



**Universidad Nacional Autónoma de
México**

Facultad de Ingeniería
División de ingeniería eléctrica -
electrónica
CGeIHC L+ (6590)



Ing. Carlos Aldair Román Balbuena
Semestre 2022-1

**Proyecto Final
Manual Técnico**

Grupo: 04

Aguilar Luna Miguel Angel

Cd. Universitaria a 22 de noviembre del 2021

Objetivo

El alumno deberá aplicar y demostrar los conocimientos adquiridos durante todo el curso.

Diagrama de Gantt

| Actividad | Inicio | Final | 12-oct | 19-oct | 26-oct | 02-nov | 09-nov | 16-nov | 21-nov | 22-nov |
|-------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Imágenes de referencia | 12-oct | 12-oct | | | | | | | | |
| Modelado de los objetos | 19-oct | 19-nov | | | | | | | | |
| Modelado del inmueble | 09-nov | 22-nov | | | | | | | | |
| Animaciones | 21-nov | 21-nov | | | | | | | | |
| Documento escrito | 22-nov | 22-nov | | | | | | | | |

Alcance

Con el desarrollo de este proyecto se planea representar un inmueble con fachada e interior, el cual incluirá distintos objetos que también serán representados en el proyecto.

Para poder llevar a cabo lo antes mencionado utilizaremos herramientas anteriormente implementadas durante el curso, las cuales son: GIMP, MAYA y C++.

Desde GIMP:

Utilizaremos este software libre para manipular las texturas, desde crear las nuestras hasta modificar alguna que consigamos de otra parte.

Desde MAYA:

Utilizaremos este software de especialidad para manipular nuestros objetos, desde modificar su forma hasta texturizarlos, también modificar la ubicación del pivote para hacer las animaciones correctamente.

Desde C++:

En C++ vamos a cargar nuestros objetos para poder manipularlos y desarrollar las animaciones si es que al objeto le corresponde alguna.

Lo ideal será representar el inmueble con todos sus elementos lo más parecido posible a las imágenes de referencia que proporcionamos al profesor.

Limitantes

Las limitantes que encontré en el desarrollo de este proyecto fueron con algunos objetos, ya que tuve que modelarlos y hacer las texturas por completo por mi cuenta, algunas animaciones serán algo difíciles de realizar ya que no son compatibles con el tipo de objetos que presenta. Intentar manejar cada cubo para dar forma a la casa fue complicado e incluso tuve que cambiar de equipo de computo a uno con más capacidad.

Documentación del código

Estas funciones son implementadas para el movimiento de cámara en C++.

```
void KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mode);  
void MouseCallback(GLFWwindow* window, double xPos, double yPos);  
void DoMovement();
```

Estas variables son requeridas para el movimiento de cada uno de los objetos que fueron animados dentro del proyecto.

```
41 float puerta = 0.0f;  
42 float silla = 0.0f;  
43 int bandera = 0.0f;  
44 int bandera1 = 0.0f;  
45 float ventana1 = 0.0f;  
46 float ventana2 = 0.0f;  
47 int bandera2 = 0.0f;  
48 int bandera3 = 0.0f;
```

Estos son los objetos que se dibujan, vemos su dirección en memoria.

```
98 Model CasaCompleta((char*)"Models/Proyecto/CasaCompleta.obj");  
99 Model Puerta((char*)"Models/Proyecto/puertabuena.obj");  
100 Model Silla((char*)"Models/Proyecto/Silla.obj");  
101 Model Ventana1((char*)"Models/Proyecto/ventana1.obj");  
102 Model Ventana2((char*)"Models/Proyecto/ventana2.obj");
```

En estas líneas del código dibujamos todos los objetos utilizados junto con la posición en la que tienen que estar y el movimiento que le corresponde.

```
129     glm::mat4 model(1);
130     glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
131     CasaCompleta.Draw(shader);
132
133     model = glm::mat4(1);
134     model = glm::translate(model, glm::vec3(5.05f, 2.0f, -10.96f));
135     model = glm::rotate(model, glm::radians(puerta), glm::vec3(0.0f, 1.0f, 0.0f));
136     glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
137     Puerta.Draw(shader);
138
139     model = glm::mat4(1);
140     model = glm::translate(model, glm::vec3(6.5f, 1.0f, -1.5f + silla));
141     glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
142     Silla.Draw(shader);
143
144     model = glm::mat4(1);
145     model = glm::translate(model, glm::vec3(-7.9f, 3.0f, -4.96f));
146     model = glm::rotate(model, glm::radians(ventana1), glm::vec3(0.0f, 1.0f, 0.0f));
147     glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
148     Ventana1.Draw(shader);
149
150     model = glm::mat4(1);
151     model = glm::translate(model, glm::vec3(-7.94f, 3.0f, -2.075f));
152     model = glm::rotate(model, glm::radians(ventana2), glm::vec3(0.0f, 1.0f, 0.0f));
153     glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
154     Ventana2.Draw(shader);
```

Teclas desatinadas al movimiento de la cámara

```
167     if (keys[GLFW_KEY_W] || keys[GLFW_KEY_UP]){
168         camera.ProcessKeyboard(FORWARD, deltaTime);
169     }
170
171     if (keys[GLFW_KEY_S] || keys[GLFW_KEY_DOWN]){
172         camera.ProcessKeyboard(BACKWARD, deltaTime);
173     }
174
175     if (keys[GLFW_KEY_A] || keys[GLFW_KEY_LEFT]){
176         camera.ProcessKeyboard(LEFT, deltaTime);
177     }
178
179     if (keys[GLFW_KEY_D] || keys[GLFW_KEY_RIGHT]){
180         camera.ProcessKeyboard(RIGHT, deltaTime);
181     }
```

Teclas para el movimiento de la puerta y la silla

```
if (keys[GLFW_KEY_P]){
    if (puerta > -90.0f && bandera == 0) {
        puerta -= 1.0f;
        if (puerta == -90.0f) {
            bandera = 1;
        }
    }
    else if (puerta < 0.0f && bandera == 1) {
        puerta += 1.5f;
        if (puerta == 0.0f) {
            bandera = 0;
        }
    }
}
if (keys[GLFW_KEY_C]){
    if (silla <= 1.0f && bandera1 == 0) {
        silla += 0.1f;
    }
}
```

Teclas destinadas al movimiento de las dos ventanas

```
202 | if (keys[GLFW_KEY_V]){ //Movimiento de la Ventana 1
203 |     if (ventana1 > -180.0f && bandera2 == 0) {
204 |         ventana1 -= 1.0f;
205 |         if (ventana1 == -180.0f) {
206 |             bandera2 = 1;
207 |         }
208 |     }else if (ventana1 < 0.0f && bandera2 == 1) {
209 |         ventana1 += 1.5f;
210 |         if (ventana1 == 0.0f) {
211 |             bandera2 = 0;
212 |         }
213 |     }
214 | }
215 |
216 | if (keys[GLFW_KEY_B]){ //Movimiento de la Ventana 2
217 |     if (ventana2 < 180.0f && bandera3 == 0) {
218 |         ventana2 += 1.0f;
219 |         if (ventana2 == 180.0f) {
220 |             bandera3 = 1;
221 |         }
222 |     }else if (ventana2 > 0.0f && bandera3 == 1) {
223 |         ventana2 -= 1.5f;
224 |         if (ventana2 == 0.0f) {
225 |             bandera3 = 0;
226 |         }
227 |     }
228 | }
```