



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e  
INTERACCIÓN HUMANO COMPUTADORA



## **REPORTE DE PRÁCTICA N° 05**

**NOMBRE COMPLETO:** Gabriel Patricio Balam Flores

**N° de Cuenta:** 320280324

**GRUPO DE LABORATORIO:** 03

**GRUPO DE TEORÍA:** 04

**SEMESTRE** 2026-1

**FECHA DE ENTREGA LÍMITE:** 28/sept

**CALIFICACIÓN:** \_\_\_\_\_

## REPORTE DE PRÁCTICA

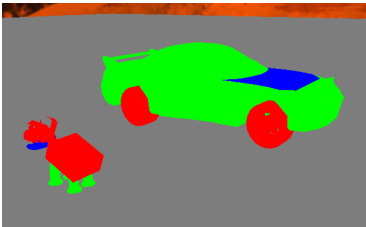


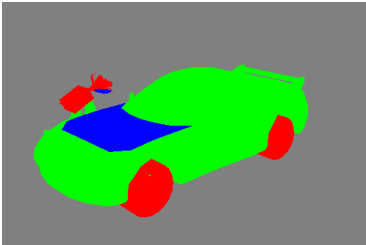
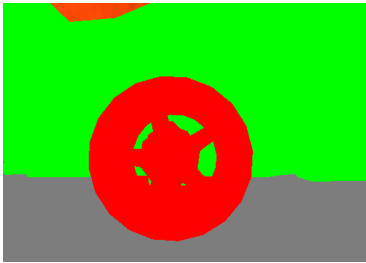
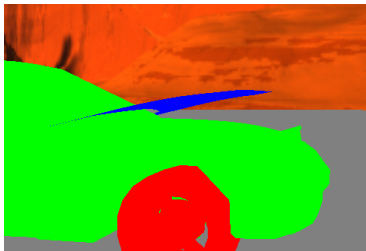
1.- Ejecución de los ejercicios que se dejaron, comentar cada uno y capturas de pantalla de bloques de código generados y de ejecución del programa.

### Ejercicio

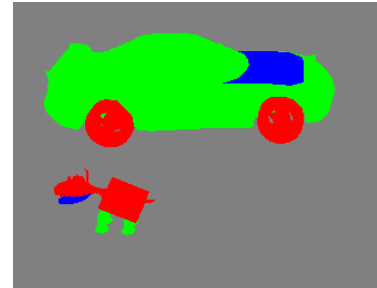
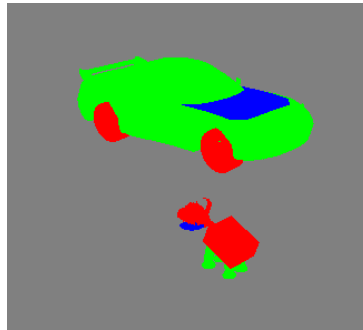
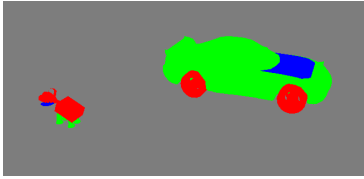
En este ejercicio teníamos que usar un modelo descargado de internet para integrarlo en OpenGL y agregarle ciertos movimientos.

El primer paso fue utilizar las herramientas de Blender para separar las partes de interés, en este caso, las llantas y el cofre, y volverlas independientes. Este proceso resultó muy sencillo gracias al shortcut que aprendimos en el laboratorio, ya que al presionar una sola tecla (L) se selecciona por completo la parte que queremos separar, lo cual ahorra mucho tiempo.

Después de exportar los tres modelos independientes, el siguiente paso fue importarlos en OpenGL utilizando la librería Assimp. Finalmente, se organizaron jerárquicamente y se posicionaron, procesos que ya habíamos trabajado en prácticas anteriores, por lo que no fue difícil.

