



**Universidad Nacional Autónoma de
México**
Facultad de Ingeniería
División de ingeniería eléctrica -
electrónica
CGeIHC (1590)



Ing. Carlos Aldair Román Balbuena
Semestre 2022-1

**Proyecto Final
Manual Técnico**

Grupo: 04

Osorio Robles Sergio de Jesús

Cd. Universitaria a 9 de diciembre del 2021

Objetivo

El alumno deberá aplicar y demostrar los conocimientos adquiridos durante todo el curso. Así como sus aptitudes de autodidacta para conocer el entorno de modelado 3D con el software Autodesk Maya.

Diagrama de Gantt

Actividad	Inicio	Final	12-oct	19-oct	26-oct	27-oct	13-nov	14-nov	15-nov	16-nov	17-nov	18-nov	19-nov	20-nov	21-nov	22-nov	01-dic	02-dic	03-dic	04-dic	05-dic	06-dic	07-dic	09-dic
Imágenes de referencia	12-oct	16-nov																						
Modelado de los objetos	19-oct	06-dic																						
Modelado de la casa	24-oct	03-dic																						
Animaciones	21-nov	07-dic																						
Documento escrito	22-nov	08-dic																						
Entrega	09-dic	09-dic																						

Alcance

Desarrollar una réplica de la fachada y del interior tomando como línea base la imagen de referencia de la caricatura “Un show más”. La misma deberá incluir distintos objetos animados y que hayan sido modelados en 3D con un software de modelado.

Las demás herramientas utilizadas para realizar este proyecto fueron GIMP para la manipulación de imágenes, Autodesk Maya 2020 para la manipulación, texturizado y modelado de objetos 3D y la IDE Visual Studio Code 2019 para la compilación y depuración del código de animación en el lenguaje de programación C++.

Alcance de GIMP: Utilizar este software para manipular el tamaño y el diseño de las texturas utilizadas.

Alcance de Maya: Manipulación de los objetos 3D para asemejar a los objetos reales, uso de las texturas creadas en GIMP para aplicarlas a los objetos.

Alcance de C++: Desde C++ con Visual Studio Code 2019 crearemos la cámara para visualizar el escenario creado en Maya, también aplicar los shaders para la iluminación y visualización de las texturas. También crearemos las animaciones desde este espacio y la versión final será compilada usando Visual Studio.

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

Limitantes

El hardware utilizado para modelar fue suficiente para recrear el espacio y la fachada. Conforme la complejidad de las divisiones de los polígonos fue aumentando el tiempo de procesamiento fue aumentando y llegando a tomar más de un minuto para visualizar el resultado del modelado en C++. Conforme se fueron añadiendo más polígonos con más divisiones y más texturas Maya fue tomando más tiempo para completar la tarea deseada.

El ancho de banda necesario para subir todos los archivos para generar el proyecto terminado fue insuficiente y tomó más del tiempo estimado.

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.

```
//Variables de animacion  
float rot = 0.0f; //rotación de la puerta de entrada  
int bandera = 1; //Bandera para el sentido de la puerta de entrada  
int bandera_puertas = 1; //Bandera para puertas de la alacena  
float rot_puertas = 0.0f; //rotacion de las puertas de la alacena  
//Variable para mover las animaciones automaticamente  
bool activanim = false;  
bool activanim_puertas = false;  
bool activanim_cassette = false;  
bool cassette_dentro = false;  
bool anim = false;  
  
//Variables incremento de las animaciones  
float puertas = 0.0f;  
float compuerta = 0.0f;
```

Estas variables fueron utilizadas para el movimiento de rotación de las puertas, en la línea 327 en adelante para mover la puerta de entrada.

Utilizamos las siguientes transformaciones para ubicar a los objetos en su posición.

```
324
325     model = glm::mat4(1);
326     model = glm::translate(model, glm::vec3(4.5f, 10.0f, 15.0f));
327     model = glm::rotate(model, glm::radians(rot), glm::vec3(0.0f, -1.0f, 0.0f));
328     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
329     puerta_entrada.Draw(lightningShader);
330
331     model = glm::mat4(1);
332     model = glm::translate(model, glm::vec3(9.0f, 7.3f, 26.0f));
333     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
334     cassette.Draw(lightningShader);
335
336
337     model = glm::mat4(1);
338     model = glm::translate(model, glm::vec3(14.09f, 7.24f, 34.65f));
339     model = glm::rotate(model, glm::radians(rot_puertas), glm::vec3(0.0f, -1.0f, 0.0f));
340     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
341     puerta_alacenaA.Draw(lightningShader);
342
343     model = glm::mat4(1);
344     model = glm::translate(model, glm::vec3(11.96f, 7.24f, 34.65f));
345     model = glm::rotate(model, glm::radians(rot_puertas), glm::vec3(0.0f, 1.0f, 0.0f));
346     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
347     puerta_alacenaB.Draw(lightningShader);
348
349     model = glm::mat4(1);
350     model = glm::translate(model, glm::vec3(11.57f, 7.24f, 34.65f));
351     model = glm::rotate(model, glm::radians(rot_puertas), glm::vec3(0.0f, -1.0f, 0.0f));
352     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
353     puerta_alacenaC.Draw(lightningShader);
354
355     model = glm::mat4(1);
356     model = glm::translate(model, glm::vec3(9.45f, 7.24f, 34.65f));
357     model = glm::rotate(model, glm::radians(rot_puertas), glm::vec3(0.0f, 1.0f, 0.0f));
358     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
359     puerta_alacenaD.Draw(lightningShader);
360
361     model = glm::mat4(1);
362     model = glm::translate(model, glm::vec3(14.50, 7.0f, 25.85f));
363     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
364     cassette_compuerta.Draw(lightningShader);
365
366     model = glm::mat4(1);
367     model = glm::translate(model, glm::vec3(3.13f, 14.30f, 23.85f));
368     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
369     lampara.Draw(lightningShader);
```

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 de entrada, las puertas de la alacena y la iluminación:

```
491
492 if (keys[GLFW_KEY_L])//Animacion de las luces
493 {
494     if (!anim)
495     {
496         //
497         anim = true;
498     }
499     else if (anim)
500     {
501         anim = false;
502     }
503 }
504
505 if (keys[GLFW_KEY_O])//Animacion puerta de entrada
506 {
507     if (activanim == false) {
508         activanim = true;
509     }
510     else if (activanim == true) {
511         activanim = false;
512     }
513 }
514
515 if (keys[GLFW_KEY_P])//Animacion de las puertas de la alacena
516 {
517     if (activanim_puertas == false) {
518         activanim_puertas = true;
519     }
520     else if (activanim_puertas == true) {
521         activanim_puertas = false;
522     }
523 }
524
```