



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: Aguilar Pérez José Ramón

N° de Cuenta: 317515048

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 22/marzo/2025

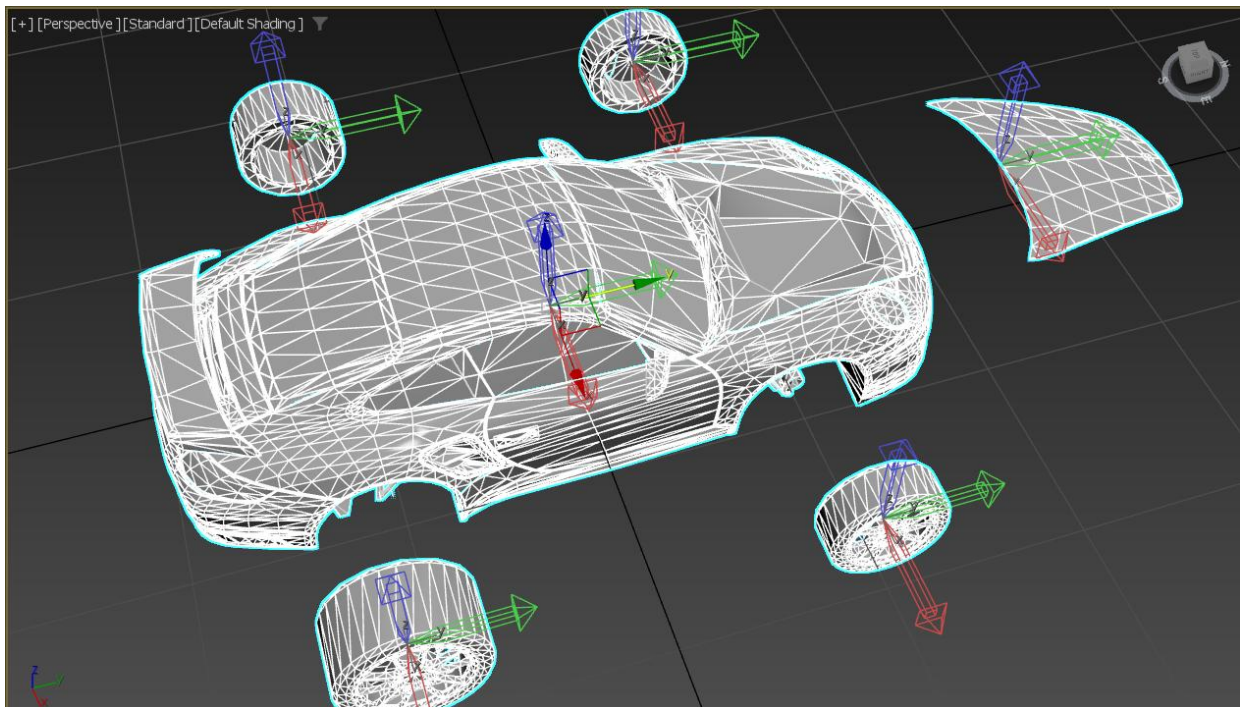
CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejercicios realizados.

- **Importar su modelo de coche propio dentro del escenario a una escala adecuada.**
 - **Importar sus 4 llantas y acomodarlas jerárquicamente, agregar el mismo valor de rotación a las llantas para que al presionar puedan rotar hacia adelante y hacia atrás.**
 - **Importar el cofre del coche, acomodarlo jerárquicamente y agregar la rotación para poder abrir y cerrar.**
 - **Agregar traslación con teclado para que pueda avanzar y retroceder de forma independiente.**

Para la realización de esta práctica, se abrió el modelo del Porsche en 3dsMax para modificar las partes necesarias. Se separaron cada una de las llantas, así como el cofre del coche para acomodar su respectivo pivote que permitirá su movimiento en el programa principal. Al momento de exportar cada uno de los elementos, se colocaron en el origen los pivotes de cada uno de los modelos para evitar movimientos no deseados.



