



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:** Uriarte Ortiz Enrique Yahir

**N° de Cuenta:** 318234757

**GRUPO DE LABORATORIO:** 02

**GRUPO DE TEORÍA:** 04

**SEMESTRE 2025-1**

**FECHA DE ENTREGA LÍMITE:** Sábado 05 de Octubre del 2024

**CALIFICACIÓN:** \_\_\_\_\_

## REPORTE DE PRÁCTICA:

### 1. Ejecución de los ejercicios.

- 1) Agregar movimiento con teclado al helicóptero hacia adelante y atrás.

Para comenzar esta actividad primero genere un nuevo movimiento llamado “**getmuevex2**” en el archivo “**Window.h**”, que usaba a “**muevex2**” declarado dentro del archivo “**Window.cpp**” para asignar el movimiento hacia atrás con la tecla H o hacia adelante con la letra J desde la perspectiva del modelo.

```
GLfloat getXChange();
GLfloat getYChange();
GLfloat getmuevex() { return muevex; }
GLfloat getmuevex2() { return muevex2; }
```

Fig. 1. Definición movimiento para el helicóptero en “Window.h”.

```
GLfloat xChange;
GLfloat yChange;
GLfloat muevex;
GLfloat muevex2;
```

Fig. 2. Definición movimiento para el helicóptero en “Window.h”.

```
Window::Window(GLint windowHeight, GLint windowHeight){
    width = windowHeight;
    height = windowHeight;
    muevex = 2.0f;
    muevex2 = 2.0f;
    for (size_t i = 0; i < 1024; i++){keys[i] = 0;}
}
```

Fig. 3. Definición movimiento para el helicóptero en “Window.cpp”.

```
void Window::ManejaTeclado(GLFWwindow* window, int key, int code, int action, int mode){
    Window* theWindow = static_cast<Window*>(glfwGetWindowUserPointer(window));

    if (key == GLFW_KEY_ESCAPE && action == GLFW_PRESS){glfwSetWindowShouldClose(window, GL_TRUE);}

    if (key == GLFW_KEY_Y){theWindow->muevex += 1.0;}
    if (key == GLFW_KEY_U){theWindow->muevex -= 1.0;}

    if (key == GLFW_KEY_H) { theWindow->muevex2 += 1.0; }
    if (key == GLFW_KEY_J) { theWindow->muevex2 -= 1.0; }
```

Fig. 4. Definición movimiento para el helicóptero en “Window.cpp”.

Después en el código main, en la parte donde se declara el modelo del helicóptero agregue en el “**translate**”, en el eje X, la llamada a “**mainWindow.getmuevex2()**” para hacer que el modelo completo se mueva de acuerdo con las teclas asignadas.

```
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(mainWindow.getmuevex2(), 5.0f, 6.0f));
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();
```

Fig. 5. Declaración para movimiento para el helicóptero.

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

Para esta segunda actividad cree un *spotlight* definido como “*spotlight[2]*” ajustado a los parámetros solicitados, primero le asigne el color amarillo, le asigne una intensidad alta así como difusión, le asigne la dirección de iluminación hacia el eje -Y, decidí que tuviera una difusión y un ángulo de corte de 20 grados para que se vea un poco grande.

```
spotLights[2] = SpotLight( //Foco Helicoptero
    1.0f, 1.0f, 0.0f,
    4.0f, 1.0f,
    0.0f, 0.0f, 0.0f,
    0.0f, -1.0f, 0.0f,
    1.0f, 0.0f, 0.0f,
    20.0f);
spotLightCount++;
```

*Fig. 6. Definición de spotlight para el helicóptero.*

Antes de la declaración para colocar y escalara el modelo del helicóptero declare la posición del *spotlight*, su posición en el plano fue en X un poco mas adelante para que estuviera en una buena posición de la base del helicóptero, un poco mas abajo en el eje de las Y y en el eje de las Z que se encontrara justo a la mitad del modelo. Por último, asigne “*mainWindow.getmuevex2()*” también en el ejes de las X para que este se moviera con el helicóptero.

