



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: Mino Guzmán Yara Amairani

N° de Cuenta: 422017028

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

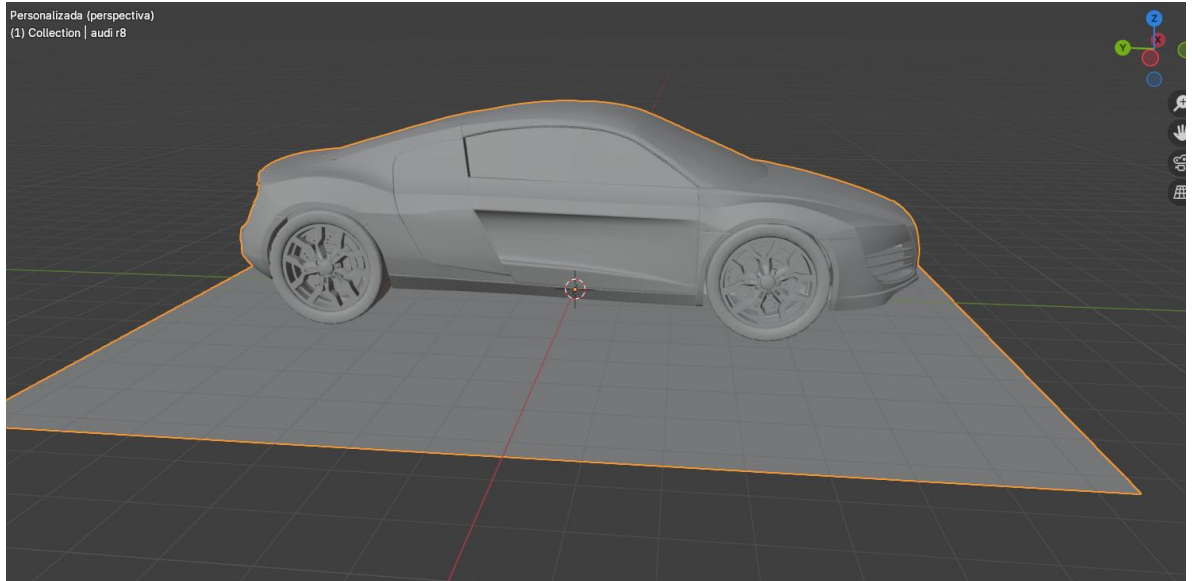
FECHA DE ENTREGA LÍMITE: 22 marzo 2025

CALIFICACIÓN: _____

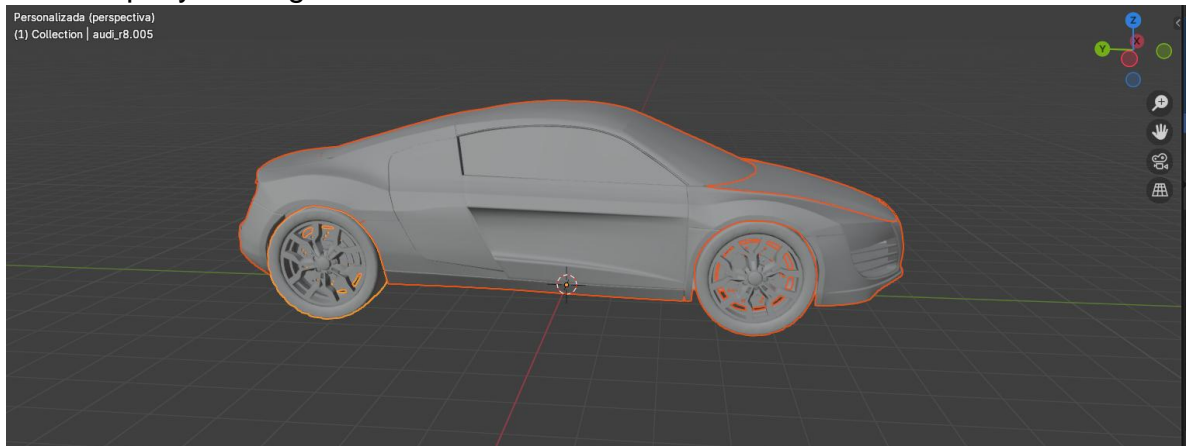
REPORTE DE PRÁCTICA:

1.- Actividades:

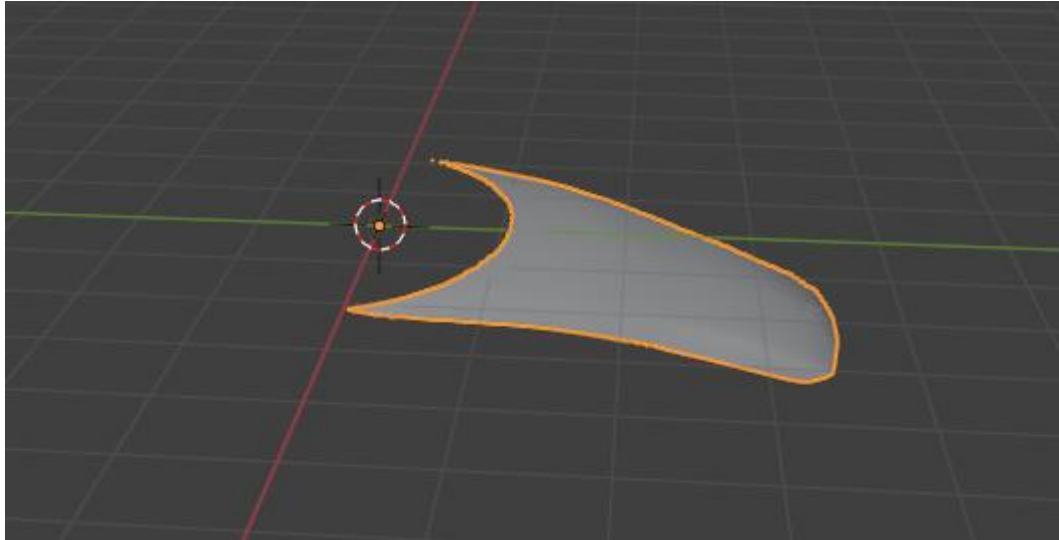
- Importar su modelo de coche propio dentro del escenario a una escala adecuada.



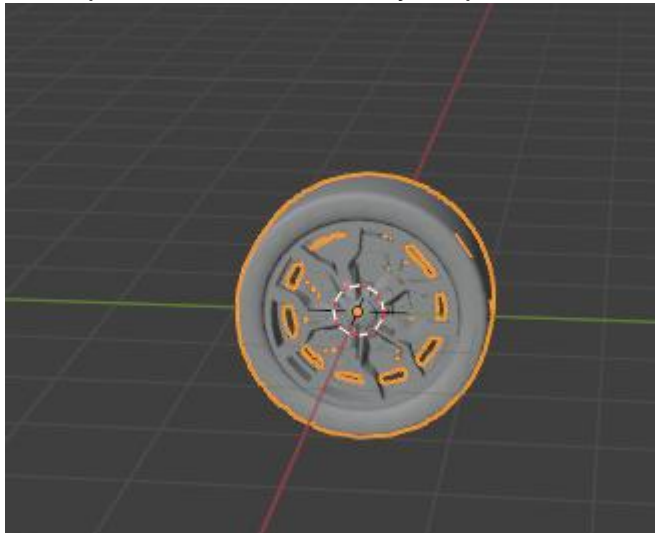
El auto que yo escogí fue un AudiR8.



Lo primero que se hizo, fue quitar el piso con el que venía.



Se separó el cofre del auto y se puso en el origen.



Se separaron las llantas, una por una y también se pusieron en el origen.

- 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.

```
//Auto
color = glm::vec3(0.0f, 0.0f, 0.5f);
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, -1.7f, 0.0f));
model = glm::translate(model, glm::vec3(0.0f, 0.0f, mainWindow.getmueveCarro()));
modelaux = model; //Se guarda la transformación del cuerpo
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Auto_M.RenderModel();
```

Lo primero que se importó, fue el auto, sin llantas y sin cofre.

```
GLfloat getrotaLlantas() { return rotaLlantas; }
```

En window.h se declaró getrotaLlantas para realizar la rotación de las llantas y que fuera la misma para las 4.

```
//Rotación para las llantas
if (key == GLFW_KEY_B)
{
    theWindow->rotaLlantas += 10.0;
}
if (key == GLFW_KEY_C)
{
    theWindow->rotaLlantas -= 10.0;
}
```