Dentro del programa principal, se fueron exportando los modelos de las partes del coche.

```
//Creando los modelos del coche
Camera camera;
Model Porsche_M;
Model Cofre_M;
Model LlantaTrasDer_M;
Model LlantaTrasIzq_M;
Model LlantaDelDer_M;
Model LlantaDelIzq_M;
```

```
//Importando los modelos del coche
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/llantadellder.obj");
LlantaDelIzq_M = Model();
LlantaDelIzq_M.LoadModel("Models/llantadelizq.obj");
```

Una vez que se instanciaron los modelos, se procedió a la creación del coche por medio de jerarquía. Primero se generó el coche, el cual será la base para los demás elementos. Para lograr el efecto de desplazamiento, se agregó una articulación dentro de la función de traslación, para que al presionar la tecla K el auto avance hacia adelante, y al presionar la tecla L avance hacia atrás.

```
//Porsche 911 GT2
model = glm::mat4(1.0);
color = glm::vec3(0.8f, 0.8f, 0.8f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
model = translate(model, glm::vec3(0.0f, 2.4f, mainWindow.getarticulacion3())); //Mueve el coche: adelante -> K, atrás -> L
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Porsche_M.RenderModel(); //Modelo base del porsche
```

```
//Movimiento del coche
if (key == GLFW_KEY_K)
{
    theWindow->articulacion3 -= 10.0;
}
if (key == GLFW_KEY_L)
{
    theWindow->articulacion3 += 10.0;
}
```

Una vez generada la base, y con ayuda de una matriz auxiliar, se fueron generando las demás partes por medio de jerarquía. Por un lado, se tiene el cofre del coche, el cual se abre al presionar la tecla F y se cierra al presionar G. Para lograr este movimiento se aplicó una condicional en la articulación, para que al llegar a 40°, deje de moverse.

```
//Cofre
model = modelaux;
color = glm::vec3(0.7f, 0.7f, 0.7f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
model = translate(model, glm::vec3(0.0f, 3.3f, -8.8f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Cofre_M.RenderModel(); //Cofre del porsche
```

```

//Movimiento del cofre
if (key == GLFW_KEY_F)
{
    if (theWindow->articulacion1 > 40.0) {
        //Se deja de mover a llegar a 40°
    }
    else {
        theWindow->articulacion1 += 10.0;
    }
}
if (key == GLFW_KEY_G)
{
    if (theWindow->articulacion1 < 10.0) {
        //Se deja de mover a llegar a 10°
    }
    else {
        theWindow->articulacion1 -= 10.0;
    }
}

```

Para las llantas se hizo algo similar, por medio de la matriz auxiliar se fueron instanciando cada una de ellas, y con ayuda de la traslación se fueron posicionando en su respectiva posición. Además, se agregó la rotación para que al presionar la tecla H las llantas se muevan al frente, y al presionar J se muevan en reversa.

```

//----- Llantas -----*/
//Llanta delantera derecha
model = modelaux;
color = glm::vec3(0.1f, 0.1f, 0.1f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
model = translate(model, glm::vec3(6.8f, -1.4f, -11.0f));
modelaux = model;
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));
LlantaDelDer_M.RenderModel();

//Llanta delantera izquierda
model = modelaux;
color = glm::vec3(0.1f, 0.1f, 0.1f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
model = translate(model, glm::vec3(-13.6f, 0.0f, 0.0f));
modelaux = model;
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));
LlantaDelIzq_M.RenderModel();

//Llanta trasera izquierda
model = modelaux;
color = glm::vec3(0.1f, 0.1f, 0.1f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
model = translate(model, glm::vec3(0.3f, 0.0f, 22.0f));
modelaux = model;
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));
LlantaTrasIzq_M.RenderModel();

//Llanta trasera derecha
model = modelaux;
color = glm::vec3(0.1f, 0.1f, 0.1f);
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
model = translate(model, glm::vec3(13.3f, 0.0f, 0.0f));
modelaux = model;
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));
LlantaTrasDer_M.RenderModel();

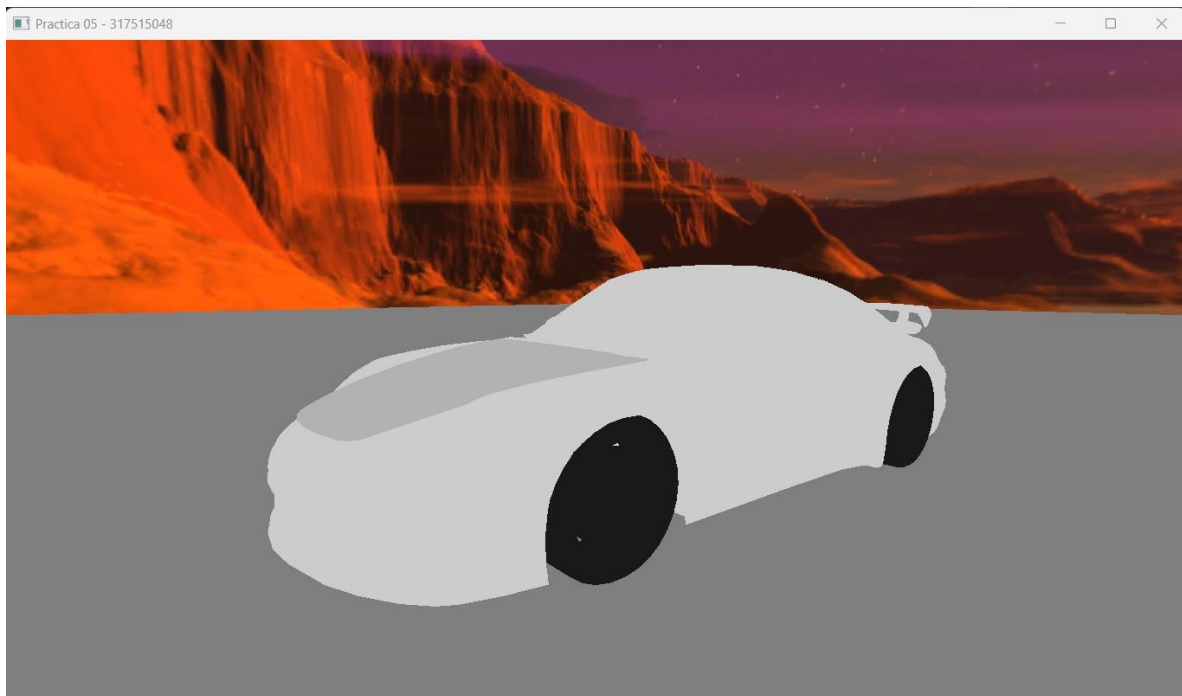
```

