



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 8

NOMBRE COMPLETO: Gomez Enríquez Agustín

Nº de Cuenta: 317031405

GRUPO DE LABORATORIO: 3

GRUPO DE TEORÍA: 5

SEMESTRE 2026-1

FECHA DE ENTREGA LÍMITE: 21 – octubre - 2025

CALIFICACIÓN: _____

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

```
39     void PointLight::SetPos(glm::vec3 pos)
40     {
41         position = pos;
42     }
```

Trabajándolo en orden como en la clase agregamos un Void Pointlight dentro del pointlight.cpp

```
17
18     ~PointLight();
19     void SetPos(glm::vec3 pos);
20
```

No olvidemos agregar el setpos dentro del pointlight.h

```
15     GLfloat getYChange();
16     GLfloat getmuevex() { return muevex; }
17     GLboolean getprendida() { return luzprendida; }
18     GLfloat getlucespuntuales() { return lucespuntuales; }
19     bool getShouldClose() {
20         return glfwWindowShouldClose(mainWindow); }
21     bool* getsKeys() { return keys; }
22     void swapBuffers() { return glfwSwapBuffers(mainWindow); }

23
24     /*float getHeliMoveZ() const { return heliMoveZ; }
25     void updateHeliMovement(float dt); // integra J/H con delta */
26
27     ~Window();
28
29     private:
30         GLFWwindow *mainWindow;
31         GLint width, height;
32         bool keys[1024];
33         GLint bufferWidth, bufferHeight;
34         void createCallbacks();
35         GLfloat lastX;
36         GLfloat lastY;
37         GLfloat xChange;
38         GLfloat yChange;
39         GLfloat muevex;
40         GLfloat lucespuntuales;
41         GLboolean luzprendida;
42         bool mouseFirstMoved;
43         static void ManejaTeclado(GLFWwindow* window, int key, int co
44         static void ManejaMouse(GLFWwindow* window, double xPos, dou
```

En window.h agregamos tanto el boolean y el float de las luces al igual que lo trabajo el profesor

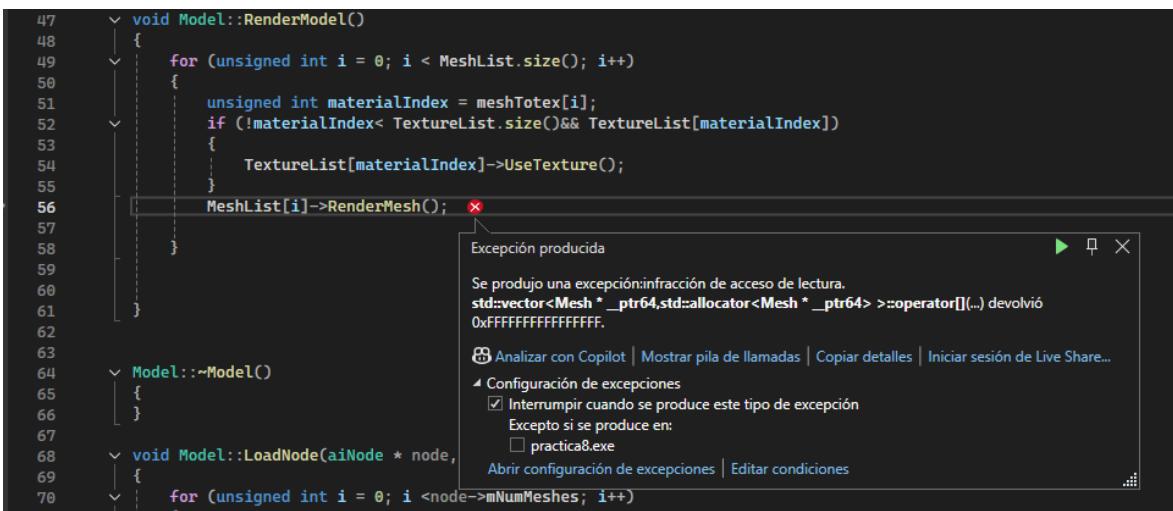
```

111         theWindow-> muevex += 1.0;
112     }
113     if (key == GLFW_KEY_U)
114     {
115         theWindow-> muevex -= 1.0;
116     }
117     if (key == GLFW_KEY_L)
118     {
119         theWindow-> lucespuntuales += 0.0;
120     }
121     if (key == GLFW_KEY_K)
122     {
123         theWindow-> lucespuntuales = 1.0;
124     }
125     if (key == GLFW_KEY_P && action == GLFW_PRESS)
126     {
127         theWindow->luzprendida = !theWindow->luzprendida;
128     }
129
130     if (key >= 0 && key < 1024)

