



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: Aguilar Pérez José Ramón

N° de Cuenta: 317515048

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 05/abril/2025

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejercicios realizados.

- Agregar movimiento con teclado al helicóptero hacia adelante y atrás.
- Crear luz Spotlight de helicóptero de color amarilla que apunte hacia el piso y se mueva con el helicóptero
- Añadir en el escenario 1 modelo de lámpara texturizada y crearle luz puntual blanca.

Para el primer ejercicio, se utilizó la función proporcionada dentro de *Window.cpp* para mover objetos en el plano X. Si se presiona la tecla Y se mueve una unidad hacia adelante y si se presiona la tecla U se mueve una unidad hacia atrás. Este movimiento se declara dentro de la traslación del helicóptero en el eje en el que se desea que se mueva, en este caso el movimiento de adelante y atrás se hace sobre el eje X.

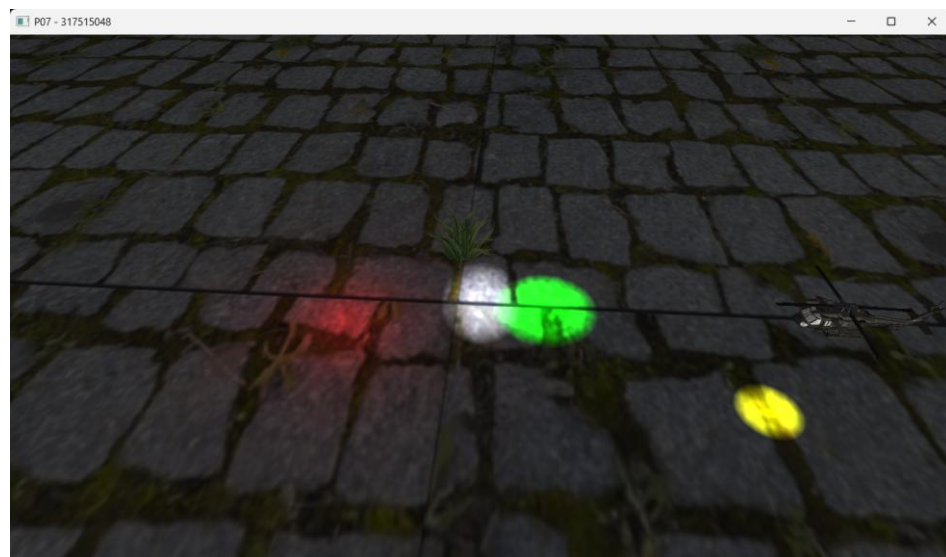
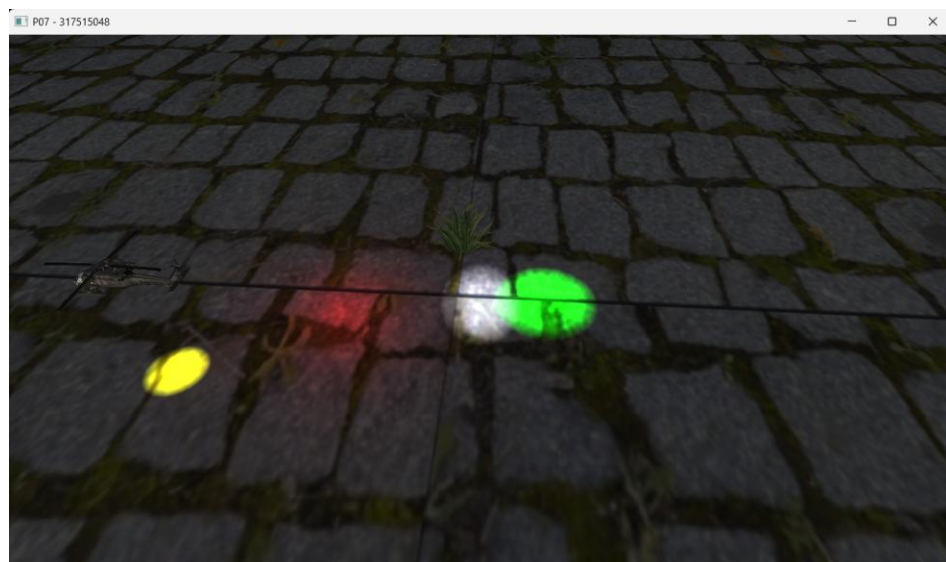
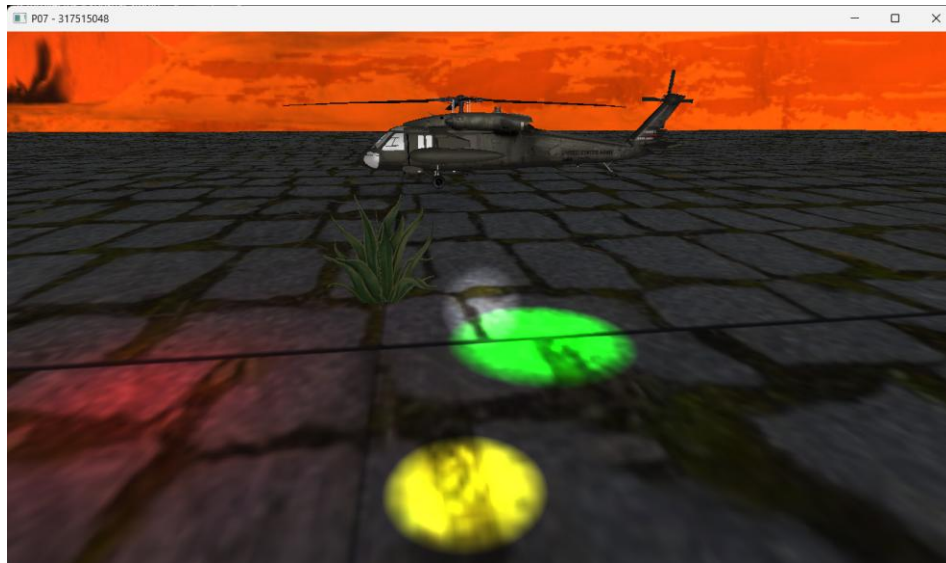
```
//Movimiento del vehiculo
if (key == GLFW_KEY_Y)
{
    theWindow-> muevex -= 1.0;
}
if (key == GLFW_KEY_U)
{
    theWindow-> muevex += 1.0;
}

//***** Instancia Helicóptero *****/
//Helicoptero
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex(), 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();
```

