



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:** Arroyo Ramírez Carlos Alberto

**N° de Cuenta:** 320185865

**GRUPO DE LABORATORIO:** 03

**GRUPO DE TEORÍA:** 04

**SEMESTRE** 2026-1

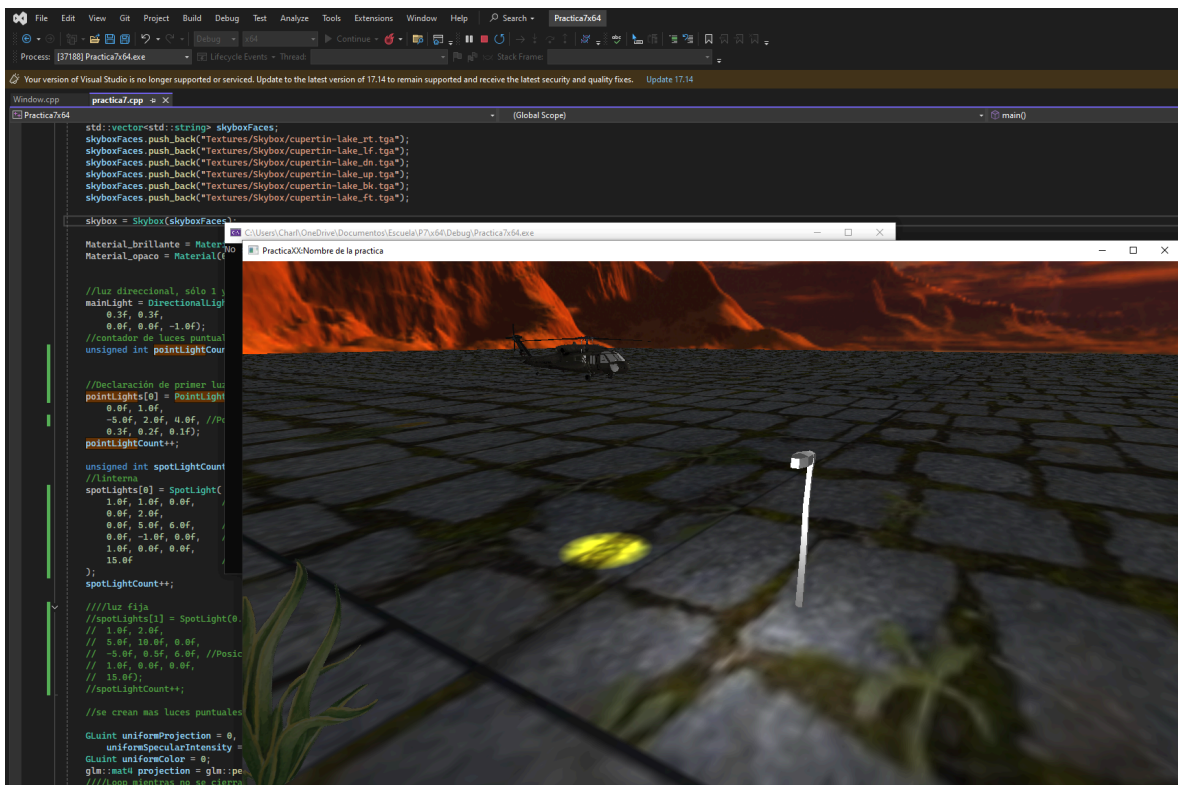
**FECHA DE ENTREGA LÍMITE:** 18/10/2025

**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.

Para esta práctica dimos movimiento a un helicóptero, le asignamos una luz y la fijamos a él y finalmente importamos un modelo de lámpara para colocarle una luz puntual.



Para asignar el movimiento del helicóptero, tome la estructura que ya existía en el carro y la asigne a el primer parámetro de la traslación de este, de esta forma avanza con las tecla y 'u' retrocede con la tecla 'y':

```
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 luz anclada al helicóptero utilice una spot light que ya estaba en código, únicamente cambie cosas como el color, más abajo inserte el código para anclarla a este y ajuste la dirección de esta:

```
//linterna
spotLights[0] = SpotLight(
    1.0f, 1.0f, 0.0f,    // color amarillo
    0.0f, 2.0f,
    0.0f, 5.0f, 6.0f,    // posición inicial
    0.0f, -1.0f, 0.0f,    // dirección hacia abajo
    1.0f, 0.0f, 0.0f,
    15.0f                // ángulo en grados
);
spotLightCount++;
```

```
// spotlight ligada al helicoptero
glm::vec3 heliPos = glm::vec3(mainWindow.getmuevex(), 5.0f, 6.0f);
glm::vec3 heliDir = glm::normalize(glm::vec3(0.0f, -1.0f, 0.0f)); // apunta al piso
spotLights[0].setFlash(heliPos, heliDir);
```

Finalmente para insertar la lámpara declare el modelo 3D primero, que es un obj y lo llame:

```
Model Kitt_M;
Model Llanta_M;
Model Blackhawk_M;
Model Lampara;
```

```
Kitt_M = Model();
Kitt_M.LoadModel("Models/Cuerpo.obj");
Llanta_M = Model();
Llanta_M.LoadModel("Models/Rueda.obj");
Blackhawk_M = Model();
Blackhawk_M.LoadModel("Models/uh60.obj");
Lampara = Model();
Lampara.LoadModel("Models/lampara.obj");
```

```
//Lampara
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(-5.0f, -1.0f, 6.0));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Lampara.RenderModel();
```

Finalmente para establecer la iluminación en esta, aproveche un point light que ya estaba en mi código, únicamente cambie el color y la ubicación de esta, para que esté acorde a la lámpara:

```
//Declaración de primer luz puntual (LAMPARA)
pointLights[0] = PointLight(1.0f, 1.0f, 1.0f,
    0.0f, 1.0f,
    -5.0f, 2.0f, 4.0f, //Posicion
    0.3f, 0.2f, 0.1f);
pointLightCount++;
```

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

Para esta práctica no tuve problemas, fue más fluido todo, sobre todo porque ya estaba la infraestructura del código, solo fue modificar cosas.

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

La explicación durante la clase fue clara y muy visual, lo que ayudó a entender mejor cómo funcionan los diferentes tipos de luces. No hubo mayores problemas al realizar la práctica, ya que el código base estaba bien estructurado y fue fácil hacer las modificaciones necesarias.

En esta práctica aprendí cómo funcionan los diferentes tipos de luces en la iluminación computacional: la luz direccional, la puntual y la spotlight. La luz direccional solo tiene una dirección y no cambia su intensidad con la distancia. La luz puntual tiene una posición y emite luz en todas direcciones, mientras que la spotlight combina ambas cosas y además permite controlar el ángulo del cono de luz.

También entendí que en la programación se pueden organizar estas luces como clases que heredan unas de otras: primero una clase general "Luz", luego "Luz puntual" que añade más parámetros, y por último "Spotlight" que incluye todos los anteriores más los suyos propios. Esto hace que el código sea más ordenado y fácil de entender.

1. Bibliografía en formato APA

Microsoft. (2022, 20 de octubre). Light types – UWP applications. Microsoft Learn.  
<https://learn.microsoft.com/en-us/windows/uwp/graphics-concepts/light-types>

PlayCanvas Ltd. (s. f.). Lights – Light Types & Shapes. PlayCanvas User Manual.  
Recuperado el 18 de octubre de 2025, de  
<https://developer.playcanvas.com/user-manual/graphics/lighting/lights/>

Stamminger, M. (2021). Basic Lighting. LearnOpenGL.  
<https://learnopengl.com/Lighting/Basic-Lighting>