



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° 08

NOMBRE COMPLETO: Sanchez Calvillo Saida Mayela

N° de Cuenta: 318164481

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 04

SEMESTRE 2025-2

FECHA DE ENTREGA LÍMITE: 12 de abril 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 un Spotlight (que no sea luz de color blanco ni azul) que parta del cofre de su coche y al abrir y cerrar el cofre ilumine en esa dirección.

Para el cofre usamos este código

```
if (key == GLFW_KEY_N)
{
    theWindow->cofre += 10.0;
    if (theWindow->cofre > 45.0f)
    {
        theWindow->cofre = 45.0f;
    }
}
if (key == GLFW_KEY_M)
{
    theWindow->cofre -= 10.0;
    if (theWindow->cofre < 0.0f)
    {
        theWindow->cofre = 0.0f;
    }
}
```

En el reflejamos el limite de apertura del cofre de 0° a 45°. Esta configuración la vamos a mezclar con la siguiente

```
if (key == GLFW_KEY_O && action == GLFW_PRESS)
{
    theWindow->articulacion10 = 2.0;
}
else if (key == GLFW_KEY_O && action == GLFW_RELEASE)
{
    theWindow->articulacion10 = 0.0;
}
if (key == GLFW_KEY_P && action == GLFW_PRESS)
{
    theWindow->articulacion10 = 1.0;
}
else if (key == GLFW_KEY_P && action == GLFW_RELEASE)
{
    theWindow->articulacion10 = 0.0;
}
```

La condición de encendido y apagado de nuestro Spotlight según este abierto o cerrado nuestro cofre.

- Si esta cerrado, no esta encendido
- Si esta abierto, la luz se enciende.

Se ve en esta parte de código:

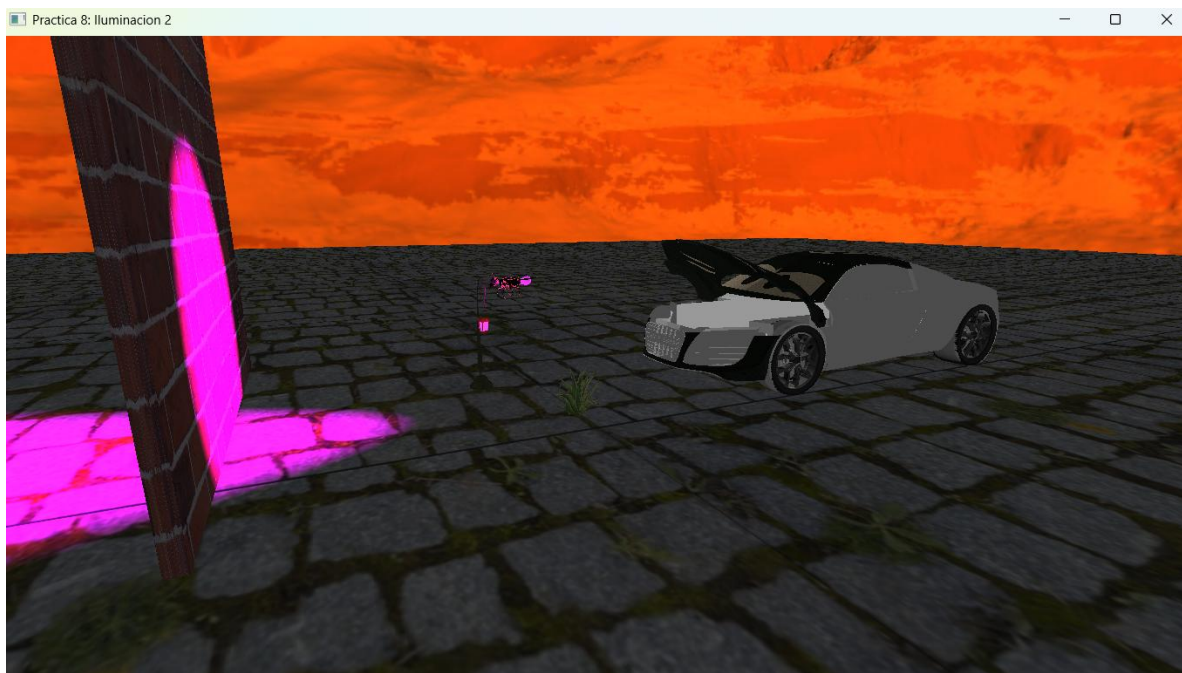
```
if (mainWindow.getarticulacion10() == 0.0f) {  
    if (mainWindow.getcofre() > 0.0) {  
        shaderList[0].SetSpotLights(spotLights2, spotLightCount2);  
    }  
    else {  
        shaderList[0].SetSpotLights(spotLights2, spotLightCount2 - 1);  
    }  
}  
else {  
    if (mainWindow.getarticulacion10() == 1.0)  
        shaderList[0].SetSpotLights(spotLights, spotLightCount);  
    else {  
        if (mainWindow.getarticulacion10() == 2.0f) {  
            shaderList[0].SetSpotLights(spotLights1, spotLightCount1);  
        }  
    }  
}
```

1. **Primera condición** (articulacion10 == 0.0f):

- Si articulacion10 es 0, verifica el valor de cofre:
 - Si cofre > 0.0: Configura todas las luces en spotLights2
 - Si no: Configura todas menos una

2. **Si articulacion10 no es 0:**

- Si articulacion10 == 1.0: Configura las luces de spotLights
- Si articulacion10 == 2.0f: Configura las luces de spotLights1





2. Agregar luz de tipo Spotlight para el coche de tal forma que al avanzar (mover con teclado hacia dirección de X negativa) ilumine con un Spotlight hacia adelante y al retroceder (mover con teclado hacia dirección de X positiva) ilumine con un Spotlight hacia atrás. Son dos Spotlight diferentes que se prenderán y apagarán de acuerdo con alguna bandera asignada por ustedes.

