

Universidad Nacional Autónoma de México
Facultad de Ingeniería

Computación Gráfica e Interacción Humano-Computadora

PROYECTO FINAL

Grupo 04

Alumno: Rosales Mendoza Armando José

Prof. Ing. Carlos Aldair Román Balbuena

México, Ciudad de México. 22 de noviembre del 2021

Objetivo

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

Descripción

El alumno deberá seleccionar una fachada y un espacio que pueden ser reales o ficticios y presentar imágenes de referencia de dichos espacios para su recreación 3D en OpenGL.

En la imagen de referencia se debe visualizar 7 objetos que el alumno va a recrear virtualmente y donde dichos objetos deben ser lo más parecido a su imagen de referencia, así como su ambientación.

Propuesta

La propuesta que se realizó fue una cabaña con temática navideña, se buscaron referencias y las imágenes se encuentran a continuación



Figura 1. Fachada de la casa



Figura 2. Interior de la casa

También se escogieron 7 objetos para modelar, los cuales se muestra a continuación

Árbol de Navidad



Figura 3. Árbol en maya

Bota navideña

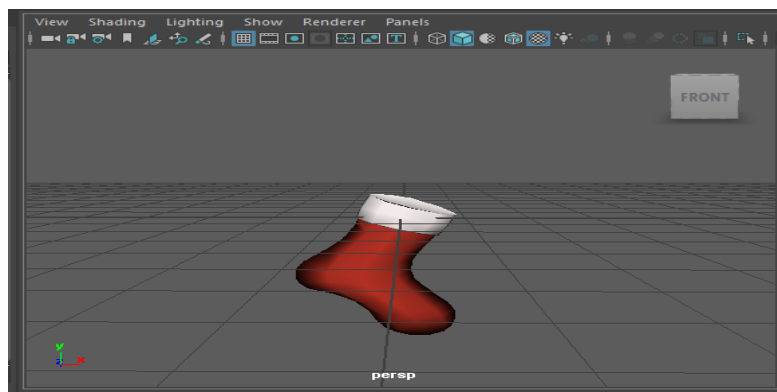


Figura 4. Bota modelada

Campanas

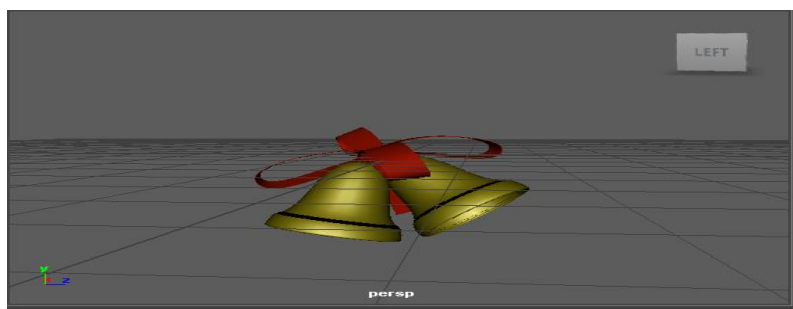


Figura 5. Campanas navideñas en maya

Chimenea



Figura 6. Chimenea

Regalos

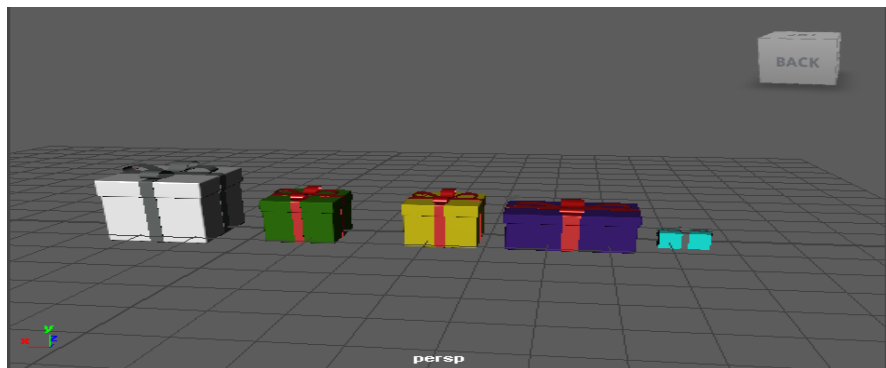


Figura 7. Regalos en maya

Santa



Figura 8. Santa Claus en maya

Sofá

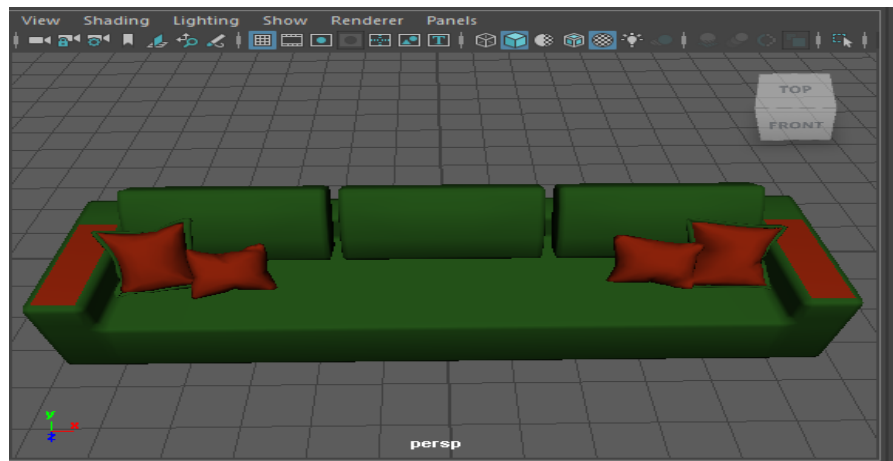


Figura 9. Sofá en maya

Fachada



Figura 10. Fachada en maya

Limitantes

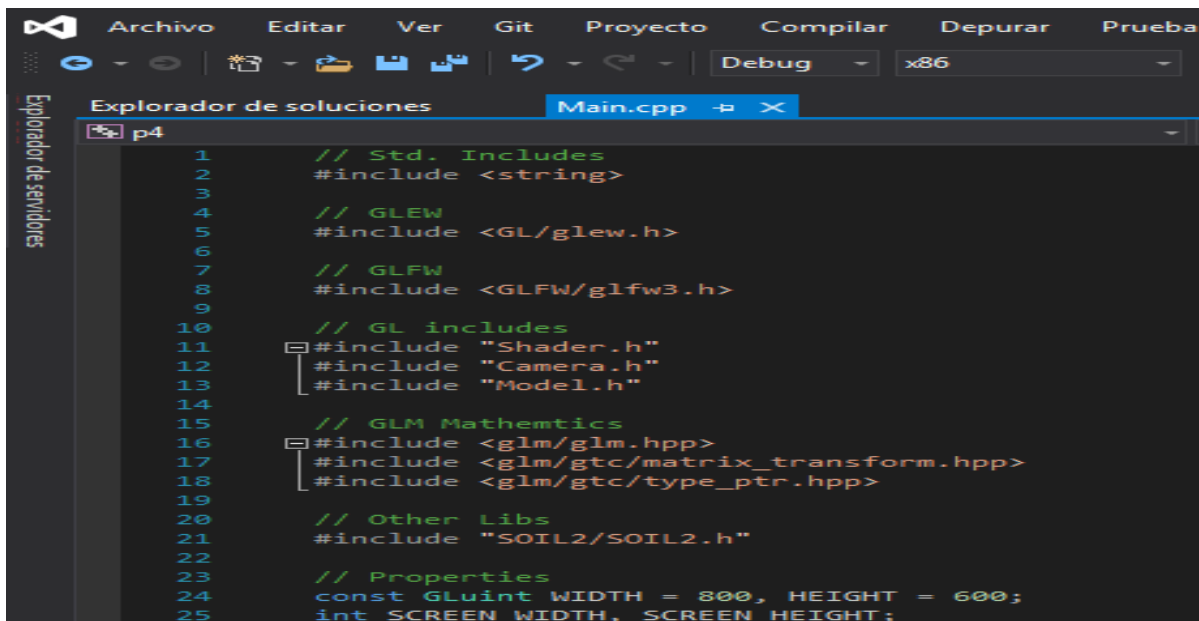
los problemas que se tuvieron al desarrollar el proyecto fueron las características de hardware de mi computadora ya que no permitían trabajar con velocidad y eficiencia además que no permiten mostrar mejores graficas.

