



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



EJERCICIOS DE CLASE N° 07

NOMBRE COMPLETO: Vázquez Reyes Sebastián

N° de Cuenta: 318150923

GRUPO DE LABORATORIO: 11

GRUPO DE TEORÍA: 6

SEMESTRE 2024-2

FECHA DE ENTREGA LÍMITE: Domingo 24 de marzo de 2024

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

- **Actividades realizadas. Una descripción de los ejercicios y capturas de pantalla de bloques de código generados y de ejecución del programa**

1. **Agregar su propio coche texturizado con la jerarquía de llantas, crear la luz de faro del coche de color azul y posicionar a que ilumine hacia adelante y se mueva con el coche**

Los modelos y texturas para el coche y las llantas vienen en el .zip agregado a la entrega. Para este trabajo, implemente el mismo código de la practica 5 para importar el modelo del carro, sus llantas. Ahora, las texturas de estos elementos fueron colocadas mediante 3DSMAX así que ya no hace falta invocarlas en el código.

Para implementar el movimiento del carro y la luz, tuve que modificar los archivos window.cpp y window.h, agregando las funciones que reconocen la pulsación de las teclas para las articulaciones1, 2, 3, etc. Cambios que si bien son importantes, ya conocemos y usamos varias veces, por lo que no es necesario mostrarlos.

Para la luz que se mueve con el carro, tengo el siguiente bloque de código:

```
//Luz del carro
spotLights[2] = SpotLight(0.0f, 0.0f, 1.0f,
    1.0f, 2.0f, //ambiental y difuso
    -4.0f, 1.0f, -8.5f,
    -5.0f, 0.0f, 0.0f,
    1.0f, 0.0f, 0.0f, //valores de la ecuacion cuadratica
    15.0f); //Angulo de apertura del cono del spotlight
spotLightCount++;
//se crean mas luces puntuales y spotlight
```

Y para moverla con el carro, utilice las siguientes líneas de código:

```
glm::vec3 luzcarro = glm::vec3(mainWindow.getarticulacion3(), 1.0f, -8.5f);
glm::vec3 luzcarro2 = glm::vec3(-mainWindow.getarticulacion4(), 1.0f, -8.5f);
```

```
spotLights[2].SetPos(luzcarro2);
spotLights[2].SetPos(luzcarro);
```

De esta forma, al presionar las teclas con las que se mueve el carro, la luz también se mueve.

Y para el código del modelo del carro, tengo los siguientes bloques:

```
//-----*INICIA DIBUJO DEL CARRO*-----*
//CHASIS
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, 0.5f, -1.5f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::translate(model, glm::vec3(9.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getrotay()), glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::translate(model, glm::vec3(0.0, 0.0f, mainWindow.getarticulacion3()));
model = glm::translate(model, glm::vec3(0.0, 0.0f, -mainWindow.getarticulacion4()));
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
chasis.RenderModel();

spotLights[2].SetPos(luzcarro2);
spotLights[2].SetPos(luzcarro);

//COFRE
model = modelaux;
model = glm::translate(model, glm::vec3(0.0f, 1.55f, -3.5f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(-1.0f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion2()), glm::vec3(-1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
cofre.RenderModel();

//LLANTA 1
model = modelaux;
model = glm::translate(model, glm::vec3(-2.0f, 0.0f, -3.3f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
llanta1.RenderModel();

//LLANTA 2
model = modelaux;
model = glm::translate(model, glm::vec3(-2.0f, 0.0f, +3.1f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
llanta2.RenderModel();

// LLANTA 3
model = modelaux;
model = glm::translate(model, glm::vec3(2.0f, 0.0f, -3.3f));

// LLANTA 4
model = modelaux;
model = glm::translate(model, glm::vec3(2.0f, 0.0f, +3.1f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
llanta4.RenderModel();
```

- **Problemas Presentados**

No conseguí mover la luz cuando el carro avanzaba, solo cuando retrocedía.

- **Conclusión:**

En este ejercicio, aprendí a texturizar elementos en 3DS Max mediante proyecciones, como la de un cubo alrededor de un elemento, y a crear y posicionar luces en un mundo virtual.