



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: Aguilar Pérez José Ramón

N° de Cuenta: 317515048

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 01/abril/2025

CALIFICACIÓN: _____

EJERCICIOS DE SESIÓN:

1. Actividades realizadas.

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

Para el ejercicio, se importaron los modelos y texturas (dentro de la carpeta *Textures* del proyecto) del Porsche trabajado en prácticas anteriores.

```
Model Porsche_M;
Model Cofre_M;
Model LlantaTrasDer_M;
Model LlantaTrasIzq_M;
Model LlantaDelDer_M;
Model LlantaDelIzq_M;

Porsche_M = Model();
Porsche_M.LoadModel("Models/porsche.obj");
Cofre_M = Model();
Cofre_M.LoadModel("Models/cofre.obj");
LlantaTrasDer_M = Model();
LlantaTrasDer_M.LoadModel("Models/llantatrasder.obj");
LlantaTrasIzq_M = Model();
LlantaTrasIzq_M.LoadModel("Models/llantatrasizq.obj");
LlantaDelDer_M = Model();
LlantaDelDer_M.LoadModel("Models/llantadelder.obj");
LlantaDelIzq_M = Model();
LlantaDelIzq_M.LoadModel("Models/llantadelizq.obj");
```

Una vez que los modelos han sido declarados, se procede a la construcción del coche por medio de jerarquía. Además, se agregan las funciones de rotación para el cofre, las llantas y de traslación para el movimiento del coche.

```
***** Instancia del coche *****//
//Porsche
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex(), 0.1f, 1.0f)); // Adelante -> U ; Atras -> Y
spotlights[2].SetPos(glm::vec3(0.0f + mainWindow.getmuevex(), 0.1f, 1.0f)); //Luz del faro de color azul
model = glm::scale(model, glm::vec3(0.25f, 0.25f, 0.25f));
model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Porsche_M.RenderModel();

//Cofre
model = modelaux;
model = translate(model, glm::vec3(-0.28f, 3.22f, -8.8f));
model = glm::rotate(model, glm::radians(mainWindow.getcofre()), glm::vec3(0.1f, 0.0f, 0.0f)); // Abrir -> F ; Cerrar -> G
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Cofre_M.RenderModel(); //Cofre del porsche

//Llanta delantera derecha
model = modelaux;
model = translate(model, glm::vec3(6.8f, -1.4f, -10.8f));
model = glm::rotate(model, glm::radians(mainWindow.getrueda()), glm::vec3(0.1f, 0.0f, 0.0f)); // Acelera -> H ; Reversa -> J
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaDelDer_M.RenderModel();

//Llanta delantera izquierda
model = modelaux;
model = translate(model, glm::vec3(-6.8f, -1.4f, -10.8f));
model = glm::rotate(model, glm::radians(mainWindow.getrueda()), glm::vec3(0.1f, 0.0f, 0.0f)); // Acelera -> H ; Reversa -> J
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaDelIzq_M.RenderModel();

//Llanta trasera izquierda
model = modelaux;
model = translate(model, glm::vec3(-6.6f, -1.4f, 11.0f));
model = glm::rotate(model, glm::radians(mainWindow.getrueda()), glm::vec3(0.1f, 0.0f, 0.0f)); // Acelera -> H ; Reversa -> J
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaTrasIzq_M.RenderModel();

//Llanta trasera derecha
model = modelaux;
model = translate(model, glm::vec3(6.6f, -1.4f, 11.0f));
model = glm::rotate(model, glm::radians(mainWindow.getrueda()), glm::vec3(0.1f, 0.0f, 0.0f)); // Acelera -> H ; Reversa -> J
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaTrasDer_M.RenderModel();
```

En el archivo *Window.cpp*, se agregaron las condicionales para que el cofre solo abriera hasta 40° y se detenga al llegar a 0°. También se encuentra la declaración de teclas para los movimientos del coche:

- Cofre: F -> abre, G -> Cierra.
- Ruedas: H -> Acelera, J -> Reversa.
- Porsche: U -> Avanza, Y -> Retrocede.

```

//Movimiento cofre
if (key == GLFW_KEY_F)
{
    if (theWindow->cofre > 40) {
    }
    else {
        theWindow->cofre += 1.0;
    }
}

//Movimineto llantas
if (key == GLFW_KEY_H)
{
    theWindow->rueda -= 10.0;
}
if (key == GLFW_KEY_J)
{
    theWindow->rueda += 10.0;
}

if (key == GLFW_KEY_Y)
{
    theWindow->muevex += 1.0;
}
if (key == GLFW_KEY_U)
{
    theWindow->muevex -= 1.0;
}

if (key == GLFW_KEY_G)
{
    if (theWindow->cofre < 0.0) {
    }
    else {
        theWindow->cofre -= 1.0;
    }
}

```

Para la luz azul que simula los faros del coche, se agregó un nuevo spotlight, el cual se apunta al frente del coche, en este caso apunta al eje X negativo, además de que se especifica el color azul para esta nueva luz.

```

//Faro azul del coche
spotLights[2] = SpotLight(0.0f, 0.0f, 1.0f,
    1.0f, 2.0f, //primer termino -> intensidad de iluminacion
    5.0f, 10.0f, 0.0f, //Posicion foco invisible
    -5.0f, 0.0f, 0.0f, //Direccion donde apunta
    1.0f, 0.0f, 0.0f, //Ecuacion de segundo grado
    15.0f); //Angulo del cono
spotLightCount++;