Diagrama de gant

Fecha (semanas) actividades	2 octubre	3 octubre	4 octubre	1 noviembre	2 noviembre	3 noviembre
Inicio	x					
Buscar imágenes		x				
modelar			x			
Cargar				x		
cambios					x	x
animar					x	x
entrega						x

Documentación

En estas primeras líneas de código se definen los encabezados que se ocuparan más adelante



```
1 // Std. Includes
2 #include <string>
3
4 // GLEW
5 #include <GL/glew.h>
6
7 // GLFW
8 #include <GLFW/glfw3.h>
9
10 // GL includes
11 #include "Shader.h"
12 #include "Camera.h"
13 #include "Model.h"
14
15 // GLM Mathematics
16 #include <glm/glm.hpp>
17 #include <glm/gtc/matrix_transform.hpp>
18 #include <glm/gtc/type_ptr.hpp>
19
20 // Other Libs
21 #include "SOIL2/SOIL2.h"
22
23 // Properties
24 const GLuint WIDTH = 800, HEIGHT = 600;
25 int SCREEN_WIDTH, SCREEN_HEIGHT;
```

Figura 11. Código 1

En esta siguiente sección encontramos la construcción de la ventana la cual será de 800 x 600. También encontramos la cámara que se utiliza para movernos en el recorrido.

```

// Properties
const GLuint WIDTH = 800, HEIGHT = 600;
int SCREEN_WIDTH, SCREEN_HEIGHT;

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

// Camera
Camera camera( glm::vec3( 10.0f, 0.0f, 0.0f ) );
bool keys[1024];
GLfloat lastX = 400, lastY = 300;
bool firstMouse = true;

GLfloat deltaTime = 0.0f;
GLfloat lastFrame = 0.0f;

```

Figura 12. Código 2

A partir de la línea 42 y hasta la línea 47 encontramos las variables que se ocupan para poder animar los objetos. Además, en esta sección se encuentra el principio de Main y el manejo de la versión, el cual esta comentado por que las características de nuestro hardware son menores a 3.

```

42 //variables de animacion
43 float trs = 0.0f;
44 float rot = 0.0f;
45 float rotCD = 0.0f;
46 float rotCI = 0.0f;
47 float rotSofa= 0.0f;
48
49 float tr = 4.0;
50
51 int main( )
52 {
53     // Init GLFW
54     glfwInit( );
55     // Set all the required options for GLFW
56     /* glfwWindowHint( GLFW_CONTEXT_VERSION_MAJOR, 3 );
57     glfwWindowHint( GLFW_CONTEXT_VERSION_MINOR, 3 );
58     glfwWindowHint( GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE );
59     glfwWindowHint( GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE );*/
60     glfwWindowHint( GLFW_RESIZABLE, GL_FALSE );
61

