



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° 7**

**NOMBRE COMPLETO: BRANDON HERNANDEZ SOLIS**

**N° de Cuenta: 318263113**

**GRUPO DE LABORATORIO: 2**

**GRUPO DE TEORÍA: 6**

**SEMESTRE 2025-1**

**FECHA DE ENTREGA LÍMITE: 05/10/2024**

**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.

Se agrego una nueva función para las letras H y J con el fin de que sean las encargadas de mover el helicoptero, además se agregó una variable para que estas funcionaran

Código:

window.cpp

```
12  Window::Window(GLint windowHeight, GLint windowHeight)
13  {
14      width = windowHeight;
15      height = windowHeight;
16      muevex = 2.0f;
17      muevex2 = 2.0f; //Helicoptero
18      for (size_t i = 0; i < 1024; i++)
19      {
20          keys[i] = 0;
21      }
22  }

118
119      // Movimiento Helicoptero
120      if (key == GLFW_KEY_H)
121      {
122          theWindow->muevex2 += 1.0;
123      }
124      if (key == GLFW_KEY_J)
125      {
126          theWindow->muevex2 -= 1.0;
127      }
128
```

Window.h

```

16     GLfloat getmuevex() { return muevex; }
17     GLfloat getmuevex2() { return muevex2; }    //Helicoptero
18     bool getShouldClose() {
19         return glfwWindowShouldClose(mainWindow);}
20     bool* getsKeys() { return keys; }
21     void swapBuffers() { return glfwSwapBuffers(mainWindow); }
22
23     ~Window();
24 private:
25     GLFWwindow *mainWindow;
26     GLint width, height;
27     bool keys[1024];
28     GLint bufferWidth, bufferHeight;
29     void createCallbacks();
30     GLfloat lastX;
31     GLfloat lastY;
32     GLfloat xChange;
33     GLfloat yChange;
34     GLfloat muevex;
35
36     GLfloat muevex2;    //Helicoptero
37

```

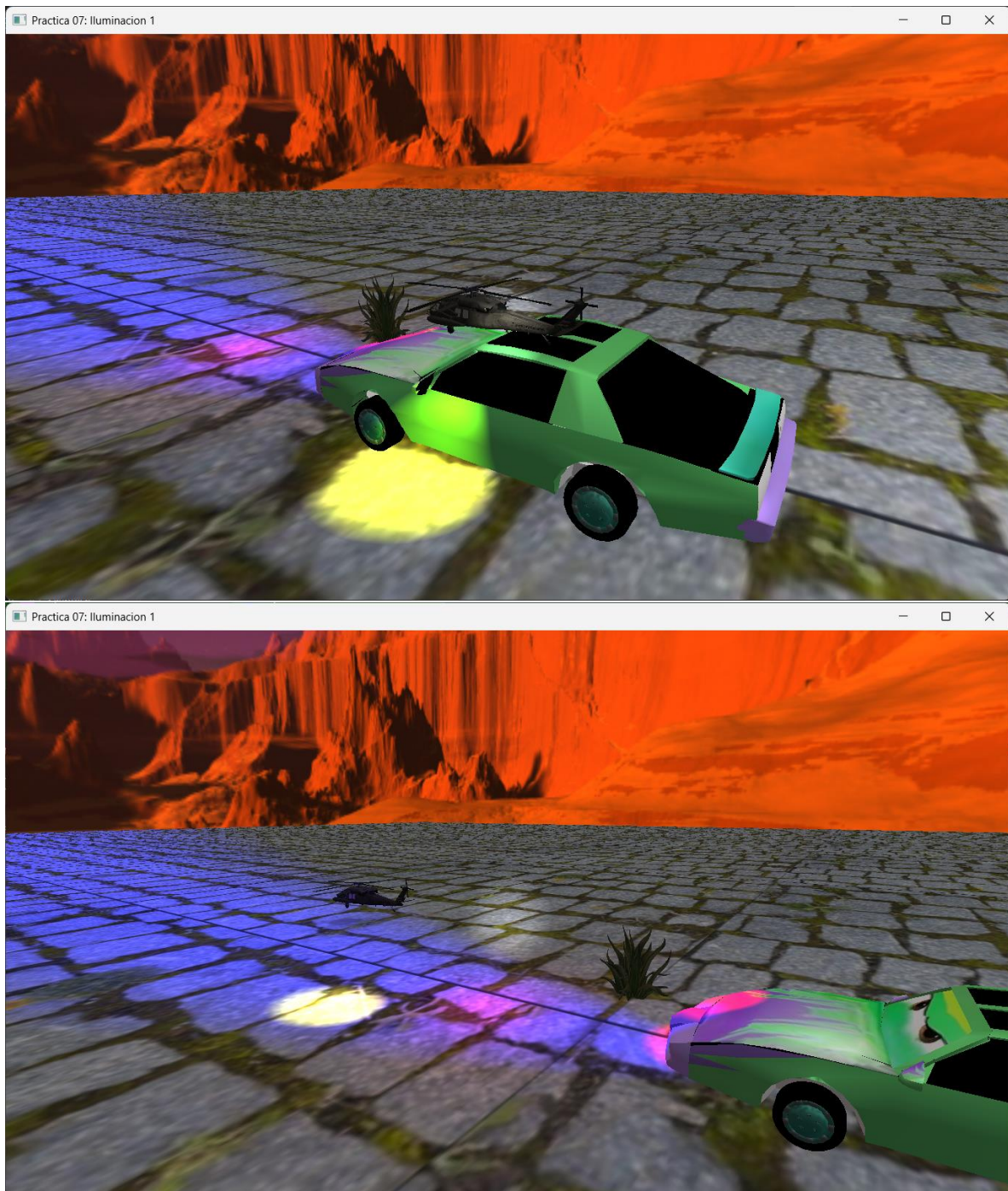
Main:

```

440     // Helicoptero
441
442     modelhel = model;
443     modelhel = glm::mat4(1.0);
444
445     // Movimiento
446     modelhel = glm::translate(modelhel, glm::vec3(mainWindow.getmuevex2() + 0.0f, 0.0f, 0.0f));
447
448     modelhel = glm::translate(modelhel, glm::vec3(0.0f, 5.0f, 6.0f));
449     modelhel = glm::scale(modelhel, glm::vec3(0.3f, 0.3f, 0.3f));
450     modelhel = glm::rotate(modelhel, -90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
451     modelhel = glm::rotate(modelhel, 90 * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
452
453     // Asigno la posición del helicoptero a la luz para que se mueva con él
454     glm::vec3 PosicionLuz = glm::vec3(modelhel[3]) + glm::vec3(0.0f, 0.0f, 0.0f);
455     spotLights[3].SetPos(PosicionLuz); // Actualiza la posición de la luz
456
457     glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(modelhel));
458     Blackhawk_M.RenderModel();
459