Primero creamos nuestras luces tipo Spotlight en la que cada una la vamos a agregar a un contador diferente para poder acceder a ellas más fácilmente

```
//luz de enfrente
spotLights[0] = SpotLight(1.0f, 1.0f, 0.0f,
    4.0f, 5.0f,
    -5.0f, 10.0f, -10.0f,
    -1.0f, 0.0f, 0.0f,
    1.0f, 0.0f, 0.0f,
    20.0f);
spotLightCount++;

//luz trasera
spotLights1[0] = SpotLight(1.0f, 1.0f, 0.0f,
    4.0f, 5.0f,
    40.0f, 10.0f, -10.0f,
    1.0f, 0.0f, 0.0f,
    1.0f, 0.0f, 0.0f,
    20.0f);
spotLightCount1++;
```


Como necesitamos que este ligada al movimiento del carro, vamos a agregar todas nuestras luces a la instancia del carro y les vamos a incorporar el movimiento del carro. Tal como lo hicimos con el faro en la practica pasada.

```
//Instancia del coche
model = glm::mat4(1.0f);
model = glm::translate(model, glm::vec3(20.0f, -0.9f, -5.0f));
//spotLights[2].SetPos(glm::vec3(0.0f + mainWindow.getmuevex(), -0.5f, -8.0f));
model = glm::translate(model, glm::vec3(0.0f + mainWindow.getmuevex(), -0.5f, -3.0f));
spotLights[0].SetPos(glm::vec3(0.0f + mainWindow.getmuevex(), 7.0f, -8.0f));
spotLights[2][0].SetPos(glm::vec3(0.0f + mainWindow.getmuevex(), 7.0f, -8.0f));
spotLights[1][0].SetPos(glm::vec3(37.0f + mainWindow.getmuevex(), 7.0f, -8.0f));
modelaux = model;
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Carro.RenderModel();
```



3. Agregar otra luz de tipo puntual ligada a un modelo elegido por ustedes (no lámpara) y que puedan prender y apagar de forma independiente con teclado tanto la luz de la lámpara como la luz de este modelo (la luz de la lámpara debe de ser puntual)

Creamos nuestra nueva luz puntual y la agregamos a otro arreglo para no complicarnos al hacer independientes las interacciones con teclado

```
//Luz amarilla (Luciernaga)
pointLights1[0] = PointLight(1.0f, 1.0f, 0.0f,
    0.5f, 1.0f,                               //Radiaci
    -3.3f, 10.0f, -5.0,                        //posici
    0.5f, 0.02f, 0.02f);                      //Ecuac
pointLightCount1++;
```

```
if (mainWindow.getLuciernaga() == true) {
    shaderList[0].SetPointLights(pointLights1, pointLightCount1); //Se apaga la luciernaga
}
else {
    shaderList[0].SetPointLights(pointLights1, pointLightCount1 - 1);
}
```

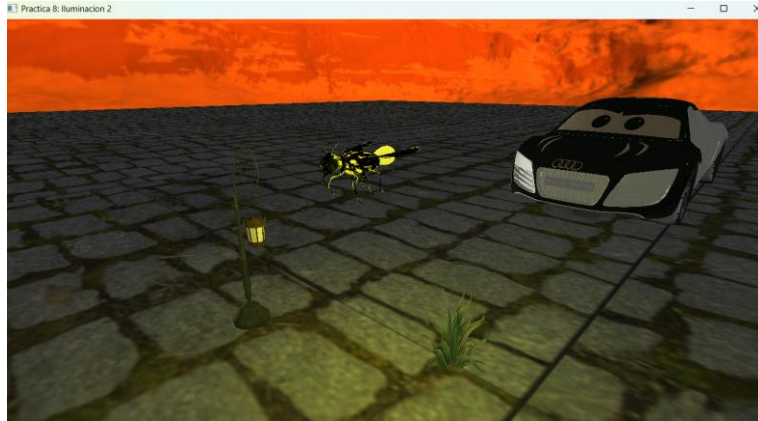
Es tal cual lo hicimos con la lámpara en el ejercicio.

Para esto se movieron las entradas y ahora se ven así

```
if (key == GLFW_KEY_Z)
{
    theWindow->Luciernaga = true;
}
if (key == GLFW_KEY_X)
{
    theWindow->Luciernaga = false;
}
```

Como es una luz puntual, ya no necesita acomodarse dentro de nuestro bucle principal.





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

Mi código de la lampara y de la luciérnaga son exactamente iguales, con la excepción de que están guardados en distintos arreglos para poder usar la condicional que el profesor nos otorgó, pero no funciona y no pude hacer que funcione.

Otro error fue que al ejecutar, mi cofre aparece volteado, pero cuando mueves las teclas que se les asignó ya se acomoda donde debe ir y no se desacomoda.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

La complejidad subió de nivel de repente, pero fue interesante investigar, moverle y ver como podemos hacer muchas formas de interacción con nuestro código.

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

Creo que faltó explicar mucho más las técnicas que pudimos utilizar en la práctica y algunos ejercicios fueron muy ambiguos.

c. Conclusión

Estuvo interesante mezclar técnicas de condicionales para las luces.

4.- Bibliografía

- Sketchfab. (s. f.). *Luciérnaga - Download Free 3D model by Asier (@asvamu)*. <https://sketchfab.com/3d-models/luciernaga-b1abb2eecf634d1eb286dcb6021b87a3>
- OpenAI. (2025). ChatGPT [Large language model]. <https://chatgpt.com>