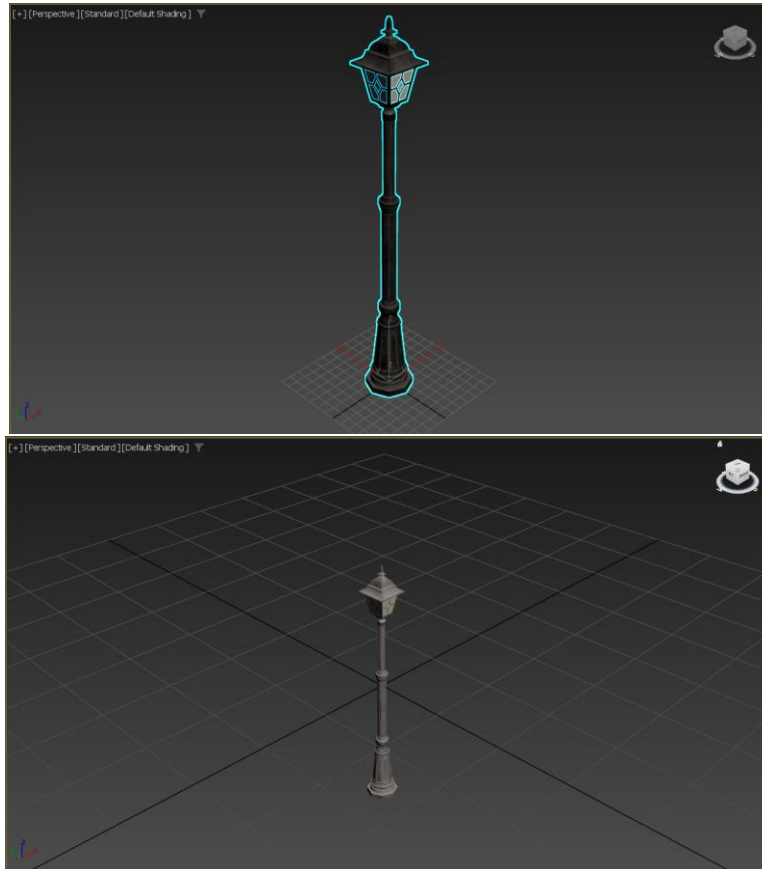
Para la creación de la luz amarilla del helicóptero, se instanció una luz spotlight con la información necesaria. Se le asignó el color amarillo a la luz y como debe apuntar al suelo desde la base del helicóptero, apunta en la dirección de Y negativa. Finalmente, se declara la posición de la luz spotlight un poco debajo de la posición del helicóptero, además que dentro del eje X se agrega la función que permite el movimiento del helicóptero para que así la luz siga en todo momento al helicóptero.

```
//Luz amarilla del helicóptero
spotLights[2] = SpotLight(1.0f, 1.0f, 0.0f,
    1.0f, 2.0f, //primer termino -> intensidad de iluminacion
    5.0f, 10.0f, 0.0f, //Posicion foco invisible
    0.0f, -5.0f, 0.0f, //Direccion donde apunta
    1.0f, 0.0f, 0.0f, //Ecuacion de segundo grado
    15.0f); //Angulo del cono
spotLightCount++;

//Helicoptero
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex(), 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));
spotLights[2].SetPos(glm::vec3(0.0f + mainWindow.getmuevex(), 4.8f, 6.0f)); //Luz de linterna amarilla
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Blackhawk_M.RenderModel();
```