```

En window.cpp agregamos el control por teclado para las luces

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



The screenshot shows a debugger interface with a code editor and a call stack window. The code editor displays a portion of the `Model::RenderModel()` function. A cursor is positioned at the line `MeshList[i]->RenderMesh();`. A tooltip box is open over this line, titled "Excepción producida", containing the message: "Se produjo una excepción: infracción de acceso de lectura. std::vector<Mesh * _ptr64, std::allocator<Mesh * _ptr64>::operator[](...) devolvió 0xFFFFFFFFFFFFFF". The call stack window below shows the stack trace, starting with `Model::~Model()`.

```

47 void Model::RenderModel()
48 {
49     for (unsigned int i = 0; i < MeshList.size(); i++)
50     {
51         unsigned int materialIndex = meshTotex[i];
52         if (!materialIndex < TextureList.size() && TextureList[materialIndex])
53         {
54             TextureList[materialIndex]->UseTexture();
55         }
56         MeshList[i]->RenderMesh(); ✖
57     }
58 }
59
60
61
62
63
64 Model::~Model()
65 {
66 }
67
68 void Model::LoadNode(aiNode * node,
69 {
70     for (unsigned int i = 0; i < node->mNumMeshes; i++)

```

Quizás se trate de un error muy simple, pero tratando de llegar al mismo punto donde el profesor dejó la clase, no he podido por este problema, (tuve otros problemas dentro de los Windows pero eran por ortografía), pero aun corrigiendo esos errores no pude avanzar mas en cuanto a los ejercicios por este detalle que sale en model.cpp (al cual tengo entendido no se le tenía que modificar nada)

3. Conclusión:

En esta práctica me estuve atorando demasiado al no llevar el ritmo de la clase, me fui atrasando y no pude llegar a la parte de los ejercicios asignados. Sin embargo me doy una idea de como se pudo llegar a trabajar:

Ejercicio1:

ANTES:

```
205     //Ejercicio 1: reemplazar con sus datos de 6 caras texturizados, agregar normales
206     // average normals
207     GLfloat cubo_vertices[] = {
208         // front
209         //x      y      z      S      T      NX      NY      NZ
210         -0.5f, -0.5f,  0.5f,  0.26f,  0.34f,  0.0f,   0.0f,  -1.0f, //0
211         0.5f,  -0.5f,  0.5f,  0.49f,  0.34f,  0.0f,   0.0f,  -1.0f, //1
212         0.5f,  0.5f,   0.5f,  0.49f,  0.66f,  0.0f,   0.0f,  -1.0f, //2
213         -0.5f,  0.5f,  0.5f,  0.26f,  0.66f,  0.0f,   0.0f,  -1.0f, //3
214         // right
215         //x      y      z      S      T
216         0.5f, -0.5f,  0.5f,  0.0f,   0.0f,  -1.0f,  0.0f,  0.0f,
217         0.5f, -0.5f, -0.5f, 1.0f,   0.0f,  -1.0f,  0.0f,  0.0f,
218         0.5f,  0.5f, -0.5f, 1.0f,   1.0f,  -1.0f,  0.0f,  0.0f,
219         0.5f,  0.5f,  0.5f,  0.0f,   1.0f,  -1.0f,  0.0f,  0.0f,
220         // back
221         -0.5f, -0.5f, -0.5f, 0.0f,   0.0f,  0.0f,   0.0f,  1.0f,
222         0.5f, -0.5f, -0.5f, 1.0f,   0.0f,  0.0f,   0.0f,  1.0f,
223         0.5f,  0.5f, -0.5f, 1.0f,   1.0f,  0.0f,   0.0f,  1.0f,
224         -0.5f,  0.5f, -0.5f, 0.0f,   1.0f,  0.0f,   0.0f,  1.0f,
225         // left
226         //x      y      z      S      T
227         -0.5f, -0.5f, -0.5f, 0.0f,   0.0f,  1.0f,   0.0f,  0.0f,
228         -0.5f, -0.5f,  0.5f,  1.0f,   0.0f,  1.0f,   0.0f,  0.0f,
229         -0.5f,  0.5f,  0.5f,  1.0f,   1.0f,  1.0f,   0.0f,  0.0f,
230         -0.5f,  0.5f, -0.5f, 0.0f,   1.0f,  1.0f,   0.0f,  0.0f,
231         -0.5f,  0.5f,  0.5f,  0.0f,   1.0f,  1.0f,   0.0f,  0.0f,
232         // bottom
233         //x      y      z      S      T
234         -0.5f, -0.5f,  0.5f,  0.0f,   0.0f,  0.0f,   1.0f,  0.0f,
235         0.5f, -0.5f,  0.5f,  1.0f,   0.0f,  0.0f,   1.0f,  0.0f,
236         0.5f, -0.5f, -0.5f, 1.0f,   1.0f,  0.0f,   1.0f,  0.0f,
237         -0.5f, -0.5f, -0.5f, 0.0f,   1.0f,  0.0f,   1.0f,  0.0f,
238         -0.5f, -0.5f,  0.5f,  0.0f,   1.0f,  0.0f,   1.0f,  0.0f,
239         //UP
240         //x      y      z      S      T
241         -0.5f,  0.5f,  0.5f,  0.0f,   0.0f,  0.0f,   -1.0f, 0.0f,
242         0.5f,  0.5f,  0.5f,  1.0f,   0.0f,  0.0f,   -1.0f, 0.0f,
243         0.5f,  0.5f, -0.5f, 1.0f,   1.0f,  0.0f,   -1.0f, 0.0f,
244         -0.5f,  0.5f, -0.5f, 0.0f,   1.0f,  0.0f,   -1.0f, 0.0f,
245         -0.5f,  0.5f,  0.5f,  0.0f,   1.0f,  0.0f,   -1.0f, 0.0f,
```

