

# OpenGL

## Tarea 06

### Computación Gráfica

#### UNAM 2022-2

Gibran Zazueta Cruz

10/mayo/2022

## 1. Introducción

Se presenta un programa que utiliza las librerías de OpenGL y QT para renderizar un objeto en una ventana. Asimismo, se utiliza la librería de Assimp para cargar un archivo formato Wavefront OBJ. EL objeto a renderizar es el conejo de stanford

En este programa se hace uso de la *Fixed Function Pipeline* de OpenGL, es decir, no se utilizan shaders. La versión de OpenGL utilizada es la 2.1

## 2. Escena y materiales

El objeto cuentan con 2 posibles materiales a seleccionar, estos se definen dentro del código como:

### Material 1

- Ambiental = 0.0, 0.0, 0.0, 1.0,
- Difusa = 0.50, 0.50, 0.50, 1.0,
- Especular 0.70, 0.70, 0.70, 1.0
- $\rho = 32.0$ .

### Material 2

- Ambiental = 0.23125, 0.23125, 0.23125, 1.0,
- Difusa = 0.2775, 0.2775, 0.2775, 1.0,
- Especular 0.773911, 0.773911, 0.773911, 1.0
- $\rho = 89.6$ .

Se utilizan las texturas



Figura 1: Textura material 1



Figura 2: Textura material 2

Finalmente, la escena a renderizar cuenta con 2 luces (blanca y azul) y 4 cámaras. Se disponen de la siguiente manera:

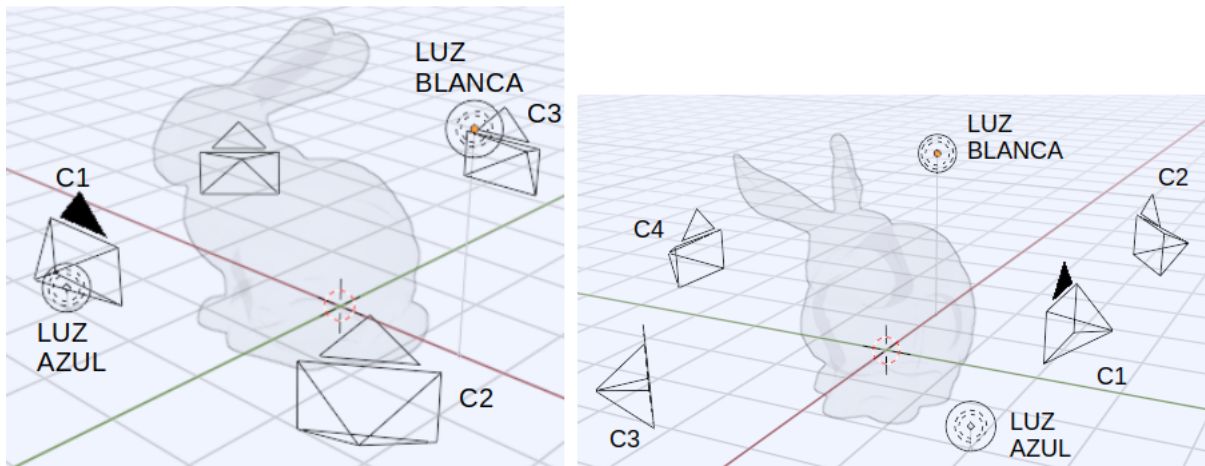


Figura 3: Posición de las componentes de la escena

### 3. Estructura del código

Para almacenar la información del objeto se define la clase *CubeObject* y la la función *import-File()* que está definida en *functions.cpp*.

Esta función se llama desde *mainwindow.cpp*. La función recibe el path del archivo (como una cadena `std::string`) y apuntadores al contenedor de vertices, indices de caras, normales y coordenadas de textura del objeto `cubeObject`, que es el objeto a renderizar en la escena. Dentro de esta función se utiliza la librería de `assimp`.

Dentro de `mainwindow` se le dice a qt que utilice OpenGL para dibujar sobre el canvas, esto es con la función `setSurfaceType(QWindow :: OpenGLSurface)`. También se definen funciones de interacción de ventana como son: *resizeGL*, *initializeGL*, *paintGL*.

Es en *paintGL* donde se dibuja el objeto. Se especifican cámaras, materiales, luces y texturas. Los vértices se definen dentro de un ciclo *for* para todos los triángulos del objeto.

## 4. Ejecutar el programa

En la carpeta de build se puede ejecutar el programa con el archivo `OpenGLRendering-Run`. Desde la consola de comandos de linux:

```
bash
```

```
./OpenGLRendering-Run
```

En la carpeta principal está el código fuente. Para generar el ejecutable primero se genera el Makefile con

```
bash
```

```
qmake OpenGLRendering.pro
```

Después se construye el proyecto con *make*

## 5. Instrucciones de uso

Para cambiar entre las camaras se utilizan las teclas de los numeros

- "1". Cambia a la cámara 1
- "2". Cambia a la cámara 2
- "3". Cambia a la cámara 3
- "4". Cambia a la cámara 4

Para encender y apagar la luz se presiona la tecla **6**. La luz inicia encendida.

Para cambiar entre materiales se utiliza la tecla **8** para el material 1 y la tecla **9** para el material 2.

## 6. Programa en ejecución



Figura 4: Renderizado cámara 1, materiales 1 y 2



Figura 5: Renderizado cámara 2, 3 y 4. Material 2