```

Resultados:



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

Para este ejercicio solo requerí agregar una nueva fuente de luz y darle las características adecuadas para que apunte al suelo y sea color amarillo, después la ligue a la posición del helicóptero para que lo siga.

Código:

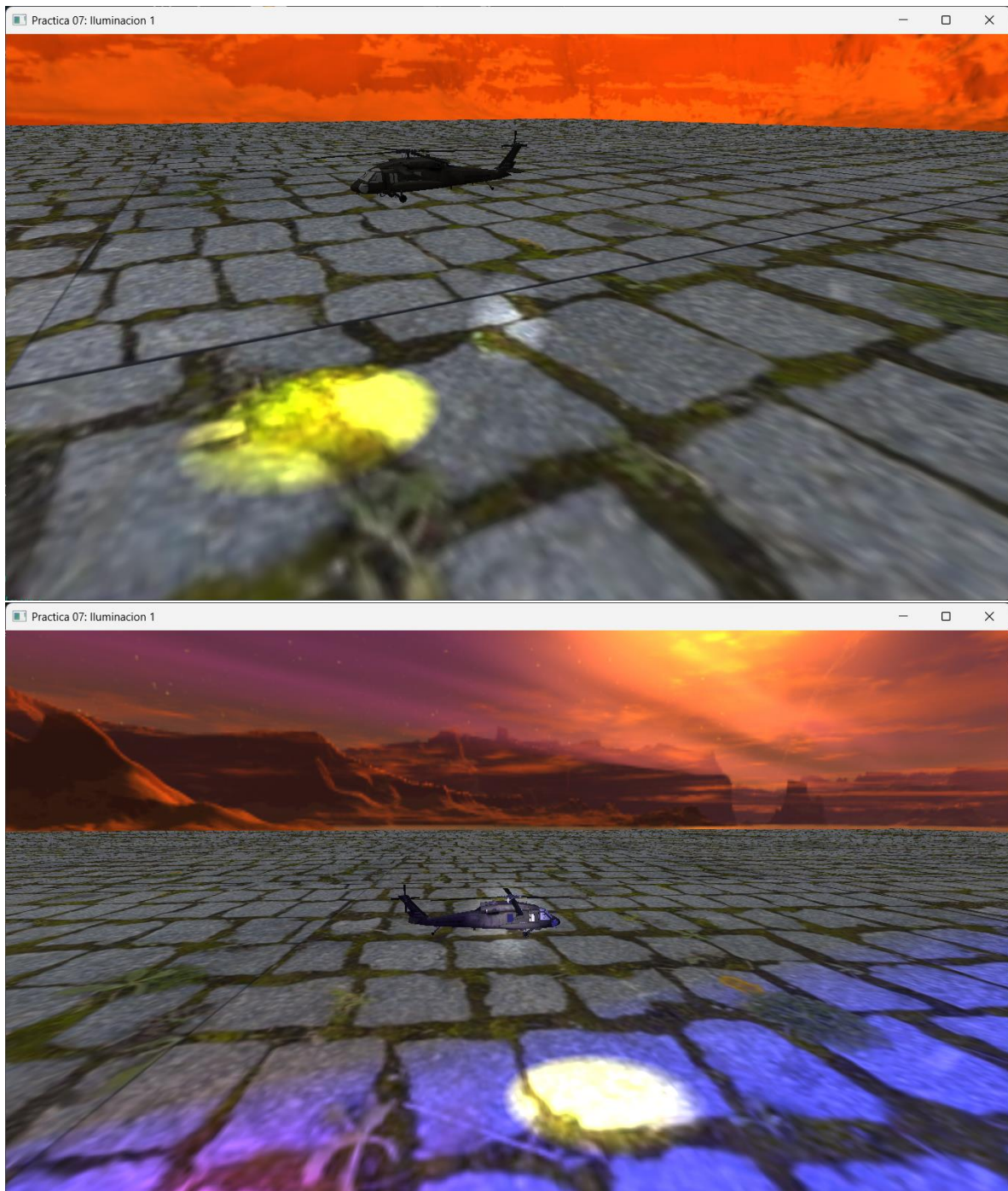
Main:

```
294 //luz Helicoptero
295 spotLights[3] = SpotLight(1.0f, 1.0f, 0.0f, //Color Amarillo
296     1.0f, 2.0f,
297     15.0f, 2.0f, 0.0f, //Posicion inicial
298     -2.0f, -5.0f, 0.0f, //Direccion Ligeramente hacia adelante para parecer realista
299     1.0f, 0.0f, 0.0f,
300     25.0f);
301 spotLightCount++;

452
453 // Asigno la posición del helicoptero a la luz para que se mueva con él
454 glm::vec3 PosicionLuz = glm::vec3(modelhel[3]) + glm::vec3(0.0f, 0.0f, 0.0f);
455 spotLights[3].SetPos(PosicionLuz); // Actualiza la posición de la luz
456
```

