



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:** Carlos Alberto Arroyo Ramírez

**N° de Cuenta:** 320185865

**GRUPO DE LABORATORIO:** 03

**GRUPO DE TEORÍA:** 04

**SEMESTRE 2026-1**

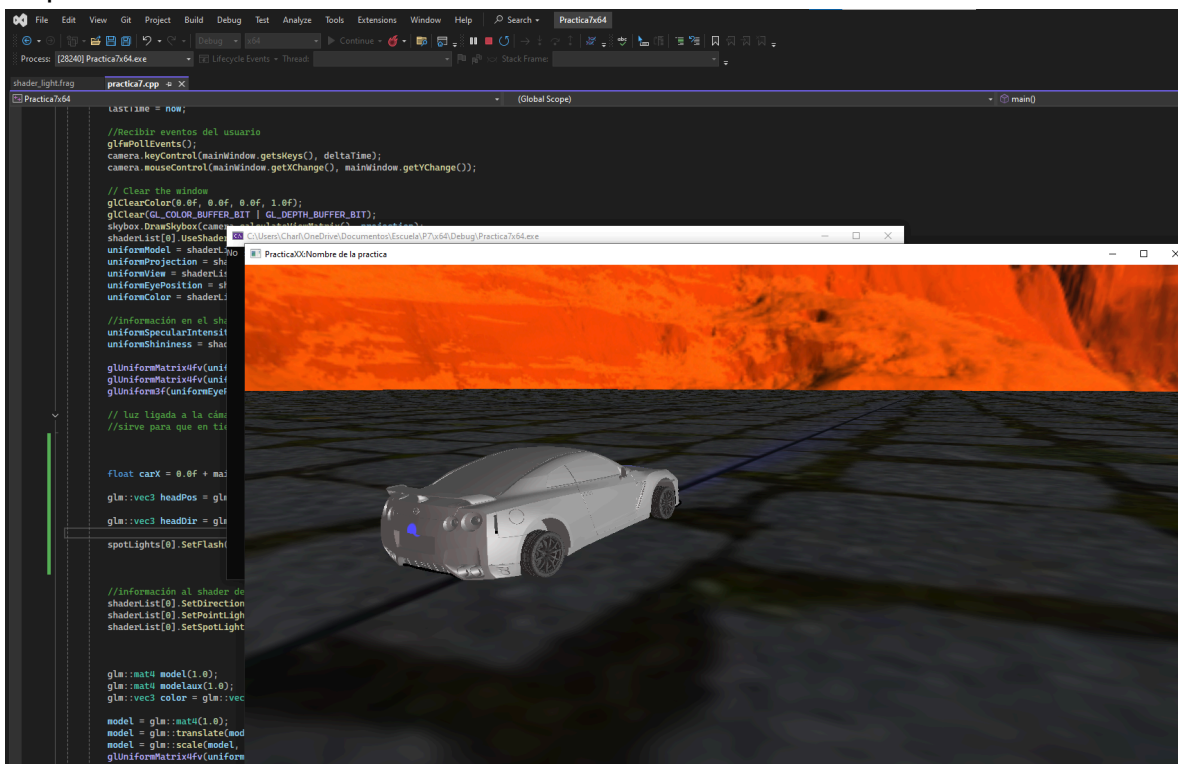
**FECHA DE ENTREGA LÍMITE:** 14/10/2025

**CALIFICACIÓN:** \_\_\_\_\_

## EJERCICIOS DE SESIÓN:

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

En esta práctica importamos nuestro modelo de coche junto con las ruedas, respetamos la jerarquía para que al momento de moverse el cuerpo lo hagan las ruedas igual, solo fue cuestión de importar el modelo 3D y acomodar las ruedas respecto al carro:



Así quedaron las coordenadas modificadas:

```

//Instancia del coche
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex(), -0.8f, 0.0f));
modelaux = model;
model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Kitt_M.RenderModel();

//Llanta delantera izquierda
model = modelaux;
model = glm::translate(model, glm::vec3(-1.7f, 0.2f, 1.0f));
model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
color = glm::vec3(0.5f, 0.5f, 0.5f); //llanta con color gris
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Llanta_M.RenderModel();

//Llanta trasera izquierda
model = modelaux;
model = glm::translate(model, glm::vec3(1.8f, 0.2f, 1.0f));
model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Llanta_M.RenderModel();

//Llanta delantera derecha
model = modelaux;
model = glm::translate(model, glm::vec3(-1.7f, 0.2f, -1.0f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Llanta_M.RenderModel();

//Llanta trasera derecha
model = modelaux;
model = glm::translate(model, glm::vec3(1.8f, 0.2f, -1.0f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
model = glm::scale(model, glm::vec3(0.4f, 0.4f, 0.4f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Llanta_M.RenderModel();

```

Para hacer la actividad de los faros, para empezar cambie el color de la linterna de blanco a azul:

```

//linterna
spotLights[0] = SpotLight(0.0f, 0.0f, 1.0f,
    0.0f, 2.0f,
    0.0f, 0.0f, 0.0f,
    0.0f, -1.0f, 0.0f,
    1.0f, 0.0f, 0.0f,
    5.0f);
spotLightCount++;

```

Finalmente este es el código que utilice para juntar el faro al carro:

```
float carX = 0.0f + mainWindow.getmuevex();  
glm::vec3 headPos = glm::vec3(carX - 4.0f, 0.0f, 0.0f);  
glm::vec3 headDir = glm::normalize(glm::vec3(1.0f, -0.05f, 0.0f));  
spotLights[0].SetFlash(headPos, headDir);
```

2. Problemas presentados. Listar si surgieron problemas a la hora de ejecutar el código

No tuve problemas para realizar este ejercicio.

3. Conclusión:
  - a. Los ejercicios de la clase: Complejidad, explicación
  - b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias.

En esta práctica aprendimos los diferentes tipos de luces (Direccional, puntual y spotlight) y sus respectivos parámetros, aprendí que hay luces 'Padre' por medio de superclases que engloban los parámetros de las demás luces, y que existe la iluminación local y global.