



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 N° 4

NOMBRE COMPLETO: Gomez Enríquez Agustin

N° 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 nueva luz que será spotlight con el color azul

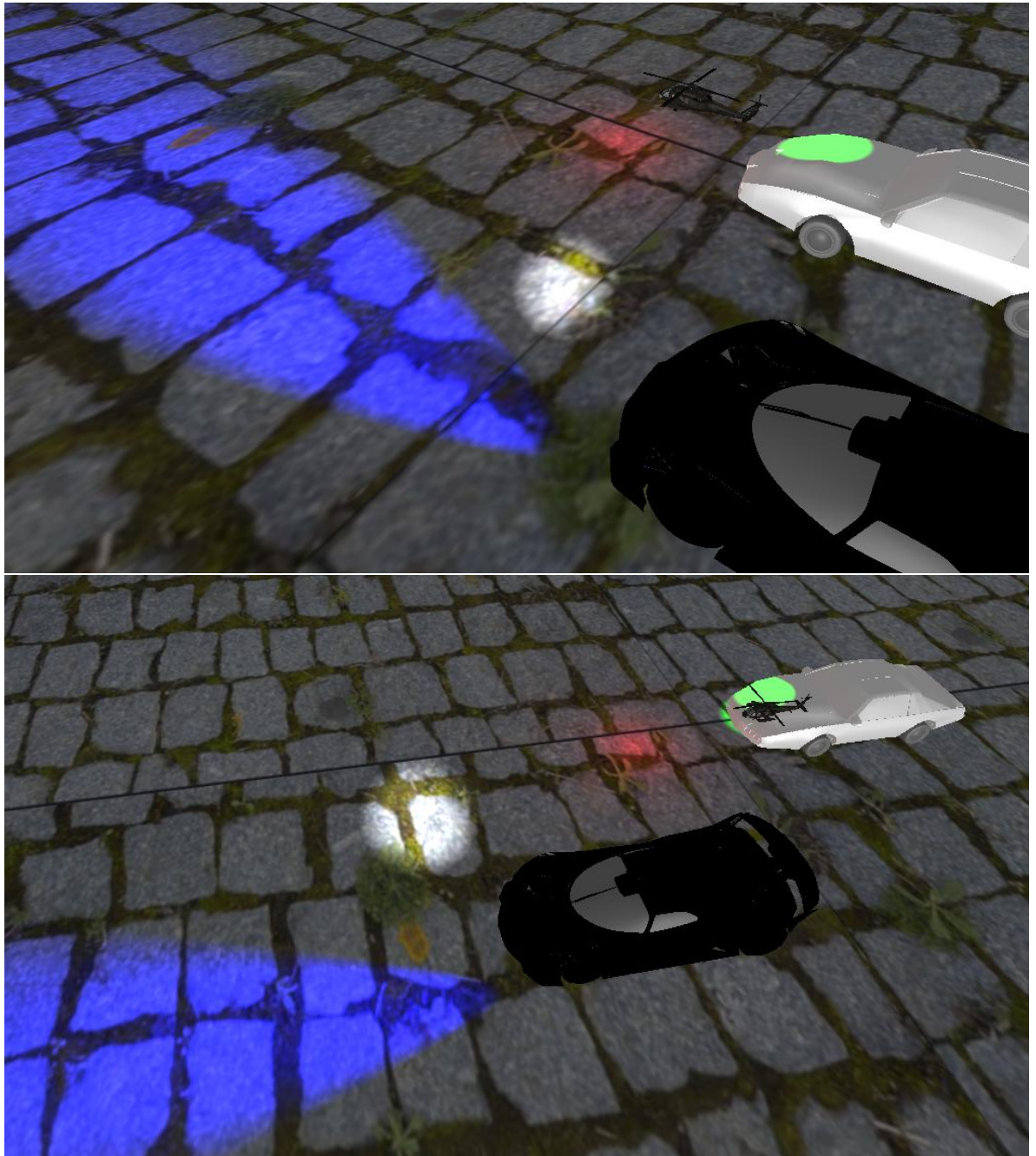
```
335 // Seguimiento de la luz del faro del coche
336 glm::mat4 M_car = glm::mat4(1.0f);
337 M_car = glm::translate(M_car, glm::vec3(0.0f + mainWindow.getmuevex(), 0.5f, -3.0f));
338 M_car = glm::scale(M_car, glm::vec3(0.5f));
339 M_car = glm::rotate(M_car, -90.0f * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
340
341 glm::vec3 headLocal = glm::vec3(47.0f, 0.0f, -13.0f); // faro delantero izquierdo
342 glm::vec3 headPos = glm::vec3(M_car * glm::vec4(headLocal, 1.0f));
343
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
348 // Actualiza posición + dirección del spotlight del faro
349 spotLights[2].setFlash(headPos, headDir);
350
351
352 glm::mat4 model(1.0);
353 glm::mat4 modelaux(1.0);
354
```

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

```
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();

//Instancia del coche
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, -1.0f, 0.0f));
modelaux = model;
model = glm::scale(model, glm::vec3(30.0f, 1.0f, 30.0f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Kitt_M.RenderModel();

//Llanta delantera izquierda
model = modelaux;
model = glm::translate(model, glm::vec3(0.0f, -1.0f, 0.0f));
model = glm::rotate(model, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
```

Excepción producida

Se produjo una excepción: infracción de acceso de lectura.
std::vector<Mesh * __ptr64, std::allocator<Mesh * __ptr64> >::operator[](...) devolvió 0xFFFFFFFFFFFFFFFF.

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 faltan detalles.