



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA
e INTERACCIÓN HUMANO COMPUTADORA



PRÁCTICA N° 5

NOMBRE COMPLETO: Vargas López Miguel Adán

N° de Cuenta: 421079522

GRUPO DE LABORATORIO: 3

GRUPO DE TEORÍA: 4

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 21/09/2024

- 1.- Importar su modelo de coche propio dentro del escenario a una escala adecuada.
- 2.- 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.
- 3.- Importar el cofre del coche, acomodarlo jerárquicamente y agregar la rotación para poder abrir y cerrar.
- 4.- Agregar traslación con teclado para que pueda avanzar y retroceder de forma independiente

Desarrollo

Primero importo del modelo del carro, su cuerpo y las llantas por separado.



Al tener los .obj, puedo declararlos e instanciarlos en el programa con OpenGL.
Habrá un objeto cuerpo y 4 instancias del objeto llanta.

```

carro_Cuerpo = Model();
carro_Cuerpo.LoadModel("Models/car_body.obj");

carro_llanta = Model();
carro_llanta.LoadModel("Models/llanta.obj");

carro_llanta2 = Model();
carro_llanta2.LoadModel("Models/llanta.obj");

carro_llanta3 = Model();
carro_llanta3.LoadModel("Models/llanta.obj");

carro_llanta4 = Model();
carro_llanta4.LoadModel("Models/llanta.obj");

```

Instanciando las llantas empezando por las frontales, para después, instanciar las traseras y asignándoles la rotación a través de `getarticulacion1`, donde la llanta rotará y a su vez el vehículo se moverá.

```

//lantas delanteras
model = glm::translate(model, glm::vec3(1.0f, 0.5f, 2.3f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(1.0f, 0.0f, 0.0f));
color = glm::vec3(0.0f, 0.0f, 0.0f);
modelaux = model;
glUniform3fv(uniformColor, 1, glm::value_ptr(color));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
carro_llanta.RenderModel();
model = modelaux;

model = glm::translate(model, glm::vec3(-1.8f, 0.0f, 0.0f));
model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(1.0f, 0.0f, 0.0f));
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
carro_llanta.RenderModel();
model = modelaux;

model = glm::translate(model, glm::vec3(0.0f, 0.0f, -3.5f));
//model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(1.0f, 0.0f, 0.0f));
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
carro_llanta.RenderModel();
model = modelaux;

model = glm::translate(model, glm::vec3(1.8f, 0.0f, 0.0f));
//model = glm::rotate(model, glm::radians(mainWindow.getarticulacion1()), glm::vec3(1.0f, 0.0f, 0.0f));
modelaux = model;
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
carro_llanta.RenderModel();
model = modelaux;

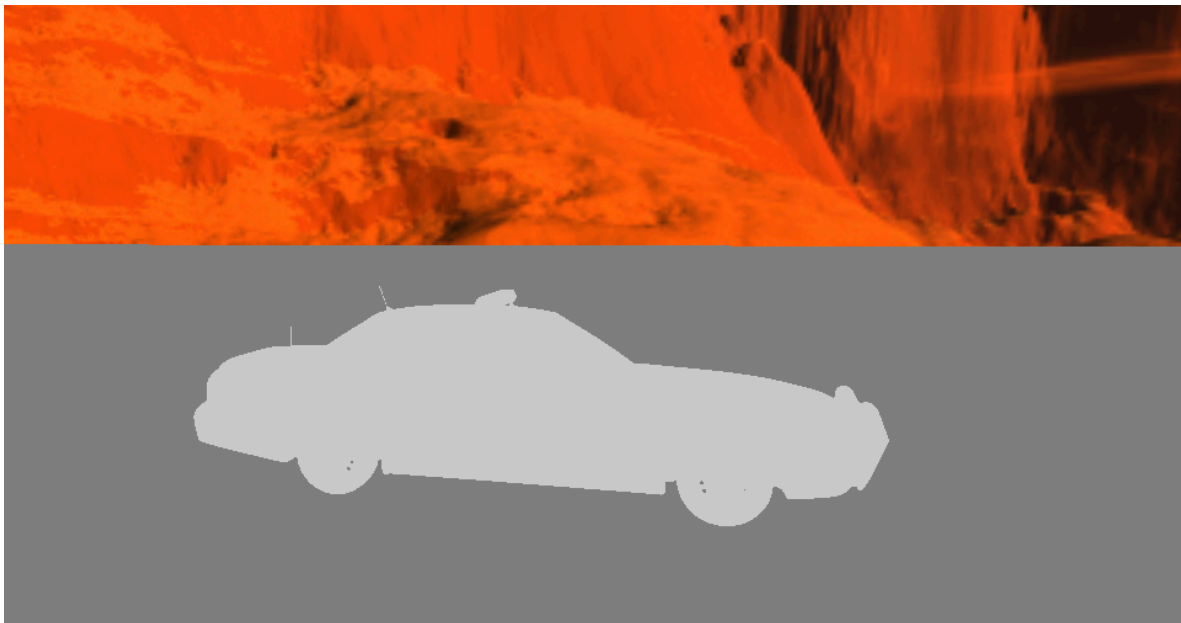
```