```

Figura 13. Código 3

En esta siguiente sección seguimos con código que crea la ventana

```

55 // Set all the required options for GLFW
56 /* glfwWindowHint( GLFW_CONTEXT_VERSION_MAJOR, 3 );
57 glfwWindowHint( GLFW_CONTEXT_VERSION_MINOR, 3 );
58 glfwWindowHint( GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE );
59 glfwWindowHint( GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE );*/
60 glfwWindowHint( GLFW_RESIZABLE, GL_FALSE );
61
62 // Create a GLFWwindow object that we can use for GLFW's functions
63 GLFWwindow *window = glfwCreateWindow( WIDTH, HEIGHT, "Proyecto", nullptr, nullptr );
64
65 if ( nullptr == window )
66 {
67     std::cout << "Failed to create GLFW window" << std::endl;
68     glfwTerminate();
69
70     return EXIT_FAILURE;
71 }
72
73 glfwMakeContextCurrent( window );
74
75 glfwGetFramebufferSize( window, &SCREEN_WIDTH, &SCREEN_HEIGHT );
76

```

Figura 14. Código 4

En la figura número 15 se muestran las instancias de los modelos que se ocupan

```

Model Arbol((char*)"Models/arbol/ArbNav.obj");
Model Bota((char*)"Models/bota/bota.obj");

Model CamIzq((char*)"Models/campana/camIzq.obj");
Model CamDer((char*)"Models/campana/camDer.obj");
Model Liston((char*)"Models/campana/liston.obj");

Model Chimenea((char*)"Models/chimenea/chimenea.obj");
Model RegaloGris((char*)"Models/regalo/regaloGris.obj");
Model RegaloVerde((char*)"Models/regalo/regaloVerde.obj");
Model RegaloAzul((char*)"Models/regalo/regaloAzul.obj");
Model RegaloAmarillo((char*)"Models/regalo/regaloAmarillo.obj");
Model RegaloTurqueza((char*)"Models/regalo/regaloTurqueza.obj");

Model Santa((char*)"Models/santa claus/Santa.obj");
Model piernasSanta((char*)"Models/santa claus/piernasSanta.obj");

Model Sofa((char*)"Models/sofa/sofa.obj");
Model Fachada((char*)"Models/fachada/fachada.obj");
Model muneco((char*)"Models/fachada/Hn.obj");
Model Estrella((char*)"Models/adornos/estrella/Gold_Star.obj");

```

Figura 15. Código 5

En esta pequeña sección tenemos el manejo del color y la proyección de nuestro proyecto


```

131 {
132     // Set frame time
133     GLfloat currentFrame = glfwGetTime();
134     deltaTime = currentFrame - lastFrame;
135     lastFrame = currentFrame;
136
137     // Check and call events
138     glfwPollEvents();
139     DoMovement();
140
141     // Clear the colorbuffer
142     glClearColor(0.5f, 0.5f, 0.5f, 1.0f);
143     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
144
145     shader.Use();
146
147     glm::mat4 view = camera.GetViewMatrix();
148     glUniformMatrix4fv(glGetUniformLocation(shader.Program, "projection"), 1, GL_FALSE, glm::value_ptr(projection));
149     glUniformMatrix4fv(glGetUniformLocation(shader.Program, "view"), 1, GL_FALSE, glm::value_ptr(view));
150
151     // Draw the loaded model