```

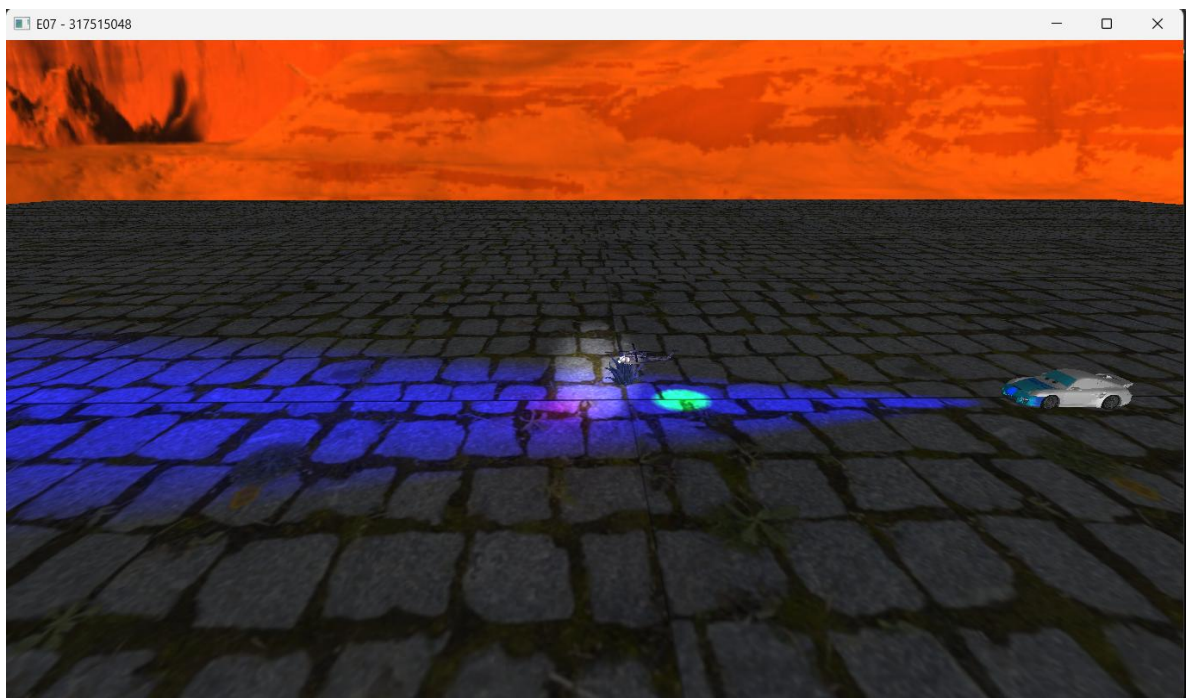
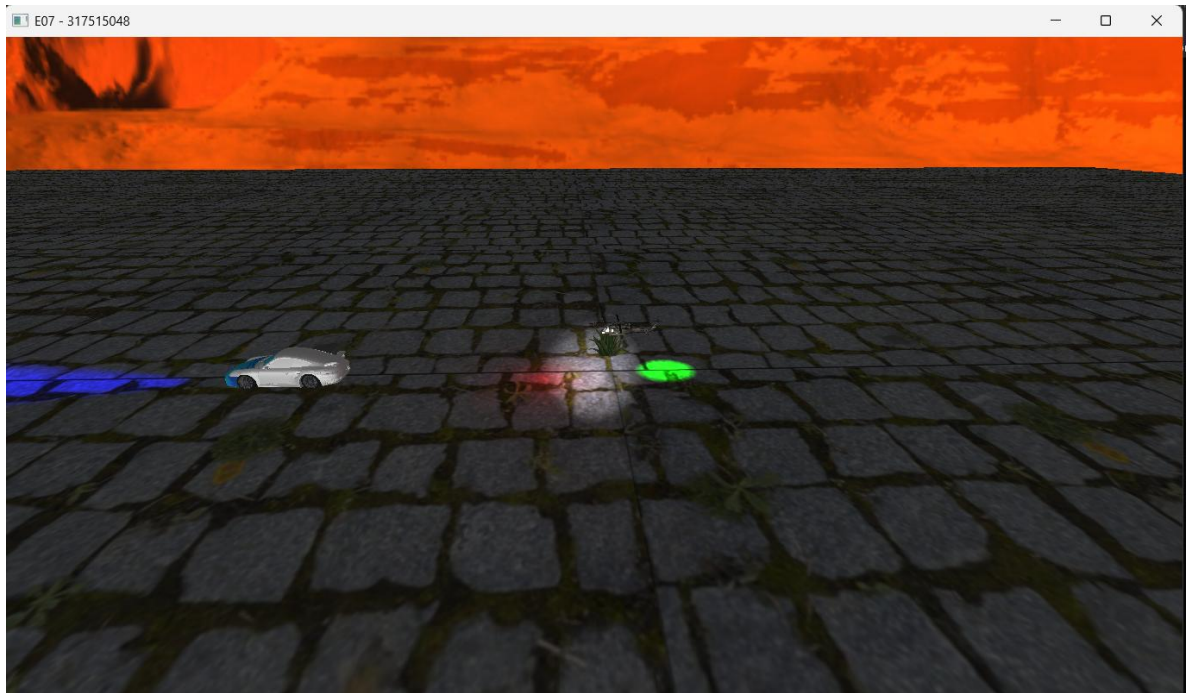
Finalmente, se agrega la luz azul al frente del coche con ayuda de la función *SetPos()*. Es importante añadir dentro de esta función la traslación dinámica del coche para que la luz haga los mismos movimientos que el coche.

```

//Porsche
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex(), 0.1f, 1.0f)); // Adelante -> U ; Atras -> Y
spotLights[2].SetPos(glm::vec3(0.0f + mainWindow.getmuevex(), 0.1f, 1.0f)); //Luz del faro de color azul
model = glm::scale(model, glm::vec3(0.25f, 0.25f, 0.25f));

```







2. Problemas presentados.

No se presentaron problemas a la hora de realizar el ejercicio.

3. Conclusión

a. Los ejercicios de la clase: Complejidad, explicación

Los ejercicios solicitados tuvieron una complejidad no tan elevada, ya que gracias a la buena explicación de cómo funcionan los distintos tipos de luces del entorno, así como los parámetros que manejan estos elementos (específicamente hablando de la luz de spotlight) se pudo hacer los ejercicios solicitados.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias.

La explicación general de los distintos tipos de luces, así como los parámetros que se deben modificar para lograr los resultados deseados fue clara y precisa. También fue de gran ayuda la explicación sobre la función `SetPos()` para posicionar las luces en una posición específica.