El auto se moverá hacia adelante con la tecla F y hacia atrás con la letra G.

```
if (key == GLFW_KEY_F)
{
    theWindow->articulacion1 += 10.0f;
    theWindow->posAutoZ += 0.5f;
}

if (key == GLFW_KEY_G)
{
    theWindow->articulacion1 -= 10.0f;
    theWindow->posAutoZ -= 0.5f;
}
```

Para tener en visualización las siguientes vistas del auto:

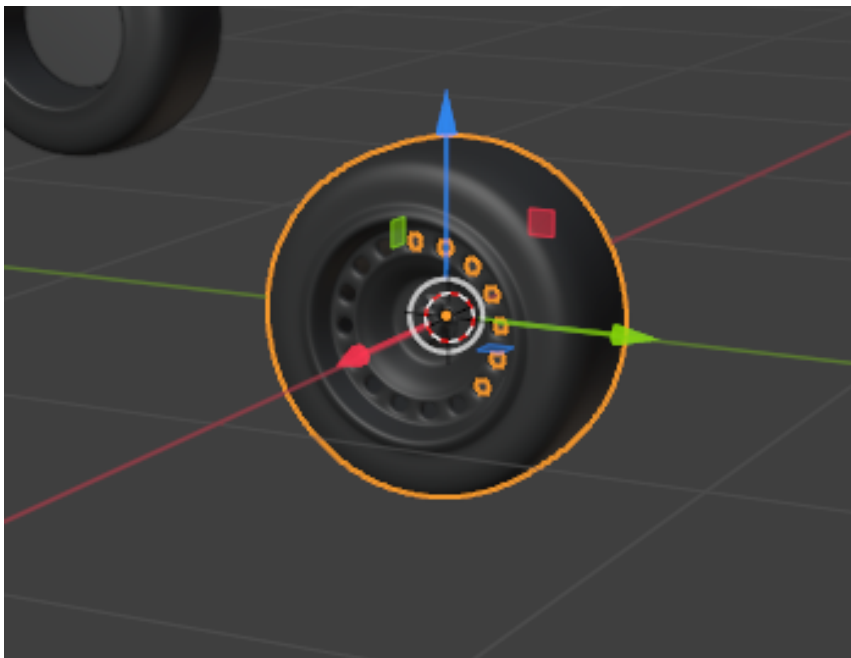


Complicaciones

Mayormente tuve dificultad tanto colocando las llantas en su lugar, pero esto lo resolví chequeando bien la jerarquía en que estaba instanciando las llantas, y además chequeando bien las traslaciones que hacía a la hora de instanciar cada una, instancie primero la del lado izquierdo enfrente, luego la de la derecha igual del frente, para después instanciar la llanta trasera del lado derecho y por último la llanta izquierda de la parte de atrás.



Otra dificultad que tuve al principio fue que las llantas no tenían su origen centrado desde la exportación del modelo en Blender, por tanto las rotaciones resultaban extrañas, las llantas no giraban sobre su propio eje pero lo resolví exportando los modelos de nuevo para que las llantas solo giraran sobre el eje x en su propio eje.



Conclusión

En la práctica de esta semana si bien no he tenido problemas a la hora de manejar Blender, me ha costado entender las importancias de los orígenes de cada objeto, en que posición exportarlos respecto al origen, porque todo ello repercute en el

resultado final a la hora de manejar cada primitiva en el programa de C++ y OpenGL, porque puede que al tener un origen distinto, va tener comportamientos extraños como rotaciones indeseadas, a su vez, en este caso no hubo problemas con las escalas pero es algo importante a considerar también, procurando mantener los objetos con escalas unitarias.