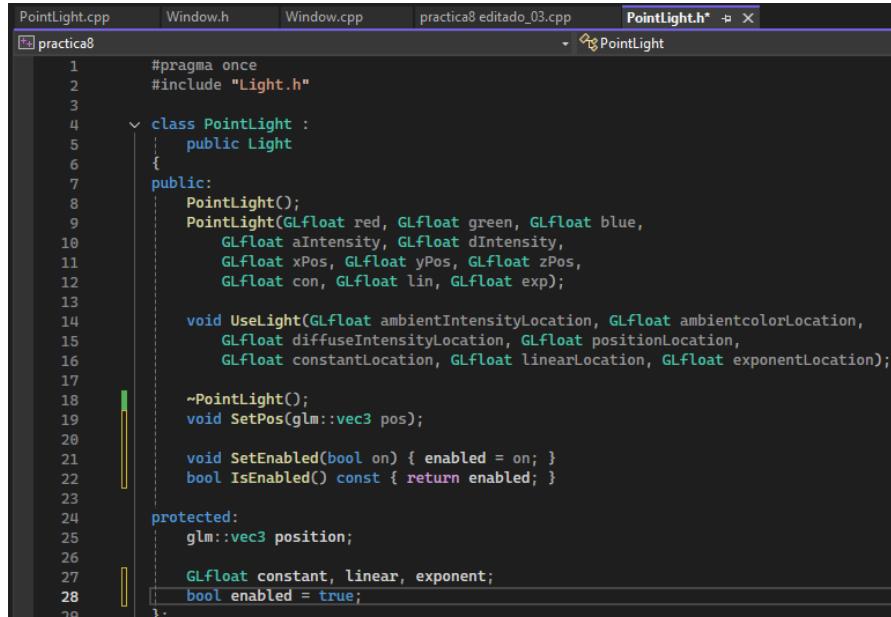
DESPUES:

```

249     GLfloat cubo_vertices[] = {
250     // front (z = +0.5) normal +Z
251     // x      y      z      S      T      NX      NY      NZ
252     -0.5f, -0.5f,  0.5f,  0.26f, 0.34f,   0.0f,  0.0f,  1.0f, //0
253     0.5f, -0.5f,  0.5f,  0.49f, 0.34f,   0.0f,  0.0f,  1.0f, //1
254     0.5f,  0.5f,  0.5f,  0.49f, 0.66f,   0.0f,  0.0f,  1.0f, //2
255     -0.5f,  0.5f,  0.5f,  0.26f, 0.66f,   0.0f,  0.0f,  1.0f, //3
256
257     // right (x = +0.5) normal +X
258     0.5f, -0.5f,  0.5f,  0.0f,  0.0f,   1.0f,  0.0f,  0.0f,
259     0.5f, -0.5f, -0.5f,  1.0f,  0.0f,   1.0f,  0.0f,  0.0f,
260     0.5f,  0.5f, -0.5f,  1.0f,  1.0f,   1.0f,  0.0f,  0.0f,
261     0.5f,  0.5f,  0.5f,  0.0f,  1.0f,   1.0f,  0.0f,  0.0f,
262
263     // back (z = -0.5) normal -Z
264     -0.5f, -0.5f, -0.5f,  0.0f,  0.0f,   0.0f,  0.0f, -1.0f,
265     0.5f, -0.5f, -0.5f,  1.0f,  0.0f,   0.0f,  0.0f, -1.0f,
266     0.5f,  0.5f, -0.5f,  1.0f,  1.0f,   0.0f,  0.0f, -1.0f,
267     -0.5f,  0.5f, -0.5f,  0.0f,  1.0f,   0.0f,  0.0f, -1.0f,
268
269     // left (x = -0.5) normal -X
270     -0.5f, -0.5f, -0.5f,  0.0f,  0.0f,   -1.0f, 0.0f,  0.0f,
271     -0.5f, -0.5f,  0.5f,  1.0f,  0.0f,   -1.0f, 0.0f,  0.0f,
272     -0.5f,  0.5f,  0.5f,  1.0f,  1.0f,   -1.0f, 0.0f,  0.0f,
273     -0.5f,  0.5f, -0.5f,  0.0f,  1.0f,   -1.0f, 0.0f,  0.0f,
274
275     // bottom (y = -0.5) normal -Y
276     -0.5f, -0.5f,  0.5f,  0.0f,  0.0f,   0.0f, -1.0f, 0.0f,
277     0.5f, -0.5f,  0.5f,  1.0f,  0.0f,   0.0f, -1.0f, 0.0f,
278     0.5f, -0.5f, -0.5f,  1.0f,  1.0f,   0.0f, -1.0f, 0.0f,
279     -0.5f, -0.5f, -0.5f,  0.0f,  1.0f,   0.0f, -1.0f, 0.0f,
280
281     // up      (y = +0.5) normal +Y
282     -0.5f,  0.5f,  0.5f,  0.0f,  0.0f,   0.0f,  1.0f,  0.0f,
283     0.5f,  0.5f,  0.5f,  1.0f,  0.0f,   0.0f,  1.0f,  0.0f,
284     0.5f,  0.5f, -0.5f,  1.0f,  1.0f,   0.0f,  1.0f,  0.0f,
285     -0.5f,  0.5f, -0.5f,  0.0f,  1.0f,   0.0f,  1.0f,  0.0f,
286 };
287

