



**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 05 de diciembre del 2021

# Objetivo

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

## Diagrama de Gantt

Actividad	Inicio	Final	16-nov	19-nov	20-nov	21-nov	22-nov	25-nov	30-nov	01-dic	06-dic
Imágenes de referencia	16-nov	16-nov									
Modelado de los objetos	16-nov	19-nov									
Modelado del inmueble	20-nov	22-nov									
Animaciones	25-nov	30-nov									
Documento escrito	01-dic	06-dic									

## 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.

```
39  GLfloat deltaTime = 0.0f;
40  GLfloat lastFrame = 0.0f;
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;
49  float Cofretapa = 0.0f;
50  int bandera4 = 0.0f;
51  float BarrilTapa = 0.0f;
52  int bandera5 = 0.0f;
```

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

```
102  // Load models
103  Model CasaCompleta((char*)"Models/Proyecto/CasaFinal.obj");
104  Model Puerta((char*)"Models/Proyecto/puertabuena.obj");
105  Model Silla((char*)"Models/Proyecto/Silla.obj");
106  Model Ventana1((char*)"Models/Proyecto/ventana1.obj");
107  Model Ventana2((char*)"Models/Proyecto/ventana2.obj");
108  Model CofreTapa((char*)"Models/Proyecto/CofretapaFinal.obj");
109  Model BarrilTapa((char*)"Models/Proyecto/BarrilTapa.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.

```
134 // Draw the loaded model
135 glm::mat4 model(1);
136 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
137 CasaCompleta.Draw(shader);
138
139
140 model = glm::mat4(1);
141 model = glm::translate(model, glm::vec3(5.05f, 2.0f, -10.96f));
142 model = glm::rotate(model, glm::radians(puerta), glm::vec3(0.0f, 1.0f, 0.0f));
143 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
144 Puerta.Draw(shader);
145
146 model = glm::mat4(1);
147 model = glm::translate(model, glm::vec3(1.99f, 6.7f, -10.00f));
148 model = glm::rotate(model, glm::radians(Cofretapa), glm::vec3(1.0f, 0.0f, 0.0f));
149 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
150 CofreTapa.Draw(shader);
151
152 model = glm::mat4(1);
153 model = glm::translate(model, glm::vec3(6.5f, 1.0f, -1.5f + silla));
154 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
155 Silla.Draw(shader);
156
157 model = glm::mat4(1);
158 model = glm::translate(model, glm::vec3(4.5f, 6.55f + BarrilTapa, 5.05f));
159 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
160 BarrilTapa.Draw(shader);
161
162 model = glm::mat4(1);
163 model = glm::translate(model, glm::vec3(-7.9f, 3.0f, -4.96f));
164 model = glm::rotate(model, glm::radians(ventana1), glm::vec3(0.0f, 1.0f, 0.0f));
165 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
166 Ventana1.Draw(shader);
167
168 model = glm::mat4(1);
169 model = glm::translate(model, glm::vec3(-7.94f, 3.0f, -2.075f));
170 model = glm::rotate(model, glm::radians(ventana2), glm::vec3(0.0f, 1.0f, 0.0f));
171 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
172 Ventana2.Draw(shader);
173
```

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     }
209     else if (ventana1 < 0.0f && bandera2 == 1) {
210         ventana1 += 1.5f;
211         if (ventana1 == 0.0f) {
212             bandera2 = 0;
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     }
223     else if (ventana2 > 0.0f && bandera3 == 1) {
224         ventana2 -= 1.5f;
225         if (ventana2 == 0.0f) {
226             bandera3 = 0;
227         }
228     }
229 }
```

Teclas destinadas al movimiento de las tapas del cofre y barril

```
247 if (keys[GLFW_KEY_T]) { //Movimiento de la Tapa del Cofre
248     if (Cofretapa > -90.0f && bandera4 == 0) {
249         Cofretapa -= 1.0f;
250         if (Cofretapa == -90.0f) {
251             bandera4 = 1;
252         }
253     }
254     else if (Cofretapa < 0.0f && bandera4 == 1) {
255         Cofretapa += 1.5f;
256         if (Cofretapa == 0.0f) {
257             bandera4 = 0;
258         }
259     }
260 }
261 if (keys[GLFW_KEY_K]) { //Movimiento de la Tapa del Barril
262     if (BarrilTapa <= 1.0f && bandera1 == 0) {
263         BarrilTapa += 0.1f;
264     }
265 }
266 }
```