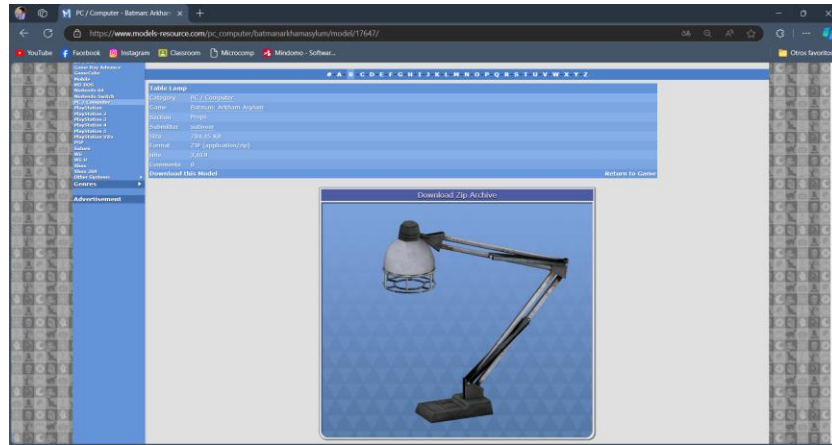
```
spotLights[2].SetPos(glm::vec3(-0.5f + mainWindow.getmuevex2(), 4.85f, 6.0f));

model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(mainWindow.getmuevex2(), 5.0f, 6.0f));
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();
```

*Fig.7. Declaración spotlight para el helicóptero.*

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

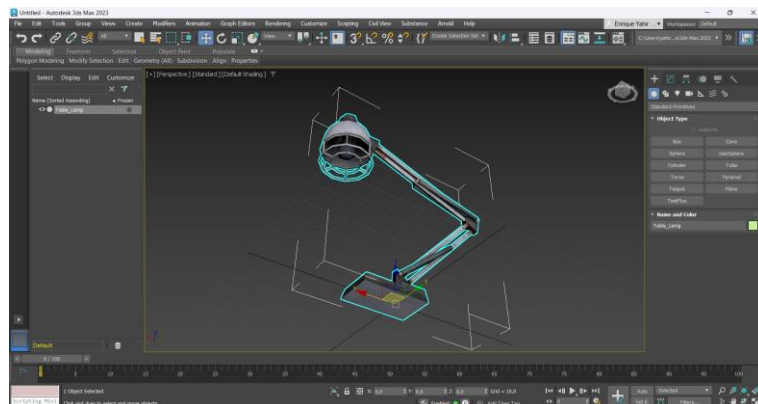
Para esta última actividad escogí este modelo de lampara de escritorio, el enlace del modelo se encuentra en la bibliografía de la práctica, después en 3ds Max ajuste su posición y ubicación del origen de sus ejes, y lo exporté con sus texturas como **“Lampara.obj”**, las texturas del modelo las coloqué tanto en la carpeta **Models** como en la carpeta **Textures**.



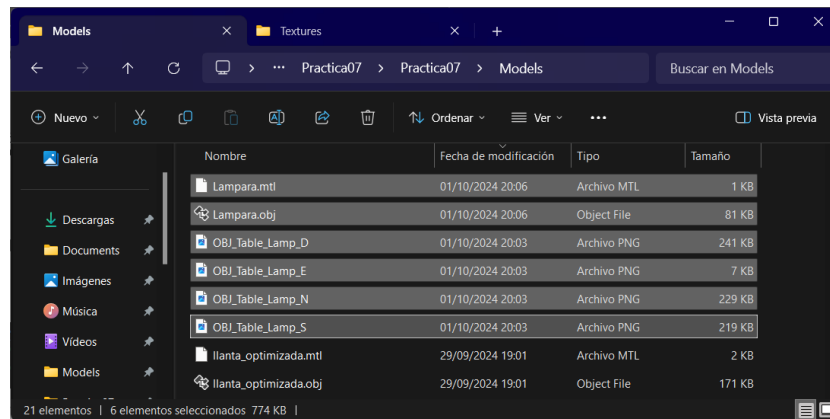
**Fig.8.** Modelo de lampara escogido.



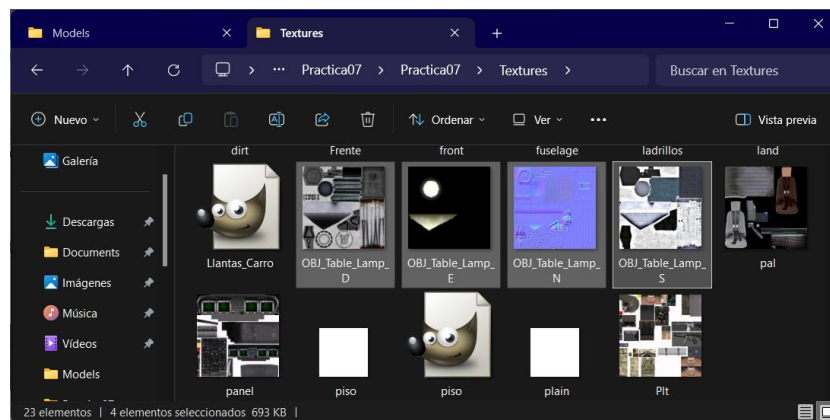
**Fig.9.** Modelo de lampara en 3ds Max.