```

Ejercicio 2:



```

PointLight.cpp      Window.h      Window.cpp      practica8_editado_03.cpp      PointLight.h*  ✘ PointLight
practica8
1 #pragma once
2 #include "Light.h"
3
4 class PointLight : public Light
5 {
6 public:
7     PointLight();
8     PointLight(GLfloat red, GLfloat green, GLfloat blue,
9                 GLfloat aIntensity, GLfloat dIntensity,
10                GLfloat xPos, GLfloat yPos, GLfloat zPos,
11                GLfloat con, GLfloat lin, GLfloat exp);
12
13     void UseLight(GLfloat ambientIntensityLocation, GLfloat ambientcolorLocation,
14                   GLfloat diffuseIntensityLocation, GLfloat positionLocation,
15                   GLfloat constantLocation, GLfloat linearLocation, GLfloat exponentLocation);
16
17     ~PointLight();
18     void SetPos(glm::vec3 pos);
19
20     void SetEnabled(bool on) { enabled = on; }
21     bool IsEnabled() const { return enabled; }
22
23 protected:
24     glm::vec3 position;
25
26     GLfloat constant, linear, exponent;
27     bool enabled = true;
28

```

Declaramos los estados activos y desactivados para pointlight en pointlight.h

```
PointLight.cpp* X Window.h Window.cpp practica8 editado_03.cpp PointLight.h* pratica8
22     }
23
24     void PointLight::UseLight(GLfloat ambientIntensityLocation, GLfloat ambientcolorLocation,
25                               GLfloat diffuseIntensityLocation, GLfloat positionLocation,
26                               GLfloat constantLocation, GLfloat linearLocation, GLfloat exp
27     {
28         glUniform3f(ambientcolorLocation, color.x, color.y, color.z);
29         glUniform1f(ambientIntensityLocation, ambientIntensity);
30         glUniform1f(diffuseIntensityLocation, diffuseIntensity);
31
32         // Sube color e intensidades de Light
33         Light::UseLight(ambientIntensityLocation, colorLocation, diffuseIntensityLocation);
34
35         // Si está apagada, pisa intensidades a 0 (el último uniform gana)
36         if (!enabled) {
37             glUniform1f(ambientIntensityLocation, 0.0f);
38             glUniform1f(diffuseIntensityLocation, 0.0f);
39     }
```

Para hacerlo notorio alteramos las intensidades de la luz en el cpp

```
Window.cpp*
27
28     GLboolean getLampOn() const { return lampOn; }
29     int     getLampIdx() const { return lampIndex; }
30     void    setLampIdx(int idx) { lampIndex = idx; }
31
32     ~Window();
33     private:
34     GLFWwindow *mainWindow;
35     GLint width, height;
36     bool keys[1024];
37     GLint bufferWidth, bufferHeight;
38     void createCallbacks();
39     GLfloat lastX;
40     GLfloat lastY;
41     GLfloat xChange;
42     GLfloat yChange;
43     GLfloat muevex;
44     GLfloat lucespuntuales;
45     GLboolean luzprendida;
46     bool mouseFirstMoved;
47     static void ManejaTeclado(GLFWwindow* window, int key, int cod
48     static void ManejaMouse(GLFWwindow* window, double xPos, doubl
49
50     GLboolean lampOn = true; // lámpara encendida al iniciar
51     int     lampIndex = 0;
52     /*
53     float heliMoveZ = 0.0f; // acumulador en X realmente
```

En window.h inicializamos nuestros estados de la luz

```
127     theWindow->cuazprendida = !theWindow->cuazprendida
128 }
129 if (key == GLFW_KEY_O && action == GLFW_PRESS) {
130     theWindow->lampOn = !theWindow->lampOn;
131 }
132 }
133 if (key >= 0 && key < 1024)
134 {
135 }
```

Para alternar con la tecla O

Dentro del main ya no sabría como trabajarla porque quería probarlo pero se que en la sección de pointlight va:

```
// Elegir arreglo activo (L / K)
```

```
PointLight* activePL = (mainWindow.getpuntuales() < 0.5f) ? pointLights : pointLights2;  
int activeCnt = pointLightCount;
```