Para el modelo de la lámpara, se descargó un modelo de internet que ya tuviera texturas. Se trabajó con un archivo .OBJ y dentro de 3dsMax se ajustó la escala del objeto, así como se ajustó el pivote del modelo para que se encontrara en el origen, lo que permitió que los movimientos que se hicieran dentro de OpenGL fueran más fáciles de realizar.



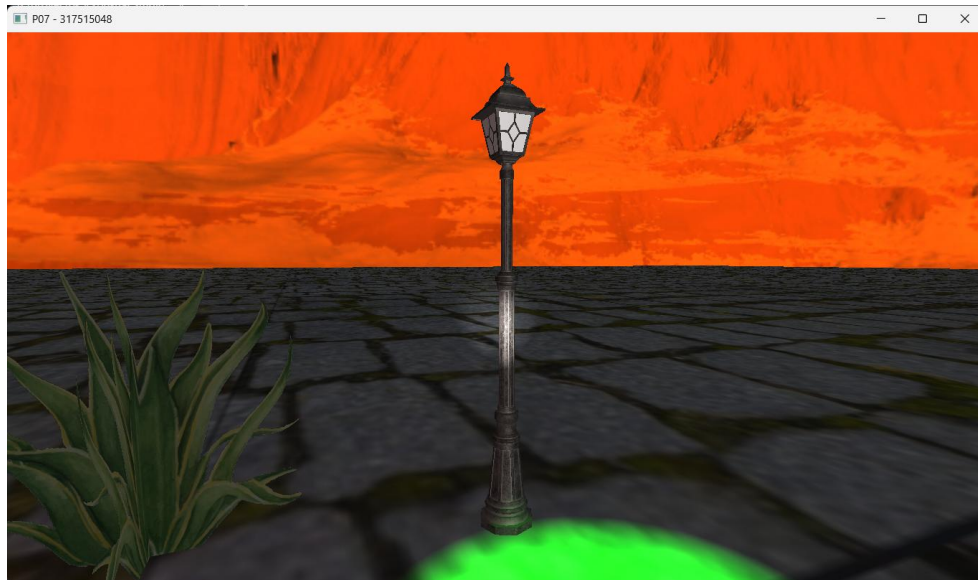
Una vez que se optimizó el modelo, se agregó el .OBJ y .MTL a la carpeta *Models* del proyecto, mientras que las texturas de la lámpara se agregaron a la carpeta *Textures* del proyecto. Dentro del programa, se declara el modelo de la lámpara y su ruta del archivo. Mientras que en el *main*, se renderiza el modelo en un lugar donde no estorbe al helicóptero.

```
//Modelos
Model Blackhawk_M; Lampara_M = Model();
Model Lampara_M; Lampara_M.LoadModel("Models/lampara.obj");

//***** Instancia Lampara *****
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(5.0f, 2.5f, -3.0f));
model = glm::scale(model, glm::vec3(0.2f, 0.2f, 0.2f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Lampara_M.RenderModel();
```


