



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°7

NOMBRE COMPLETO: Araiza Valdés Diego Antonio

N° de Cuenta: 423032833

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 06

SEMESTRE: 2026-1

FECHA DE ENTREGA LÍMITE: 19/10/2025

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.
 - 1.1. Ejercicio 1: Agregar movimiento con teclado al helicóptero hacia adelante y atrás.
 - 1.2. Ejercicio 2: Crear luz spotlight de helicóptero de color amarilla que apunte hacia el piso y se mueva con el helicóptero

```
in vec4 vCol;
in vec2 TexCoord;
in vec3 Normal;
in vec3 FragPos;
in vec4 vColor;

out vec4 color;

#ifdef COMMONVALS
#define COMMONVALS
#include "stb_image.h"

const int MAX_POINT_LIGHTS = 3;
const int MAX_SPOT_LIGHTS = 5;
#endif
```

Primero aumentamos el número máximo de luces spot ya que como creamos 2 faros necesitamos una luz más.

Window.cpp

```
if (key == GLFW_KEY_Y)
{
    theWindow-> muevex += 1.0;
}
if (key == GLFW_KEY_U)
{
    theWindow-> muevex -= 1.0;
}
if (key == GLFW_KEY_J)
{
    theWindow->muevex2 += 1.0;
}
if (key == GLFW_KEY_H)
{
    theWindow->muevex2 -= 1.0;
}
```

Window.h

```
GLfloat getmuevex() { return muevex; }
GLfloat getmuevex2() { return muevex2; }

GLfloat muevex;
GLfloat muevex2;
```

Luego, siguiendo el mismo formato que muevex en Window.h y Window.cpp, creamos la función muevex2, la cual se mueve con las letras J y H.

```
//HELICÓPTERO
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex2(), 5.0f, 8.0f));
model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Blackhawk_M.RenderModel();
```

Posteriormente al traslate del Helicóptero se le asigno la función muevex2 y se le recorrió 2 unidades más en Z.

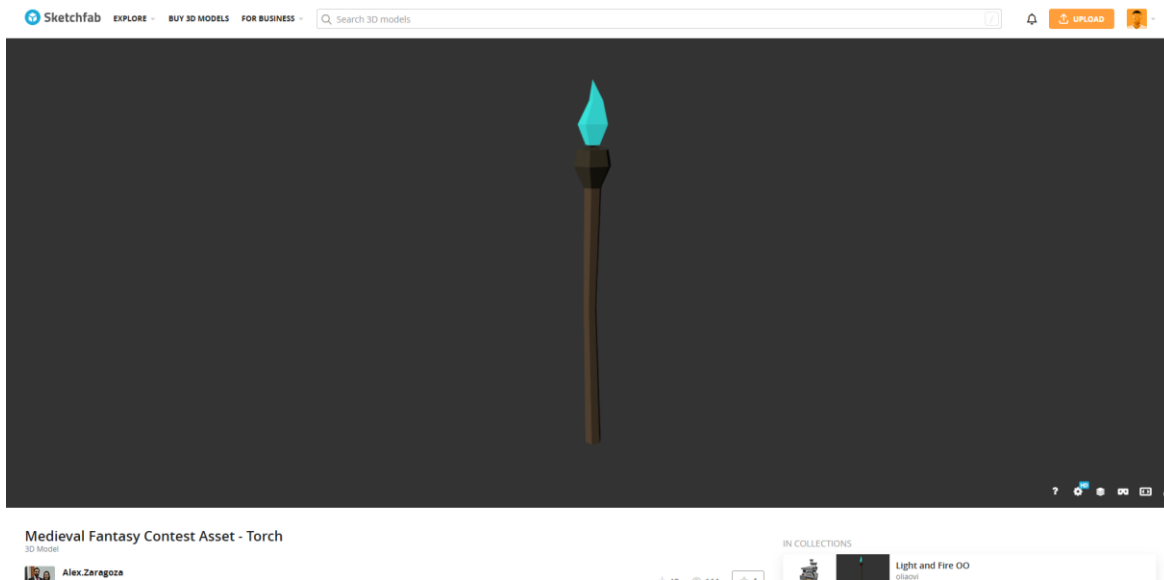
```
//LUZ HELICÓPTERO
spotLights[4] = SpotLight(1.0f, 0.96f, 0.176f, //Color
    1.0f, 1.0f, //Intensidad ambiental y difusa
    0.0f, 4.8f, 8.0f, //Posición
    0.0f, -1.0f, 0.0f, //Dirección
    0.0f, 0.0f, 0.05f, //Atenuación Constante, lineal y exponente
    15.0f); //Edge
spotLightCount++;
glm::vec3 Fdir2(0.0f, -1.0f, 0.0f);
```

