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 Nº 08

NOMBRE COMPLETO: MORENO SANTOYO MARIANA

Nº de Cuenta: 319170252

GRUPO DE LABORATORIO: 11

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 02/04/2025

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.

2.- 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

3.- Conclusión:

a. *Los ejercicios del reporte: Complejidad, Explicación.*
b. *Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica*
c. *Conclusión*

Bibliografía en formato APA

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

Para realizar esta parte del ejercicio, se optó por replicar exactamente el mismo movimiento del carro, tomando o generando otras articulaciones para este para que fueran teclas independientes a las del carro.

WINDOW.H

```
GLfloat getarticulacion4() { return articulacion4; }
GLfloat getarticulacion5() { return articulacion5; }
```

```
private:
    GLFWwindow* mainWindow;
    GLint width, height;
    GLfloat rotax, rotay, rotaz, articulacion1, articulacion2, articulacion3,valor,articulacion4, articulacion5 ;
```

WINDOW.CPP

```
articulacion4 = 0.0f;
articulacion5 = 0.0f;
```

```
if (key == GLFW_KEY_N)
{
    theWindow->articulacion4 += 0.1;
}

if (key == GLFW_KEY_M)
{
    theWindow->articulacion5 -= 0.1;
    }
```

MAIN.CPP

```
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, 20.0f, 0.0f));
model = glm::scale(model, glm::vec3(1.0f, 1.0f, 1.0f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
model = glm::rotate(model, 180 * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::translate(model, glm::vec3(0.0f, 1.0f, 0.0f) * mainWindow.getarticulacion4()); //traslacion del cuerpo adelante
model = glm::translate(model, glm::vec3(0.0f, 1.0f, 0.0f) * mainWindow.getarticulacion5()); //traslacion del cuerpo hacia atras
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Blackhawk_M.RenderModel();
```

NOTA *Se realizaron cambios en las rotaciones y escala del helicóptero tanto para facilidad de la realización de los traslados como cambios en las escalas para darle un efecto un tanto mas realista respecto al spotlight que actuaba como el flash o luz de este*

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

De igual manera a como se realizo en el ejercicio previo de esta práctica se recurrió a exactamente la misma técnica para hacer que el carrito se moviera, y a método no muy exacto de prueba y error se obtuvieron los valores que a mi parecer daban un efecto mas realista en cuestiones del haz de luz, haciendo uso de los mismos valores del helioptero para posicionar la luz y de ahí ajustar.

MAIN.CPP

```
spotLights[2] = SpotLight(0.8f, 0.8f, 0.0f, // color    // Color amarillo chillón (RGB)
    0.5f, 0.1f,           // aIntensity y dIntensity
    0.0f, 35.0f, 0.0f,     // Posición del helicóptero en el centro y a 5 unidades de altura
    0.0f, -10.0f, 0.0f,    // La luz apunta hacia abajo (eje Y negativo)
    1.0f, 0.0f, 0.00f,
    30.0f);
spotLightCount++;
```

```
float a3 = mainWindow.getarticulacion4(), a4 = mainWindow.getarticulacion5();
glm::vec3 lightPosition2 = glm::vec3(0.0f, 0.0f, 1.0f) + glm::vec3(0.0f, 0.0f, 1.0f) * (a3 + a4)
spotLights[2].SetFlash(lightPosition2, glm::vec3(0.0f, -5.0f, 0.0f));
```

```
//información al shader de fuentes de iluminación
shaderList[0].SetDirectionalLight(&mainLight);
shaderList[0].SetPointLights(pointLights1, pointLightCount_ARRAY1);
shaderList[0].SetSpotLights(spotLights, spotLightCount);
```

EJECUCIONES DE LA LUZ:

3.- Añadir en el escenario 1 modelo de lámpara texturizada y crearle luz puntual blanca

Para realizar esta parte del ejercicio se tomo un modelo de internet que venia texturizado, pero sin las imágenes por lo que se opto que fue mejor re texturizarlo con dos texturas sencillas:

Se Re texturizo en Bender y de manera normal se exporto en formato obj al visual:

```
Lamp = Model();
Lamp.LoadModel("Models/lamp.obj");
```

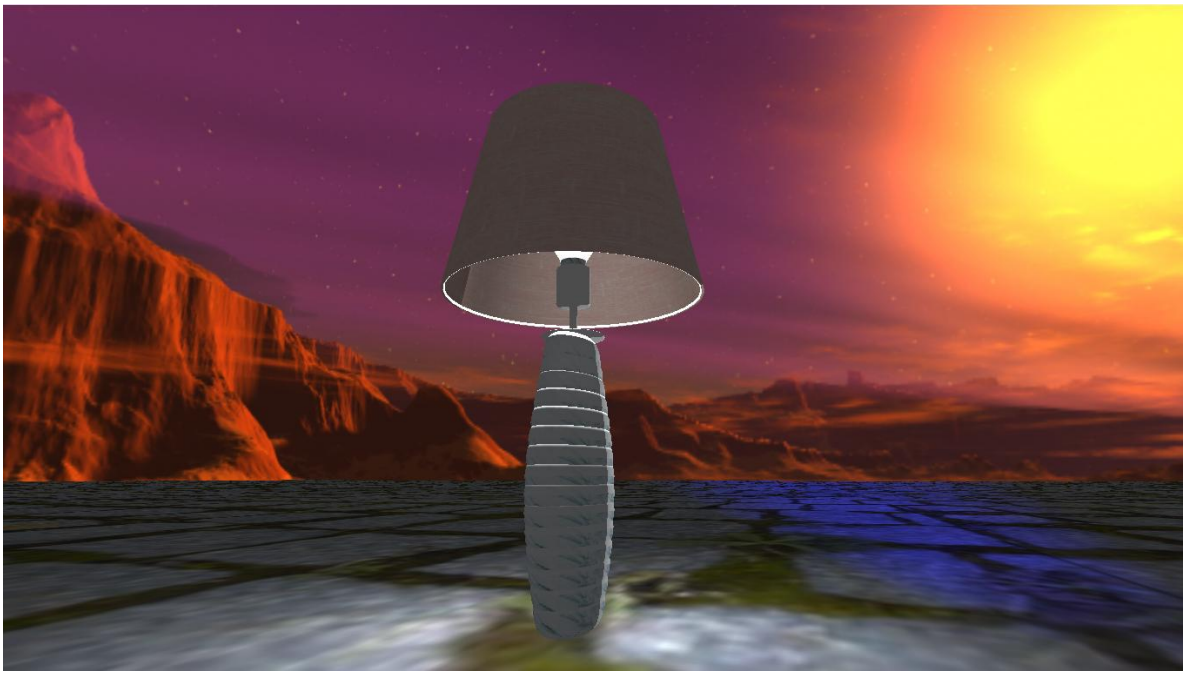
Se ajustaron escalas y traslaciones:

```
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(8.0f, 1.0f,8.0f));
model = glm::scale(model, glm::vec3(0.1f, 0.1f, 0.1f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Lamp.RenderModel();
```

Con el point light blanco creado en el ejercicio anterior practica 07 únicamente se posiciono en la dirección adecuada y se mando al shader para el render:

```
pointLights1[0] = PointLight(1.0f, 1.0f, 1.0f,
    0.0f, 10.0f,
    8.0f, 3.0f, 8.0f,
    0.0f, 0.1f, 0.5f);
pointLightCount_ARRAY1++;
```

RESULTADO



CONCLUSIONES:

La realización de esta practica me fue bastante sencilla ya que tanto en el anterior como en el ejercicio se realizaron cosas bastante similares, además de que texturizar la lampara fue relativamente sencillo, únicamente me pareció algo tedioso manejar las luces a prueba y error y me gustaría poco a poco encontrar una forma más rápida o más exacta.