



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° 07

NOMBRE COMPLETO: Lopez Flores Diego Alberto

N° de Cuenta: 315081143

GRUPO DE LABORATORIO: 11

GRUPO DE TEORÍA: 04

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 02 de Agosto de 2024

CALIFICACIÓN: _____

1. Agregar movimiento con teclado al helicóptero hacia adelante y atrás.

Primero tenemos que inicializar las teclas H y J, en los diferentes .cpp, primero en Windows.h en las variables agregamos nuestra variable helicóptero

```
GLfloat gethelicopter() { return helicopter; }
```

```
GLfloat helicopter;
```

Después en Windows.cpp definimos cuanta unidades se moverá y que teclas le asignamos

```
helicopter = 2.0f;
```

```
if (key == GLFW_KEY_H)
{
    theWindow-> helicopter += 1.0;
}
if (key == GLFW_KEY_J)
{
    theWindow-> helicopter -= 1.0;
}
```

Y agregamos la Gethelicopter() en la taslacion del helicóptero

```
//Helicoptero
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.gethelicopter(),
5.0f, 6.0));
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();
```

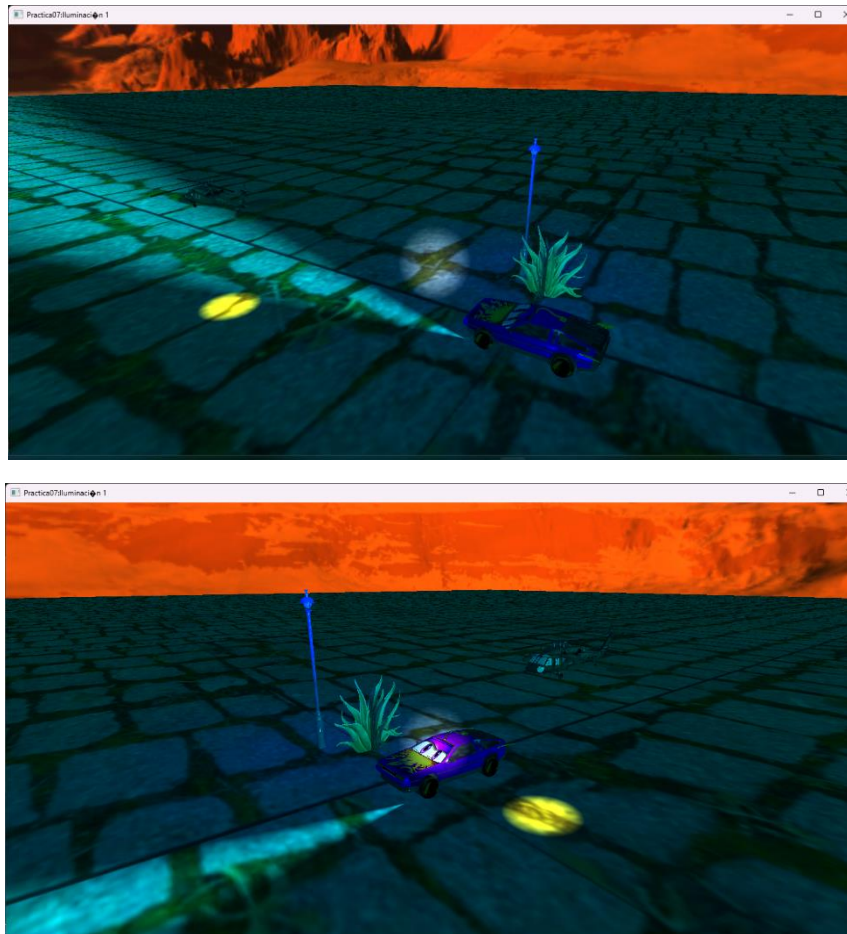
2. crear luz spotlight de helicóptero de color amarilla que apunte hacia el piso y se mueva con el helicóptero

Ahora creamos la luz Spotlight de color amarilla y que se proyecte hacia abajo

```
//luz helicoptero
spotLights[2] = Spotlight(0.9f, 0.80f, 0.0f, //color
1.0f, 2.0f, //intensidad ambiental y difusa
0.0f, 5.0f, 6.0f, //posicion
0.0f, -1.0f, 0.0f, //direccion
1.0f, 0.0f, 0.0f, //constnate lineal y exponencial
15.0f); //apertura
spotLightCount++;
```