Finalmente, se creó una luz puntual para la lámpara. Esta luz es de color blanco y se encuentra justo en la pantalla de la lámpara. Para los valores de atenuación e intensidad, se fueron modificando hasta obtener el resultado deseado.

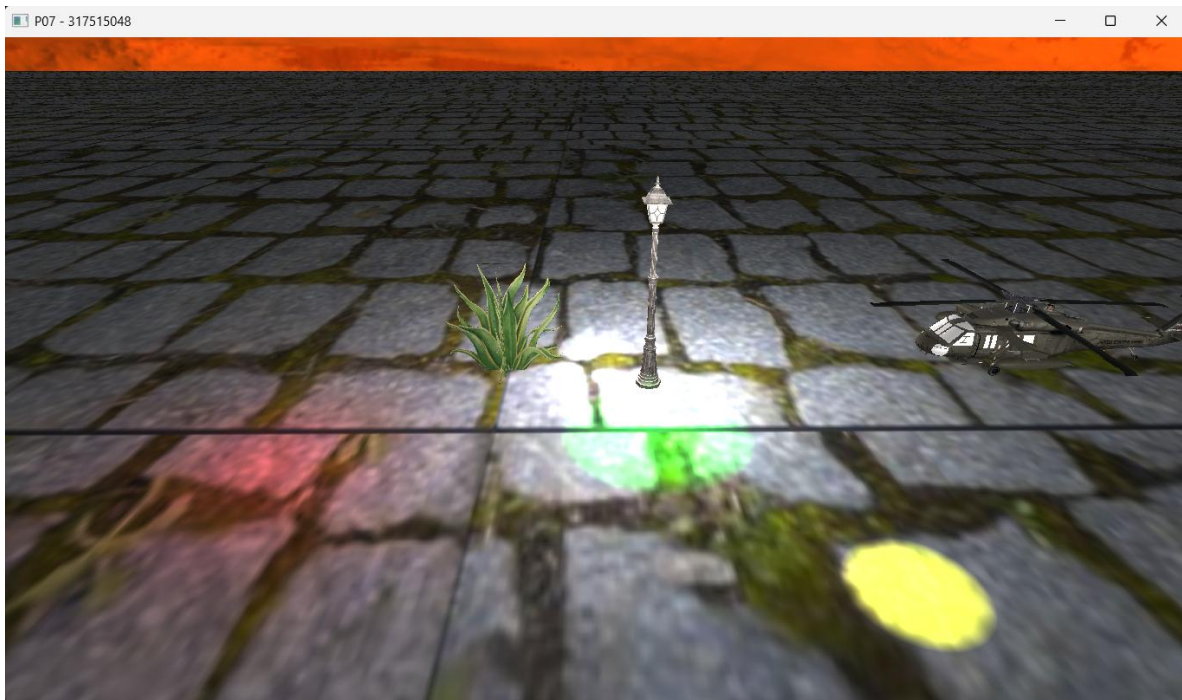
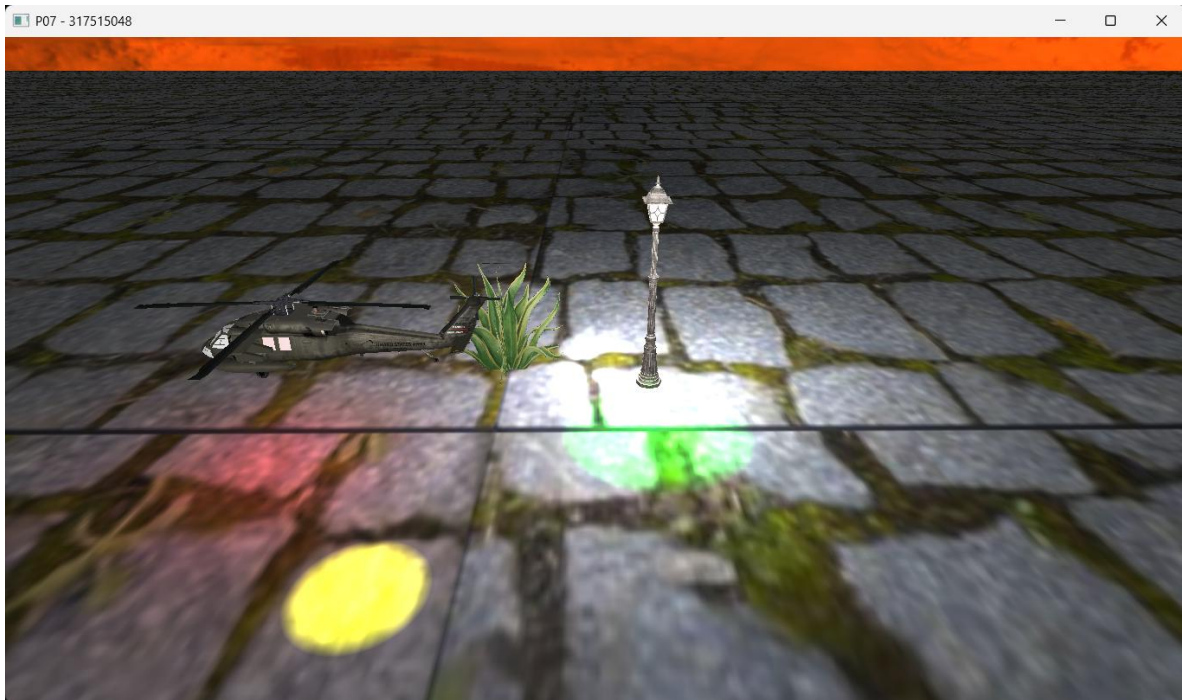
```
//Luz puntual para la lámpara
pointLights[1] = PointLight(1.0f, 1.0f, 1.0f, //Color blanco para la luz
    0.5f, 1.0f, //Intensidad de la luz
    5.0f, 5.0f, -3.0, //Posición
    0.3f, 0.05f, 0.01f); //Atenuación
pointLightCount++;
```