**Fig.10.** Modelo de lampara en 3ds Max.



**Fig. 11.** Archivos de lampara en .obj en carpeta Models.



**Fig. 12.** Archivos de texturas de lampara en carpeta Textures.

En el archivo main modifique la luz puntual que esta definida desde un inicio en el archivo, ajustándola en el color blanco solicitado, también con una intensidad media y difusa, en cuanto a la posición la ajuste para que se encontrara en el foco del modelo de la lampara.

```
unsigned int pointLightCount = 0; //Luz de Lampara
pointLights[0] = PointLight(
    1.0f, 1.0f, 1.0f,
    1.8f, 1.0f,
    1.45f, 2.62f, -10.0f,
    0.3f, 0.2f, 0.1f);
pointLightCount++;
```

**Fig. 13.** Definición de luz puntual para el helicóptero.

```
glm::vec3 lowerLight = camera.getCameraPosition();
lowerLight.y -= 0.3f;
//spotLights[0].SetFlash(lowerLight, camera.getCameraDirection());
shaderList[0].SetDirectionalLight(&mainLight);

shaderList[0].SetPointLights(pointLights, pointLightCount); //Luz de Lampara
shaderList[0].SetSpotLights(spotLights, spotLightCount);
```

**Fig. 14.** Definición de luz puntual para el helicóptero.

Después de la ejecución del helicóptero coloque las instrucciones para la posición y escala de la lampara, en posición la coloque del lado contrario de la planta del eje



Z, hacia -Z, en cuanto a la escala fue lo que más tuve que disminuir ya que el modelo original era demasiado grande.

```
//Lampara
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0, -1.0f, -10.0f));
model = glm::scale(model, glm::vec3(0.05f, 0.05f, 0.05f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Lampara.RenderModel();
```

**Fig. 15.** Definición de modelo de lampara.

Con todas esas modificaciones, la ejecución del ejercicio fue la siguiente:



**Fig. 16.** Ejecución del ejercicio.



**Fig. 17.** Ejecución del ejercicio.

Como evidencia de la tercera actividad de que la luz fue implementada correctamente cuando la luz esta asignada, el color del foco de la lampara se ve

blanco, dando el efecto de encendido, pero si la luz no esta asignada el color del foco seria vería apagado.



**Fig. 18.** Ejecución del ejercicio.



**Fig. 19.** Ejecución del ejercicio.

Como evidencia para la práctica, realice un video donde se muestra la ejecución, el archivo se llama P07-318234757\_Ejercicios.mp4, adjunto el enlace compartido: <https://drive.google.com/file/d/1OEZAi44q4bIGAFrsRc105RNVCHNFNdOn/view?usp=sharing>

## 2. Problemas presentados.

Para estas 3 actividades no presente problemas, las 3 considero que fueron muy sencillas de realizar, aunque la más laboriosa fue la tercera por el ajuste del modelo en 3ds Max y su exportación en el código.

### 3. *Conclusión:*

Tras realizar los ejercicios de la practica me parece que la complejidad era normal, la implementación de iluminación en los modelos para dar los efectos de que estaban encendidas las luces de estos me pareció interesante de implementar, los tres tipos de iluminación básicos me parecieron acorde para cada situación, la luz direccional para el espacio en general, la luz puntual para dar efecto de encendido a focos, de que la generan para un área, y la luz spotlight para simular una linterna. Al final consideró que esta práctica cumplió su objetivo de introducirnos al uso de iluminación en nuestros modelos, darle un poco más de realismo a estos, siendo esto a mi parecer una buena implementación para el desarrollo del proyecto final.

### 4. *Bibliografía.*

- *PC / Computer - Batman: Arkham Asylum - Table Lamp. The Models*  
*Recuperado de [https://www.models-resource.com/pc\\_computer/batmanarkhamasylum/model/17647/](https://www.models-resource.com/pc_computer/batmanarkhamasylum/model/17647/).*  
*Accedido octubre 1, 2024.*