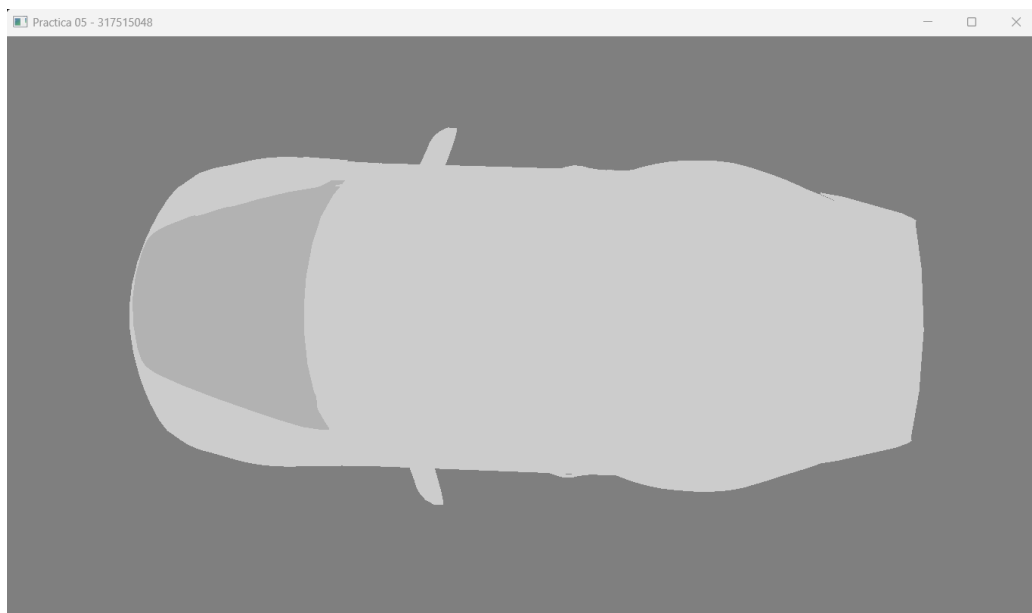
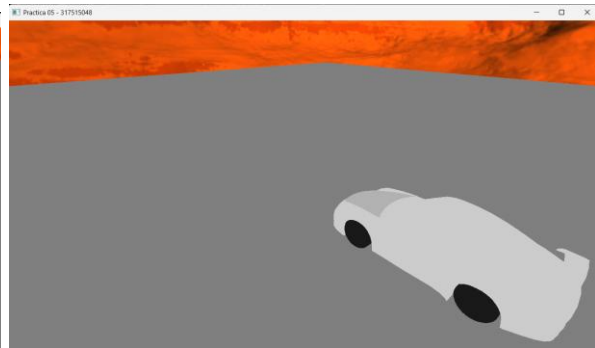
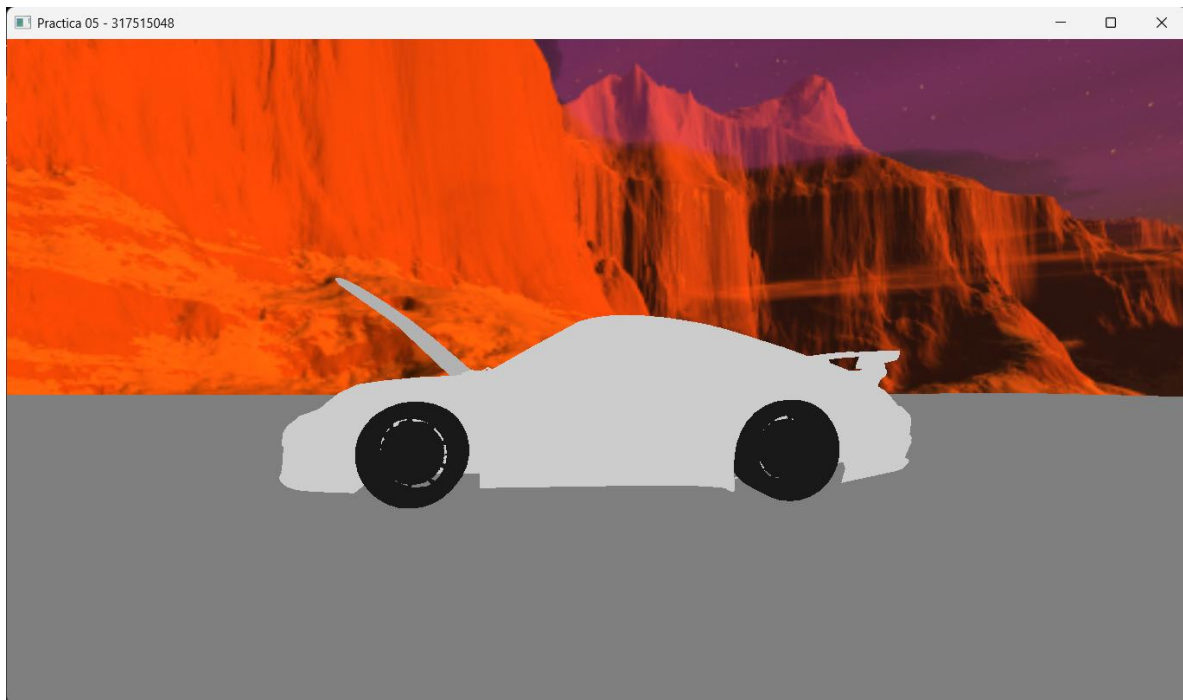
```

//Movimiento de ruedas
if (key == GLFW_KEY_H)
{
    theWindow->articulacion2 -= 10.0;
}
if (key == GLFW_KEY_J)
{
    theWindow->articulacion2 += 10.0;
}