En window.cpp, se aplicó la misma rotación para las 4 llantas para que se movieran hacia adelante y hacia atrás.

```
//LLANTA DELANTERA DERECHA
model = modelaux;
color = glm::vec3(0.0f, 0.0f, 0.0f); //Llanta de color negro
model = glm::translate(model, glm::vec3(-2.5f, 1.0f, 3.6f));
model = glm::rotate(model, glm::radians(mainWindow.getrotaLlantas()), glm::vec3(1.0f, 0.0f, 0.0f));
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaDD_M.RenderModel(); //Muestra Llanta

//LLANTA TRASERA DERECHA
model = modelaux;
color = glm::vec3(0.0f, 0.0f, 0.0f); //Llanta de color negro
model = glm::translate(model, glm::vec3(-2.39f, 1.0f, -3.9f));
model = glm::rotate(model, glm::radians(mainWindow.getrotaLlantas()), glm::vec3(1.0f, 0.0f, 0.0f));
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaTD_M.RenderModel(); //Muestra Llanta

//LLANTA DELANTERA IZQUIERDA
model = modelaux;
color = glm::vec3(0.0f, 0.0f, 0.0f); //Llanta de color negro
model = glm::translate(model, glm::vec3(2.2f, 1.0f, 3.6f));
model = glm::rotate(model, glm::radians(mainWindow.getrotaLlantas()), glm::vec3(1.0f, 0.0f, 0.0f));
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaDD_M.RenderModel(); //Muestra Llanta

//LLANTA TRASERA IZQUIERDA
model = modelaux;
color = glm::vec3(0.0f, 0.0f, 0.0f); //Llanta de color negro
model = glm::translate(model, glm::vec3(2.35f, 1.0f, -3.9f));
model = glm::rotate(model, glm::radians(mainWindow.getrotaLlantas()), glm::vec3(1.0f, 0.0f, 0.0f));
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
LlantaDD_M.RenderModel(); //Muestra Llanta
```

Se importaron las llantas de manera jerárquica, su rotación se encuentra en el eje z.

- Importar el cofre del coche, acomodarlo jerárquicamente y agregar la rotación para poder abrir y cerrar.

```
GLfloat getrotaCofre() { return rotaCofre; }
```

En window.h, se declaró rotaCofre para la rotación del cofre.

```
//Rotación para el cofre
if (key == GLFW_KEY_E)
{
    if (theWindow->rotaCofre < -45)
    {
    }
    else
    {
        theWindow->rotaCofre -= 10.0;
    }
}

if (key == GLFW_KEY_F)
{
    if (theWindow->rotaCofre == 0.0)
    {
    }
    else
    {
        theWindow->rotaCofre += 10.0;
    }
}
```

En window.cpp se declaró la rotación del cofre, para simular como si estuviera abriendo y cerrando.

```
//COFRE
model = modelaux;
color = glm::vec3(1.0f, 0.0f, 0.0f); //Cofre de color verde oscuro
model = glm::translate(model, glm::vec3(0.01f, 2.5f, 2.62f));
model = glm::rotate(model, glm::radians(mainWindow.getrotaCofre()), glm::vec3(1.0f, 0.0f, 0.0f));
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Cofre_M.RenderModel(); //Muestra cofre
```

Se importó el cofre de manera jerárquica y se le agregó su rotación para que pudiera abrir y cerrar, está rotando sobre el eje x.

- Agregar traslación con teclado para que pueda avanzar y retroceder de forma independiente.

```
GLfloat getmueveCarro() { return mueveCarro; }
```

Se declaró mueveCarro en window.h, para poder realizar su traslación y que pareciera que estuviera avanzando o retrocediendo.