Coche	Llantas [ V, B ]	Cofre [ C ]
		
		

## Movimiento [ N, M ]



## Importar modelos

```
Model Carro_cuerpo;
Model Carro_llanta_1;
Model Carro_llanta_2;
Model Carro_llanta_3;
Model Carro_llanta_4;
Model Carro_cofre;
```

```
Carro_cuerpo = Model();
Carro_cuerpo.LoadModel("Models/Carro_cuerpo.obj");

Carro_llanta_1 = Model();
Carro_llanta_1.LoadModel("Models/Carro_llanta_der.obj");

Carro_llanta_2 = Model();
Carro_llanta_2.LoadModel("Models/Carro_llanta_der.obj");

Carro_llanta_3 = Model();
Carro_llanta_3.LoadModel("Models/Carro_llanta_der.obj");

Carro_llanta_4 = Model();
Carro_llanta_4.LoadModel("Models/Carro_llanta_der.obj");

Carro_cofre = Model();
Carro_cofre.LoadModel("Models/Carro_cofre.obj");
```

## Organizar modelos

```
model = auxmodel;
model = glm::translate(model, glm::vec3(0.0f, 2.3f, -20.0f));
model = glm::scale(model, glm::vec3(5.0f, 5.0f, 5.0f));

model = glm::translate(model, glm::vec3(mainWindow.getarticulacion8(), 0.0f, 0.0f));

auxmodel = model;

// Para el cuerpo del carro
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(0.0f, 1.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
Carro_cuerpo.RenderModel();

// Para el cofre
model = glm::translate(model, glm::vec3(0.76f, 0.38f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion6()), glm::vec3(0.0f, 0.0f, 1.0f));

glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(0.0f, 0.0f, 1.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
Carro_cofre.RenderModel();
```

```

// Para la llanta izquierda delantera
model = auxmodel;
model = glm::translate(model, glm::vec3(1.8f, -0.4f, 0.9f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));

model = glm::rotate(model, glm::radians(mainWindow.getarticulacion7()), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion8()), glm::vec3(0.0f, 0.0f, 1.0f));

glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
Carro_llanta_1.RenderModel();

// Para la llanta izquierda trasera
model = auxmodel;
model = glm::translate(model, glm::vec3(-1.5f, -0.4f, 0.9f));
model = glm::rotate(model, glm::radians(180.0f), glm::vec3(0.0f, 1.0f, 0.0f));

model = glm::rotate(model, glm::radians(mainWindow.getarticulacion7()), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion8()), glm::vec3(0.0f, 0.0f, 1.0f));

glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
color = glm::vec3(1.0f, 0.0f, 0.0f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
Carro_llanta_1.RenderModel();

```

## Aumentos en las articulaciones

```

if (key == GLFW_KEY_C)
{
    theWindow->articulacion6 += 2.5 * theWindow->dir6;

    if (theWindow->articulacion6 == 60)
        theWindow->dir6 *= -1.0f;
    else if (theWindow->articulacion6 == 0)
        theWindow->dir6 *= -1.0f;
}

if (key == GLFW_KEY_V || key == GLFW_KEY_N )
{
    theWindow->articulacion7 += 10.0;
}

if (key == GLFW_KEY_B || key == GLFW_KEY_M)
{
    theWindow->articulacion7 -= 10.0;
}

if (key == GLFW_KEY_N)
{
    theWindow->articulacion8 += 0.1;
}

if (key == GLFW_KEY_M)
{
    theWindow->articulacion8 -= 0.1;
}

```

2.- Liste los problemas que tuvo a la hora de hacer estos ejercicios y si los resolvió explicar cómo fue, en caso de error adjuntar captura de pantalla.

## Contratiempos

Esta práctica se sintió mucho más sencilla, ya que las habilidades necesarias para realizarla se han ido desarrollando en las prácticas anteriores. Además, en mi caso, el uso constante de Blender en la clase teórica facilitó aún más el proceso.

3.- Conclusión:

- a. Los ejercicios del reporte: Complejidad, Explicación.
  - b. Comentarios generales.
  - c. Conclusión
1. Bibliografía en formato APA

## Conclusión

En esta práctica aprendí a importar modelos externos a OpenGL utilizando la biblioteca Assimp en diferentes formatos. También trabajé con modelos descargados de internet y los acondicioné para poder utilizarlos y agregarles movimiento.

## Bibliografía

- Página principal. (26 de septiembre de 2018). *Wiki de OpenGL* . Recuperado el 29 de agosto de 2025 de [http://www.khronos.org/opengl/wiki\\_opengl/index.php?title=Main\\_Page&oldid=14430](http://www.khronos.org/opengl/wiki_opengl/index.php?title=Main_Page&oldid=14430) .