



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

FACULTAD DE INGENIERÍA  
DIVISIÓN DE INGENIERÍA ELÉCTRICA  
INGENIERÍA EN COMPUTACIÓN



LABORATORIO DE COMPUTACIÓN GRÁFICA e  
INTERACCIÓN HUMANO COMPUTADORA

## EJERCICIOS DE CLASE Nº 4

**NOMBRE COMPLETO:** Gomez Enríquez Agustín

**Nº de Cuenta:** 317031405

**GRUPO DE LABORATORIO:** 3

**GRUPO DE TEORÍA:** 5

**SEMESTRE 2026-1**

**FECHA DE ENTREGA LÍMITE:** 14 – octubre - 2025

**CALIFICACIÓN:** \_\_\_\_\_

## EJERCICIOS DE SESIÓN:

### 1. Actividades realizadas. Una descripción de los ejercicios y capturas de pantalla de bloques de código generados y de ejecución del programa

```
// --- FARO DEL COCHE
spotLights[2] = SpotLight(
    0.0f, 0.0f, 1.0f,           // color (R,G,B)
    5.0f, 1.0f,                // intensidades: ambiental, difusa
    0.0f, 0.0f, 0.0f,          // posición (placeholder, se actualiza cada frame)
    0.0f, -1.0f, 0.0f,         // dirección (placeholder)
    1.0f, 0.022f, 0.0019f,    // atenuación (alcance realista)
    18.0f                      // ángulo del cono (edge, en grados)
);
spotLightCount++;
```

Creamos una lleva luz que será spotlight con el color azul

```
335
336
337     // Seguimiento de la luz del faro del coche
338     glm::mat4 M_car = glm::mat4(1.0f);
339     M_car = glm::translate(M_car, glm::vec3(0.0f + mainWindow.getmuevex(), 0.5f, -3.0f));
340     M_car = glm::scale(M_car, glm::vec3(0.5f));
341     M_car = glm::rotate(M_car, -90.0f * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));

342     glm::vec3 headLocal = glm::vec3(47.0f, 0.0f, -13.0f); // faro delantero izquierdo
343     glm::vec3 headPos = glm::vec3(M_car * glm::vec4(headLocal, 1.0f));

344     // Dirección "hacia delante" del coche en espacio mundo.
345     // Si tu modelo usa +Z como frente, usa (0,0,1); si usa -Z, usa (0,0,-1).
346     glm::vec3 headDir = glm::normalize(glm::vec3(M_car * glm::vec4(0.0f, 0.0f, 1.0f, 0.0f)));

347     // Actualiza posición + dirección del spotlight del faro
348     spotLights[2].SetFlash(headPos, headDir);

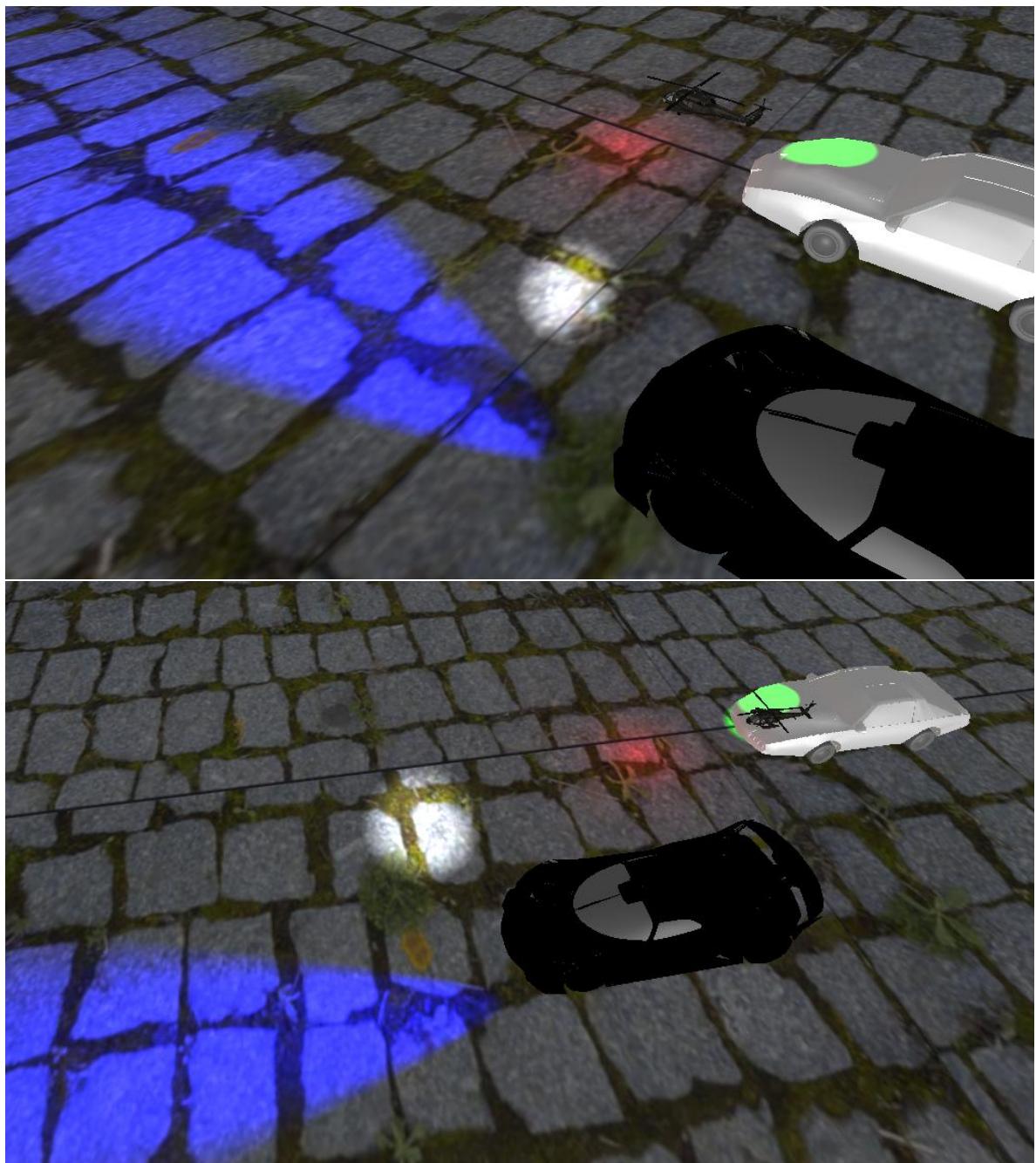
349
350
351
352
353
354     glm::mat4 model(1.0);
355     glm::mat4 modelaux(1.0);
```