Por consiguiente, se creó la spotlight, con un color amarillo linterna, en una posición 0.2 unidades más baja que la del helicóptero, apuntando hacia abajo, con una atenuación exponencial de 0.05 y 15 de Edge. Además, se creó el vector de dirección al que apunta.

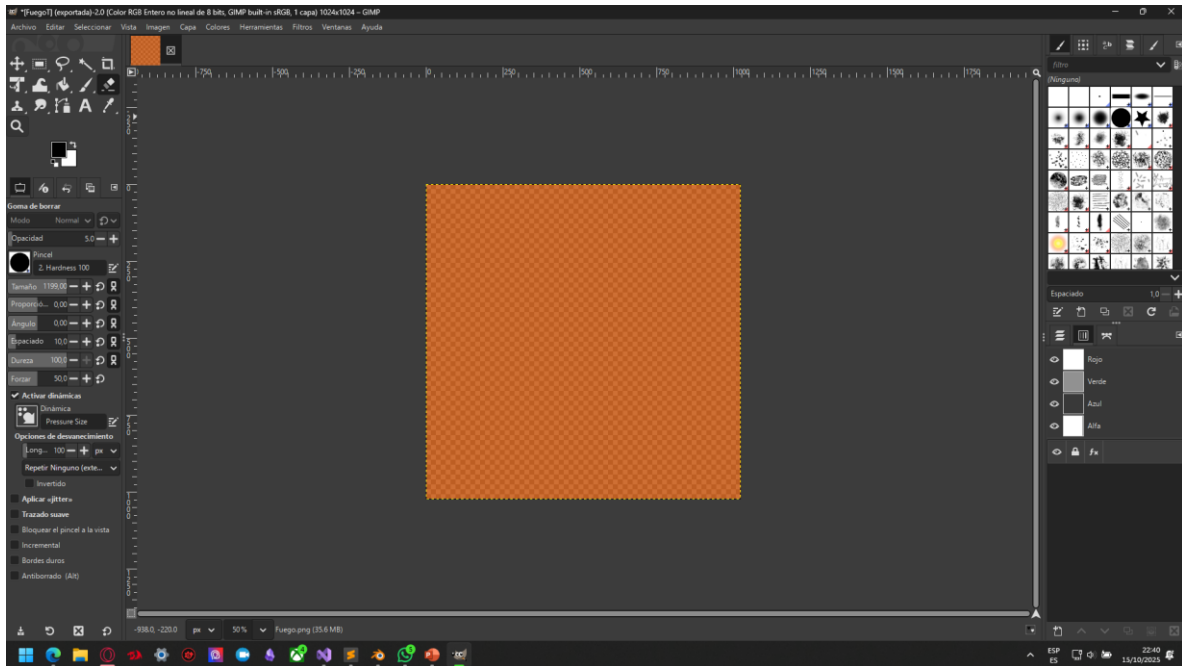
```
glm::vec3 Fpos3 (0 + mainWindow.getmuevex2(), 4.8f, 8.0f);
spotLights[4].SetFlash(Fpos3, Fdir2);
```

Finalmente, dentro del while se creó el vector posición 3 al cual le asignamos el movimiento con la función muevex2 y creamos el flash de la luz.