Resultados:

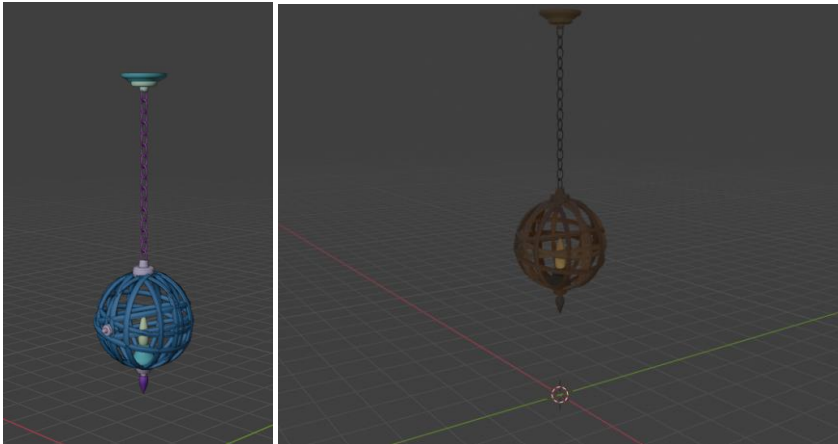




3.- Añadir en el escenario 1 modelo de lámpara texturizada (diferente a los que usarán en su proyecto final) y crearle luz puntual blanca.

Para este ejercicio hubo 3 pasos, primero el modelo en blender, se texturizo y se exporto como tipo .obj. Después se tuvo que importar el modelo y posicionarlo en el escenario donde no estorbara a los otros modelos. Por último, se requirió agregar la luz puntual al centro del modelo para dar el efecto de lampara.

Blender:



Código:

Main:

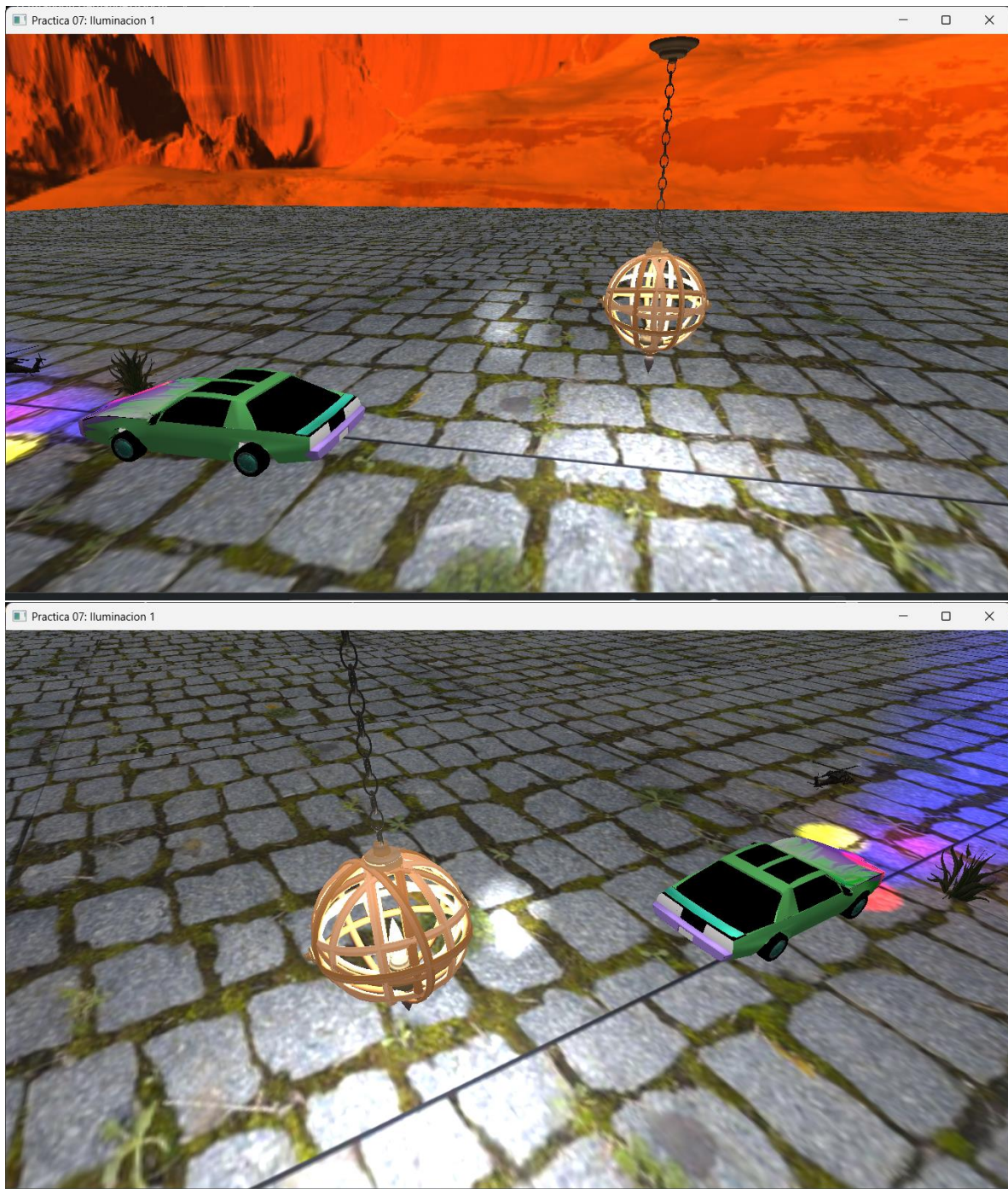
```
223 Blackhawk_M.LoadModel("Models/un60.obj");
224
225 Lampara = Model();
226 Lampara.LoadModel("Models/Lampara.obj");
227
```

```
459
460 // Lampara
461 modelhel = model;
462 modellamp = glm::mat4(1.0);
463
464 // Ajusto posicion
465 modellamp = glm::translate(modellamp, glm::vec3(40.0f, 0.0f, 0.0f));
466 modellamp = glm::scale(modellamp, glm::vec3(0.25f, 0.25f, 0.25f));
467 modellamp = glm::rotate(modellamp, -90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
468
469 // Renderizo
470 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(modellamp));
471 Lampara.RenderModel();
472
473
```

```
258 //Declaración de luz de mi lampara
259 pointLights[1] = PointLight(1.0f, 1.0f, 1.0f, // Color blanco
260 1.0f, 3.0f, // Intensidad alta para que se note
261 40.0f, 10.0f, 0.0f, // Posicion centrada en la lampara
262 0.1f, 0.1f, 0.02f); // Atenuacion
263 pointLightCount++;
264
```

Resultados:





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 hubo problemas

### 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

Los ejercicios de esta práctica englobaron varias cosas que ya habíamos visto anteriormente, desde la jerarquía de los modelos, hasta el movimiento de ellos, también volvimos a usar Blender para texturizar y GIMP para hacer la imagen de la textura optimizada. Las luces que se utilizaron en esta ocasión fueron fáciles de implementar, debido a que solo era agregar luces y adecuarlas para cumplir con cada ejercicio, la parte del movimiento de las luces con los modelos solo consistía en ligar la posición del modelo a la posición de la lámpara, por lo que resulto simple de implementar.

### Bibliografía:

Modelo 3D Lampara: <https://free3d.com/es/modelo-3d/--580269.html>