro cubo con éxito.

**5. Control de la cámara con el teclado. Controles mínimos que deberán incluirse: movimiento hacia adelante, retroceso, movimientos laterales (izquierda y derecha) y giros (izquierda y derecha). Importante: la posición de la luz debe ser invariante con respecto a la posición de la cámara.**

Para este apartado hemos utilizado lo aprendido en la práctica 1 sobre el control de la cámara por teclado, en este caso hemos creado una matriz view auxiliar, la cual varía dependiendo de las entradas por teclado (w,a,s,d).

Después para aplicar los cambios multiplicamos la matriz view por la matriz view auxiliar.

Para la realización de la práctica hemos obtenido la información de las siguientes páginas web:

- <http://acodigo.blogspot.com/2016/04/modelo-de-iluminacion-phong-tutorial.html>
- <http://acodigo.blogspot.com/2016/05/texturas-e-iluminacion.html>
- <http://www.ieef.upm.es/webantigua/spain/Asignaturas/InfoInd/teoria/Apuntes%20de%20OpenGL.pdf>
- <https://es.stackoverflow.com/questions/206323/mantener-aspect-ratio-al-cambiar-tamaño-de-ventana>
- <http://acodigo.blogspot.com/2016/04/tutorial-opengl-camara.htm>
- <https://learnopengl.com/Getting-started/Camera>
- <http://www.opengl-tutorial.org/es/beginners-tutorials/tutorial-6-keyboard-and-mouse/>
- <https://fabiensanglard.net/bumpMapping/index.php>