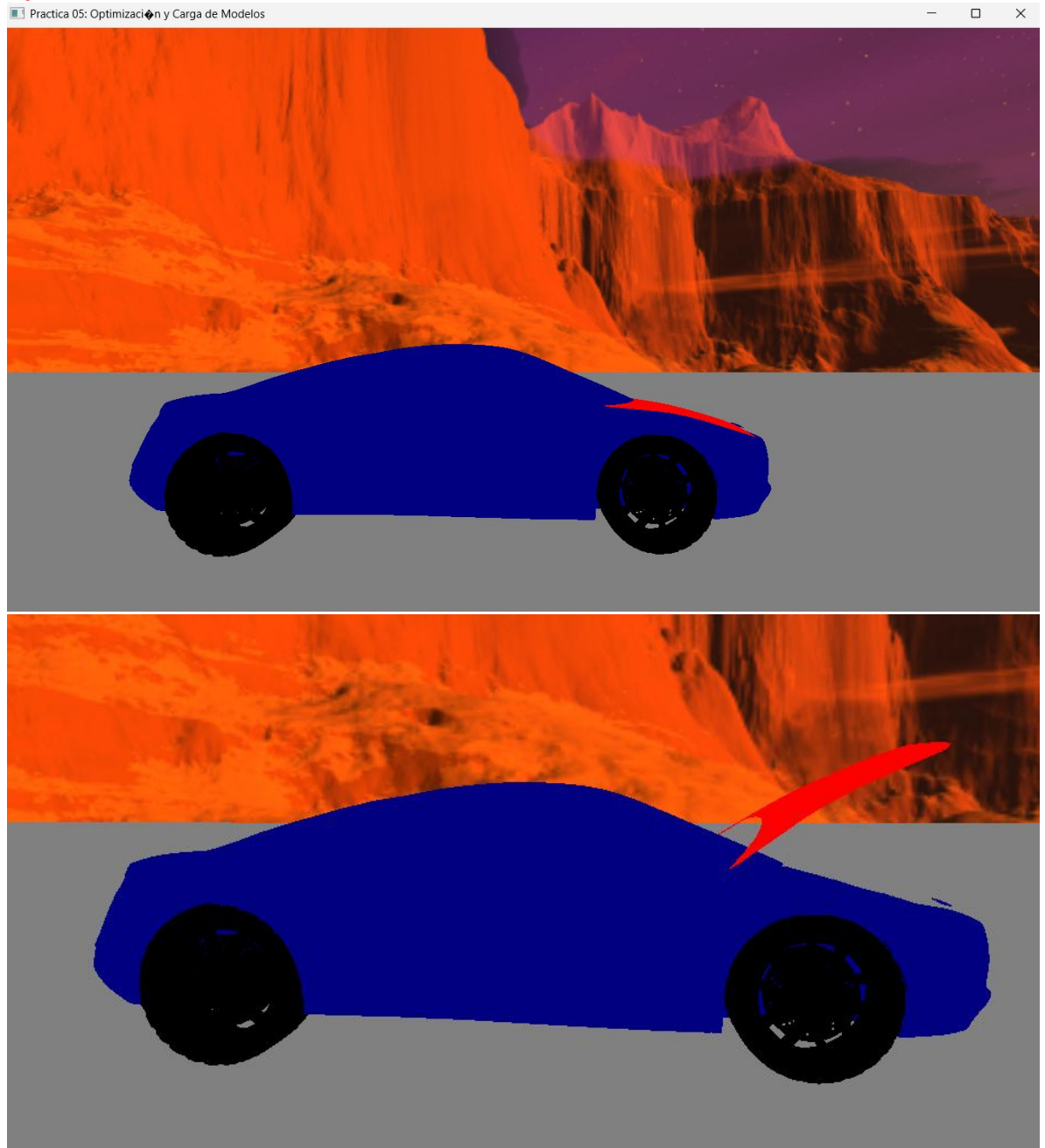
```
//Trasladar el carro
if (key == GLFW_KEY_G)
{
    theWindow->mueveCarro -= 0.5;
    theWindow->rotaLlantas += 10.0;
}
if (key == GLFW_KEY_H)
{
    theWindow->mueveCarro += 0.5;
    theWindow->rotaLlantas -= 10.0;
}
```

En window.cpp, se aplicó la traslación del auto, pero también se agregaron las rotaciones de las llantas, para que el auto pudiera avanzar y retroceder.

```
//Auto
color = glm::vec3(0.0f, 0.0f, 0.5f);
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, -1.7f, 0.0f));
model = glm::translate(model, glm::vec3(0.0f, 0.0f, mainWindow.getmueveCarro()));
modelaux = model; //Se guarda la transformación del cuerpo
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Auto_M.RenderModel();
```

Se agregó su traslación, su traslación se encuentra en el eje z.

Ejecución:



2.- Problemas a la hora de hacer las actividades solicitadas:

- Se me presentó el problema para poder realizar la traslación del auto, de ahí en fuera, las dudas se habían resuelto al momento de realizar el ejercicio del Goddart.

3.- Conclusión:

- a. Los ejercicios del reporte fueron los adecuados para poder comprender el uso de blender, importar los objetos e irlos agregando jerárquicamente; así como la rotación y la traslación de los objetos.
- b. Las explicaciones fueron claras y, gracias a eso, se realizó con éxito la práctica.
- c. En conclusión, la práctica me ayudó a explorar nuevas cosas de la computación gráfica, también me ayudó a entender mejor el uso de blender y a comprender mejor la jerarquización.

Bibliografía

- learnopengl.com. (s.f.). Transformations. LearnOpenGL.
<https://learnopengl.com/Getting-started/Transformations>
- OpenGL. (s.f.). OpenGL Programming Guide (The Red Book). Khronos Group.
<https://www.khronos.org/registry/OpenGL-Refpages/gl4/>
- TutorialsPoint. (s.f.). OpenGL - Transformations.
https://www.tutorialspoint.com/opengl/opengl_transformations.htm
- Scratchapixel. (s.f.). Understanding 3D Transformations.
<https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/geometry>
- GeeksforGeeks. (2021). Translation in Computer Graphics.
<https://www.geeksforgeeks.org/translation-in-computer-graphics/>