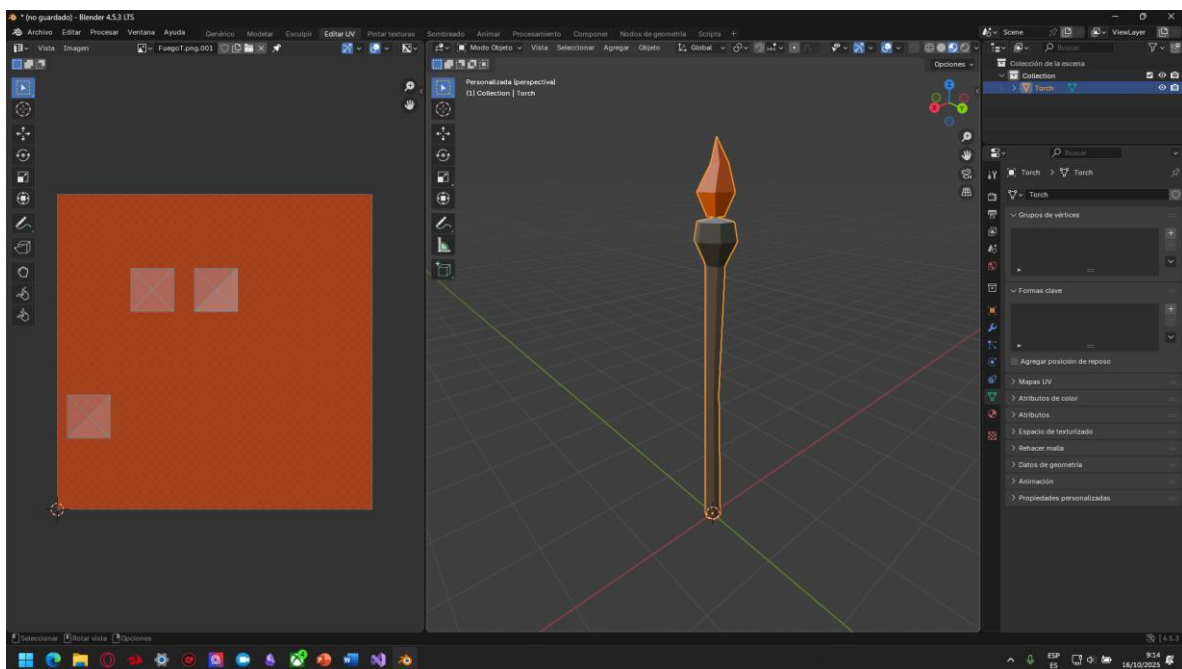
1.3. Ejercicio 3: Añadir en el escenario 1 modelo de lámpara texturizada (que usarán en su proyecto final) y crearle luz puntual blanca



Primero buscamos un modelo de una antorcha que sea lo más acorde posible a la temática, ya que decidimos utilizar antorchas para iluminar, pero solo contábamos con un modelo.