Generamos el seguimiento de nuestra luz que va al M\_car que será nuestro carro personalizado. La luz seguirá el desplazamiento que tenga el carro.

```
426
427     //Bugatti
428     model = glm::mat4(1.0);
429     model = glm::translate(model, glm::vec3(12.0f + mainWindow.getmuevex(), -1.0f, 17.0f));
430     modelaux = model;
431     model = glm::scale(model, glm::vec3(4.0f, 4.0f, 4.0f));
432     model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
433     glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
434     Bugatti_M.RenderModel();
435
436     //blending: transparencia o traslucidez
```

Agregamos nuestro modelo del carro junto con su respectivo movimiento

**Ejecución:**



**2. Problemas presentados. Listar si surgieron problemas a la hora de ejecutar el código**

Solo tuve un problema al intentar cargar 3 luces más, honestamente ya no me puse a investigar el porque por el tiempo de entrega, pero me imagino que es por la cantidad de elementos que se están ocupando en el ciclo while.

The screenshot shows a debugger interface with a code editor and a call stack window. The code editor contains C++ code related to rendering a car model. A tooltip from the debugger indicates a memory access violation at address 0xFFFFFFFFFFFFFF. The call stack window shows the stack trace and configuration settings for handling exceptions.

```
glm::mat4 model(1.0);
glm::mat4 modelaux(1.0);
glm::vec3 color = glm::vec3(1.0f, 1.0f, 1.0f);

model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, -1.0f, 0.0f));
model = glm::scale(model, glm::vec3(30.0f, 1.0f, 30.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniform3fv(uniformColor, 1, glm::value_ptr(color));

pisoTexture.UseTexture();
Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);

meshList[2]->RenderMesh(); ✖
```

**Excepción producida**  
Se produjo una excepción: infracción de acceso de lectura.  
std::vector<Mesh \* \_\_ptr64, std::allocator<Mesh \* \_\_ptr64> >::operator[](...)  
devolvió 0xFFFFFFFFFFFFFF.

✖ Analizar con Copilot | Mostrar pila de llamadas | Copiar detalles | Iniciar sesión de Live Share...

✖ Configuración de excepciones  
 Interrumpir cuando se produce este tipo de excepción  
Excepto si se produce en:  
 Practica7x64.exe  
[Abrir configuración de excepciones](#) | [Editar condiciones](#)

### 3. Conclusión:

En esta practica pude comprender el manejo y control de los tipos de luces, tanto en posición, intensidad y funcionamiento en general. Fue un reto hacer que la luz se proyectara en forma de cono y pudiera iluminar el piso. además a esto ya aprendí a usar mi modelo del carro aunque aun le falten detalles.