```

Con todo lo anterior, se logró los movimientos solicitados para el coche, las llantas y el cofre. Cabe añadir que las articulaciones se modificaron del archivo *Window.cpp*.





2.- Problemas presentados

No se presentaron problemas significativos a la hora de realizar los ejercicios.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

La elaboración de esta práctica fue muy interesante de realizar, ya que permitió comprender de mejor manera como se exportan los modelos de objetos a OpenGL. Gracias a la explicación de como importar los modelos, así de la colocación de los pivotes, se llevo a cabo la práctica sin contratiempos.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica

La explicación de la colocación de los pivotes de los diferentes elementos, así como la explicación del ambiente de 3dsMax fue clara y precisa.

c. Conclusión

Esta práctica me permitió comprender mejor cómo importar modelos a OpenGL, además de la importancia de importarlos con la jerarquía de pivotes correcta, así como la colocación del modelo desde el origen, ya que, si no se hace estos dos pasos importantes, las traslaciones y rotaciones que se deben aplicar en el programa principal son muy diferentes a las esperadas. Trabajar en 3dsMax fue una nueva experiencia, pero gracias a esta práctica aprendí los puntos más importantes para que la manipulación de modelos sea más eficiente.

1. Bibliografía en formato APA

- 3ds Max Quick Start Guide. (s/f). Autodesk.com. Recuperado el 20 de marzo de 2025, de <https://www.autodesk.com/learn/ondemand/curated/3ds-max-quick-start-guide>
- Tutorial 3: Matrices. (s/f). Opengl-tutorial.org. Recuperado el 18 de marzo de 2025, de <http://www.opengl-tutorial.org/es/beginners-tutorials/tutorial-3-matrices/>