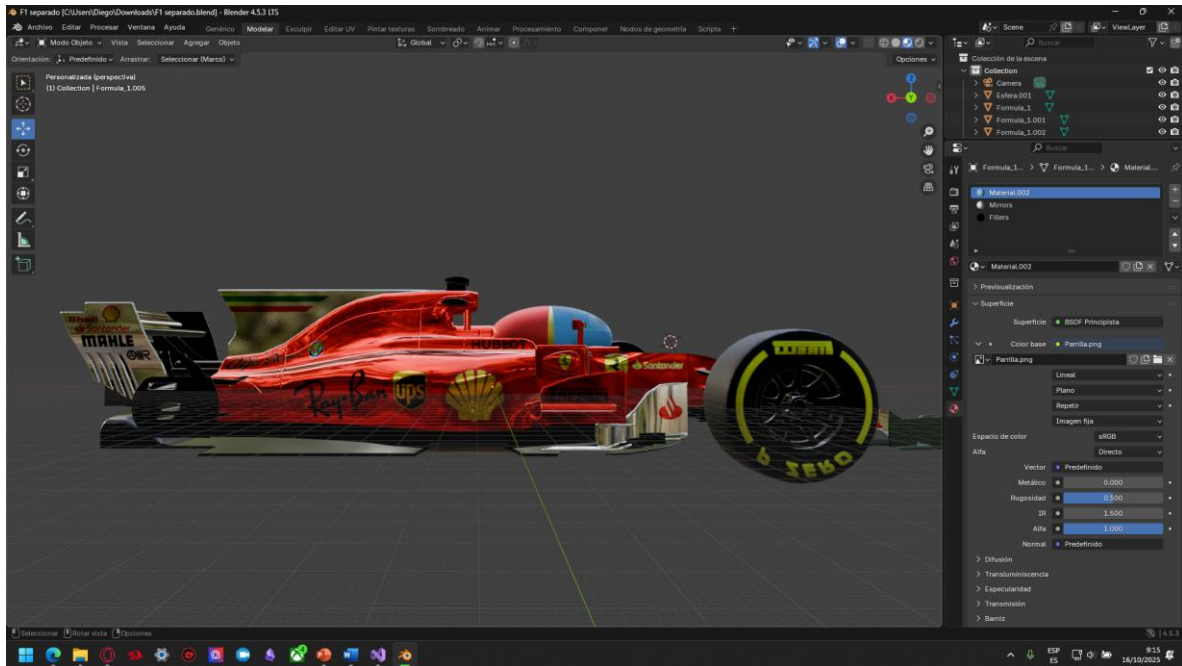


Luego, dentro de Gimp, creamos una textura optimizada para el fuego de la antorcha con una transparencia del 25%,



Así texturizamos la antorcha y la exportamos con formato .obj para utilizarla dentro de OpenGL. Para esto fue necesario invertir las normales de una de las caras del fuego ya que estaba invertida y no se apreciaba la transparencia en OpenGL.

<https://sketchfab.com/3d-models/medieval-fantasy-contest-asset-torch-b978b1bc59414a3288164d7092869b78>



Además, se invirtió la textura del vehículo ya que estaba espejeada.

```

F1Cuerpo_M = Model();
F1Cuerpo_M.LoadModel("Models/F1Cuerpo.obj");
F1Cofre_M = Model();
F1Cofre_M.LoadModel("Models/F1Cofre.obj");
LlantaDI_M = Model();
LlantaDI_M.LoadModel("Models/LlantaDI.obj");
LlantaDD_M = Model();
LlantaDD_M.LoadModel("Models/LlantaDD.obj");
LlantaTI_M = Model();
LlantaTI_M.LoadModel("Models/LlantaTI.obj");
LlantaTD_M = Model();
LlantaTD_M.LoadModel("Models/LlantaTD.obj");
Torch_M = Model();
Torch_M.LoadModel("Models/Torch.obj");

Model F1Cuerpo_M;
Model F1Cofre_M;
Model LlantaDI_M;
Model LlantaDD_M;
Model LlantaTI_M;
Model LlantaTD_M;
Model Torch_M;

```

Posteriormente, declaramos los modelos y los cargamos en el código

```

//TORCH
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(0.0f, -1.0f, -10.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Torch_M.RenderModel();
glDisable(GL_BLEND);

```

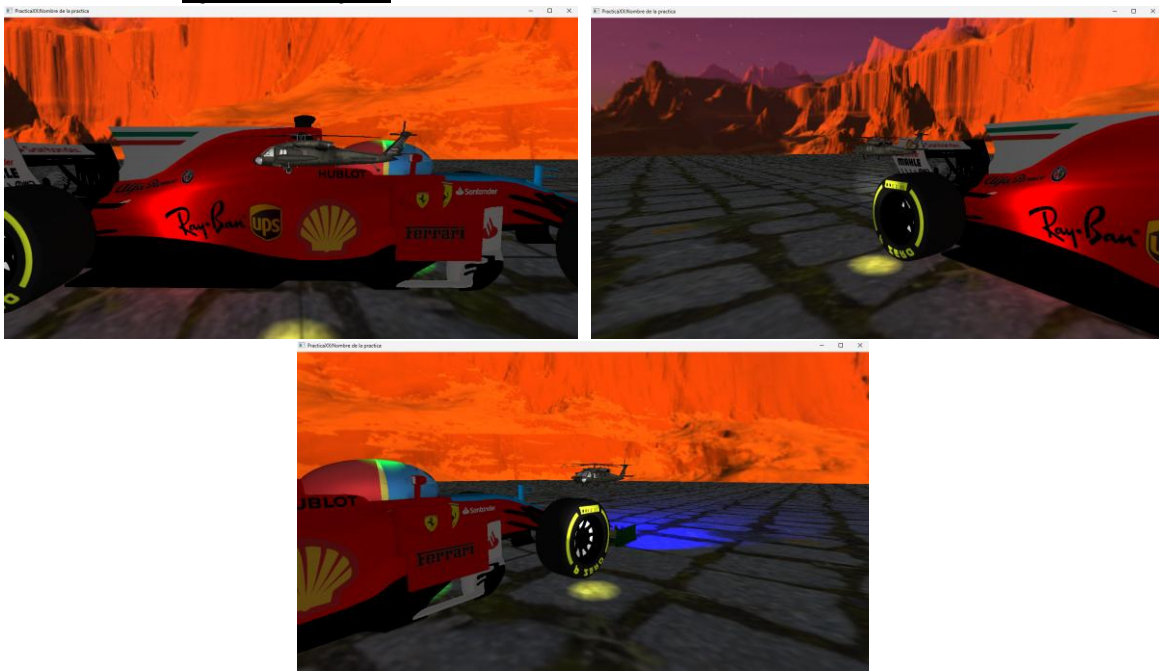
Luego, declaramos y dibujamos la antorcha, para esto recorrimos el glDisable que estaba después del cofre hasta después de la antorcha. Además se desplazó el Agave.

```
//LUZ TORCH
pointLights[1] = PointLight(1.0f, 1.0f, 1.0f,    //Color
    0.2f, 0.9f,                                //Intensidad ambiental y difusa
    0.0f, 3.1f, -10.0f,                         //Posición
    0.45f, 0.1f, 0.1f);                        //Atenuación Constante, lineal y exponente
pointLightCount++;
```

