



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: García Soto Jean Carlo

N° de Cuenta: 319226304

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 05/04/2025

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.

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

Para agregar movimiento al helicóptero tendremos reutilicé una función que ya teníamos definida en nuestro **Window.h** que es un **getmuevex()**, dado que el helicóptero está mirando hacia el eje -X, su único desplazamiento va a ser sobre ese eje, por lo que solo haremos que el valor que estará cambiando será la coordenada X

```
412 //Helicoptero
413 model = glm::mat4(1.0);
414 model = glm::translate(model, glm::vec3(0.0f, 5.0f, 6.0f));
415 model = glm::translate(model, glm::vec3(mainWindow.getmuevex(), 0.0f, 0.0f)); // Mover(trasladar) con Y y U
416 model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
417 model = glm::rotate(model, -90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
418 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
419 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
420 Blackhawk_M.RenderModel();
```



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

Se creó un nuevo spotlight, el cual es de color amarillo, a este se le dio la posición relativa del helicóptero y se ajustó para que no iluminara por dentro del helicóptero, sino que estuviera afuera para que pudiera alumbrar al suelo.

Se le dio una atenuación con la que se pareciera a una luz amarilla de la vida real, al igual que se le configuró su ángulo de apertura de esta, para que no se viera ni muy grande o pequeña.

```

315 //Luz del helicoptero
316 spotLights[3] = SpotLight(0.7f, 0.7f, 0.0f,
317     1.0f, 1.0f, //Rango, color
318     0.0f, 4.0f, 6.0, //Posición
319     0.0f, -1.0f, 0.0f, //Vector de dirección
320     0.0f, 0.4f, 0.0f, //Atenuación (No poner en ceros)
321     20.0f); //angulo de apertura
322 spotLightCount++;

```

Se agregó un setFlash() con el motivo de que cada que se mueva el helicóptero se mueva la luz amarilla junto a este, se le pasó la misma función de translación del helicóptero para que cuando reciba por el teclado el valor para moverse se mueva de misma cantidad y dirección.

```

371
372 //Mueve Luz amarilla del helicoptero de acuerdo si se mueve el helicoptero o no
373 glm::vec3 mueveLuz2 = glm::vec3(0.0f, 4.5f, 6.0) + glm::vec3(mainWindow.getmuevex(), 0.0f, 0.0f); //Tecla Y y U
374 spotLights[3].setFlash(mueveLuz2, glm::vec3(0.0f, -1.0f, 0.0f));
375

```



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

La lámpara se escogió de internet, el cual se tuvo que texturizar desde blender, se usó imágenes de texturizado para el metal y de acrílico. Se eliminaron del modelos los objetos que venían como extra y se realizó el texturizado de este mismo.



Una vez realizado esto, se importaron en OpenGL el modelo .obj y la trasladamos en una posición en el cual se pudiera ver el efecto de la luz que se le va a agregar. Se creó para esta lámpara un pointLight de color blanco, y a esta también anduvimos ajustando sus coordenadas de donde aparece y su atenuación.

```
65      //Modelo de la lámpara
66      Model Lampara_M;
67
250      Lampara_M = Model();
251      Lampara_M.LoadModel("Models/Lampara.obj");
252
279      //Lampara
280      pointLights[1] = PointLight(1.0f, 1.0f, 1.0f,    //Blanca
281      2.0f, 1.0f,
282      12.0f, 10.0f, 20.0,
283      1.0f, 0.09f, 0.03f);    //con, lin, exp
284      pointLightCount++;
```

```

422 //Lámpara
423 model = glm::mat4(1.0);
424 model = glm::translate(model, glm::vec3(5.0f, -0.95f, 20.0));
425 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
426 Lampara_M.RenderModel();

```



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

No tuve problemas a la hora de hacer esta práctica dado que gran parte de esta, fue realizada con ejercicios que ya habíamos hecho en anteriores prácticas. Como mucho fue encontrar la configuración o valores para la atenuación que más me agradaran para las luces que usé.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

Considero yo que no tuvo mucha dificultad esta práctica, por lo que fue relativamente sencilla, lo único que fue algo tardado, fue encontrar la configuración ideal o la que más me convenciera para poner a las luces, por cómo se había dicho iba a ser por prueba y error el encontrar la configuración ideal para nuestras luces, de ahí en fuera no hubo mucha dificultad por lo que siento que esta práctica fue muy buena para poder aprender el manejo de las luces en los objetos o modelos que estaremos trabajando.

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

Considero que no faltó explicar, dado que todo se nos explicó en el laboratorio y aun así no se nos pedía una gran modificación a los parámetros de las luces, por lo que estuvo bien la explicación.

c. Conclusión

Gracias a esta práctica pude aprender a como crear y modificar luces, el como podemos hacer uso de arreglos, para cambiarlos y hacer efectos con esta misma, lo cual considero que me va a servir para poder realizar el proyecto ya que los efectos de las luces hacen que todo tenga un aspecto muy diferente de ver alguna cosa u objeto.

1. Bibliografía en formato APA

- Brown, C. W. (s.f.). 9.7 - Light Attenuation. LearnWebGL. Recuperado el 5 de abril de 2025, de https://learnwebgl.brown37.net/09_lights/lights_attenuation.html
- Khronos Group. (s.f.). glLight. OpenGL Reference Pages. Recuperado el 5 de abril de 2025, de <https://registry.khronos.org/OpenGL-Refpages/gl2.1/xhtml/glLight.xml>
- LearnOpenGL. (s.f.). Light casters. Recuperado el 5 de abril de 2025, de <https://learnopengl.com/Lighting/Light-casters>
- Valve Developer Community. (2024). Constant-Linear-Quadratic Falloff. Recuperado el 5 de abril de 2025, de https://developer.valvesoftware.com/wiki/Constant-Linear-Quadratic_Falloff