Lámpara sin luz puntual.



Lámpara con luz puntual.



2.- Problemas presentados

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

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 las diferencias entre los distintos tipos de luces que pueden aplicarse a un escenario. Además, como este tipo de elemento es muy empírico, se tuvieron que hacer varias pruebas hasta obtener el resultado deseado.

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 los distintos tipos de luces, así como los elementos que las conforman como la intensidad, su atenuación y color fue bastante clara y precisa, lo que permitió realizar los ejercicios sin contratiempos.

c. Conclusión

Esta práctica me permitió comprender mejor cómo funcionan las luces dentro de OpenGL. Dentro de un escenario, solo puede existir una luz principal, que en este caso simula ser la luz del sol, mientras que para las luces spotlight y puntuales puede haber, como máximo establecido para no forzar mucho el equipo, 8 luces en total. Fue interesante manipular los distintos elementos que conforman las luces para encontrar un resultado satisfactorio.

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>
- De programación, T. (2016, abril 24). Iluminación - Tipos de Luces. Blogspot.com. <https://acodigo.blogspot.com/2016/04/iluminacion-tipos-de-luces.html>
- Street lamp single. (s/f). Turbosquid.com. Recuperado el 4 de abril de 2025, de <https://www.turbosquid.com/3d-models/street-lamp-single-1841096>