```

Figura 16. Código 6

Seguimos con el dibujo de nuestros modelos, así como las transformaciones que les aplicamos

```

// Draw the loaded model

//Fachada
glm::mat4 model(1);
model = glm::translate(model, glm::vec3(0.0f, 0.0f, -10.0f));
model = glm::scale(model, glm::vec3(2.5f, 2.5f, 2.5f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
Fachada.Draw(shader);

//bota

model = glm::mat4(1);
model = glm::rotate(model, glm::radians(rot), glm::vec3(-45.0f, 270.0f, 180.0));
model = glm::translate(model, glm::vec3(15.0f, 7.0f, -12.0f));
model = glm::translate(model, glm::vec3(0.0f, trs, 0.0f));
model = glm::scale(model, glm::vec3(0.5f, 0.5f, 0.5f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
Bota.Draw(shader);

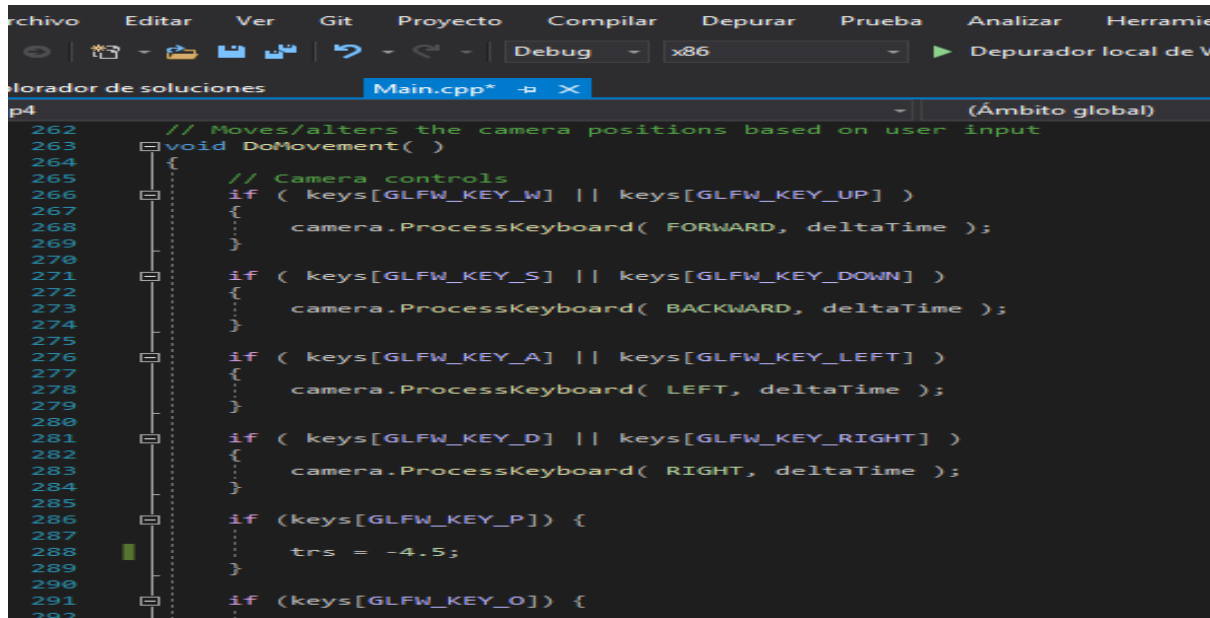
// arbol

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(11.5f, 1.3f, 11.0f));
model = glm::scale(model, glm::vec3(1.2f, 1.25f, 1.2f));
glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
Arbol.Draw(shader);