Al igual que para el ejercicio práctico declaramos el vector de posición de la luz, su ubicación y el input del helicóptero

```
//helicopter
glm::vec3 lowerLight2 = glm::vec3(0.0f + mainWindow.gethelicopter(), 5.0f,
6.0f);
spotLights[2].setFlash(lowerLight2, glm::vec3(0.0f, -1.0f, 0.0f));
```



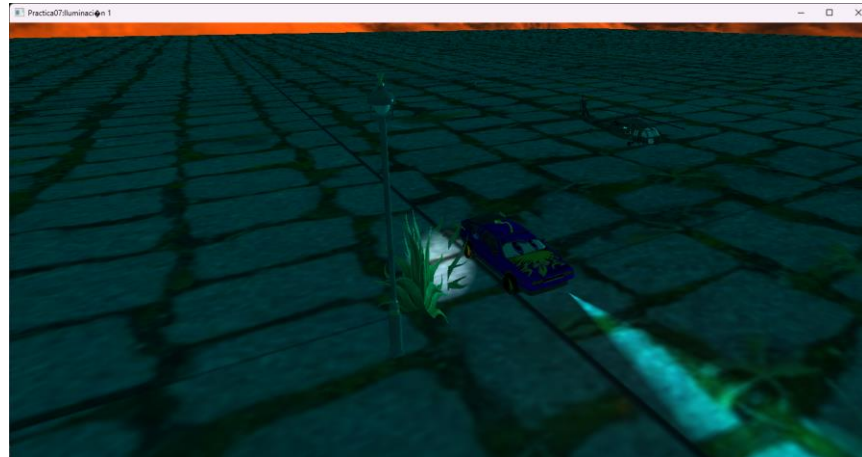
3. Añadir en el escenario 1 modelo de lámpara texturizada (diferente a los que usarán en su proyecto final) y crearle luz puntual blanca

Bajaremos el modelo, lo optimizamos y exportamos como obj y su textura en tga, lo cargaremos en OpenGL

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

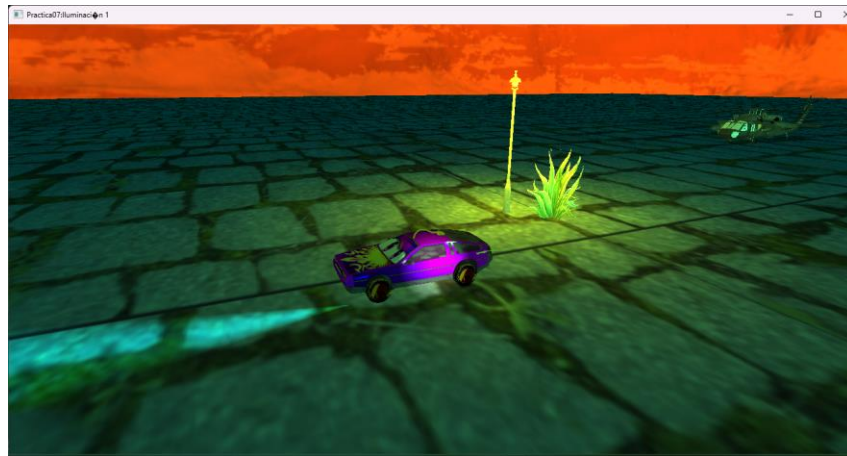
y después instanciamos dándole una escala adecuada para el entorno

```
//Farola
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-2.0f, -1.0f, -6.0));
model = glm::scale(model, glm::vec3(1.8f, 1.8f, 1.8f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Farola_M.RenderModel();
```



Para la luz creamos la luz y la ubicamos en el plano donde esta la farola, solo se ajusta la altura para que este bien posicionada, modificamos la intensidad ambiental y difusa para que sea mas inmersiva

```
//Luz Farola
pointLights[0] = PointLight(0.8f, 0.8f, 0.0f, //color
8.0f, 1.5f, //intensidad ambiental y difusa (Rango, Color)
-2.0f, 6.0f, -6.0f, //direccion
0.3f, 0.2f, 0.1f);
pointLightCount++;
```



Conclusión

Esta práctica sirvió como un repaso de lo antes echo, el asignar un input para que este modifique una variable, en este caso, la posición del helicóptero, el colocar otra luz, pero ahora proyectado el plano XZ y agregarle el movimiento de otro input

Y por ultimo el optimizar un modelo, cargarlo con su textura y darle una luz estática para que este ilumine el ambiente, fue un refuerzo a lo antes visto durante el semestre

Referencias

"London street lamp" (<https://skfb.ly/oXX8L>) by mrrobinson2502 is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).