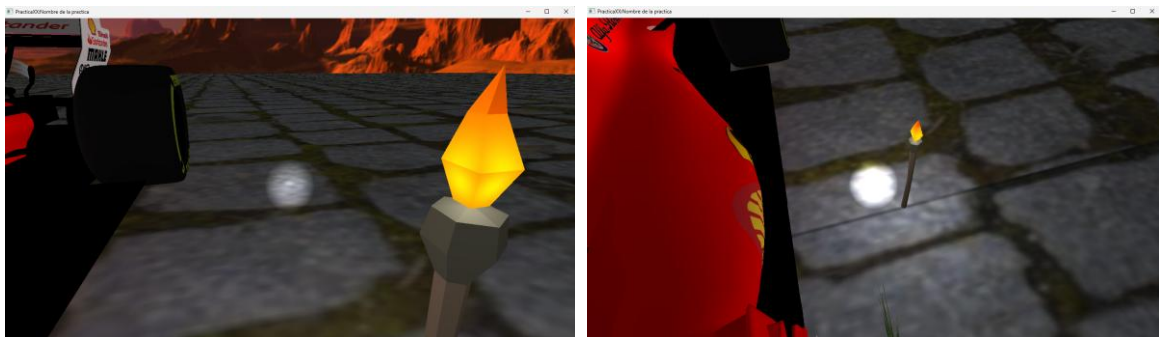
Finalmente, creamos la pointlight de la antorcha, con color blanco, centrada en el fuego de la antorcha y con una intensidad y atenuación tal que ilumine y se termine de texturizar el fuego de la antorcha.

Evidencias:

1.3.1. Ejercicio 1 y 2:



1.3.2. Ejercicio 3:



1. 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

R: No hubo problemas.

2. Conclusión:

1. Los ejercicios del reporte: Complejidad, Explicación.

Respecto a los ejercicios realizados, fueron bastante fáciles, bastante adecuados a lo visto en clase ya que los primeros 2 ejercicios fueron hacer lo que hicimos en el ejercicio y el tercero solo fue hacer lo que habíamos hecho previamente y crear una pointlight. Además tuve que investigar como invertir las normales y las texturas.

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

La explicación fue muy buena, para lo que fue la creación de luces no hubo mayor problema, la atenuación fue bastante fácil de entender

3. Conclusión

Personalmente creo que se cumplieron los objetivos de la práctica. Con el ejercicio y la práctica aprendí como implementar fuentes y modelos de iluminación. Con los ejercicios aprendí las diferencias entre las 3 luces (ambiental, puntual y spot), como declararlas y como modificar sus parámetros (posición, intensidad, dirección). Además, aprendí a asignarles una traslación dinámica (movimiento) y crear los modelos de iluminación.