```

Figura 17. Código 7

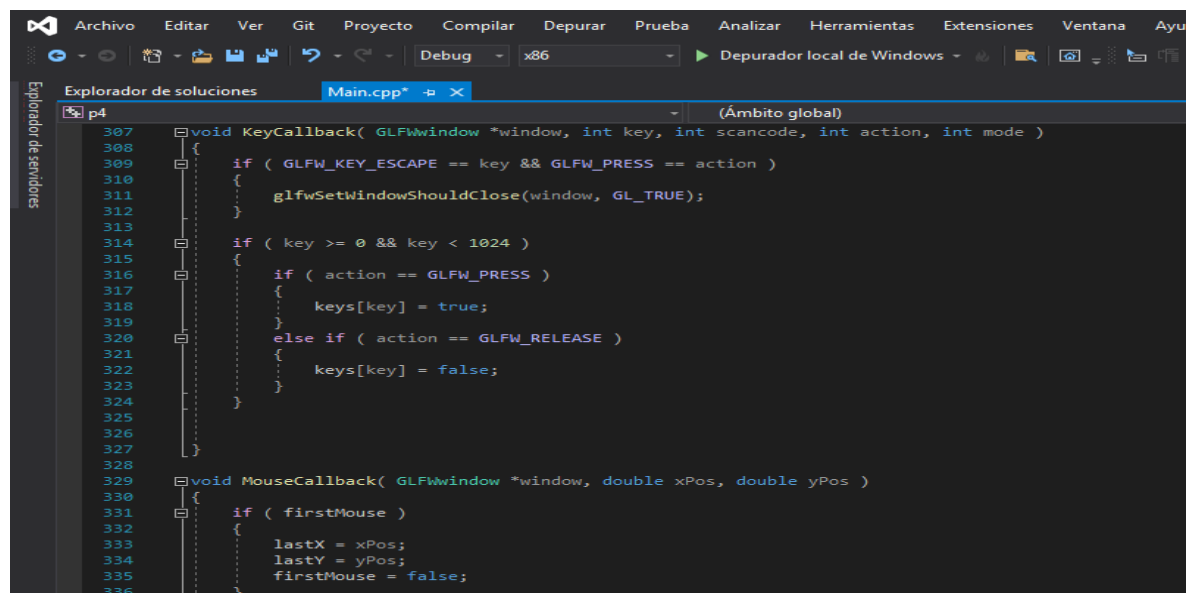
En la figura numero 18 tenemos la función DoMovement donde se encuentra el manejo de nuestra cámara y las animaciones ya que aquí se encuentran las posiciones y teclas con las que se activan dichos movimientos.



```
262 // Moves/alters the camera positions based on user input
263 void DoMovement( )
264 {
265     // Camera controls
266     if ( keys[GLFW_KEY_W] || keys[GLFW_KEY_UP] )
267     {
268         camera.ProcessKeyboard( FORWARD, deltaTime );
269     }
270
271     if ( keys[GLFW_KEY_S] || keys[GLFW_KEY_DOWN] )
272     {
273         camera.ProcessKeyboard( BACKWARD, deltaTime );
274     }
275
276     if ( keys[GLFW_KEY_A] || keys[GLFW_KEY_LEFT] )
277     {
278         camera.ProcessKeyboard( LEFT, deltaTime );
279     }
280
281     if ( keys[GLFW_KEY_D] || keys[GLFW_KEY_RIGHT] )
282     {
283         camera.ProcessKeyboard( RIGHT, deltaTime );
284     }
285
286     if (keys[GLFW_KEY_P]) {
287         trs = -4.5;
288     }
289
290     if (keys[GLFW_KEY_O]) {
291
292
```

Figura 18. Código 8

Para finalizar tenemos esta última sección de código



```
307 void KeyCallback( GLFWwindow *window, int key, int scancode, int action, int mode )
308 {
309     if ( GLFW_KEY_ESCAPE == key && GLFW_PRESS == action )
310     {
311         glfwSetWindowShouldClose(window, GL_TRUE);
312     }
313
314     if ( key >= 0 && key < 1024 )
315     {
316         if ( action == GLFW_PRESS )
317         {
318             keys[key] = true;
319         }
320         else if ( action == GLFW_RELEASE )
321         {
322             keys[key] = false;
323         }
324     }
325
326 }
327
328 void MouseCallback( GLFWwindow *window, double xPos, double yPos )
329 {
330     if ( firstMouse )
331     {
332         lastX = xPos;
333         lastY = yPos;
334         firstMouse = false;
335     }
336
```

Figura 19. Código 9

Resultados



Figura 20. Resultados



Figura 21. Resultados

Conclusiones

Este proyecto nos ayudó a aplicar y reforzar todos los conocimientos aprendidos durante el curso ya que en cada practica se realizo por separado una sección del trabajo aquí realizado, algunos de estos son modelar, texturizar, animar etcétera. Además, nos ayudó a conocer algunos otros aspectos que se deben considerar en la realización de un proyecto como lo son el tiempo y el equipo de trabajo, que en mi caso es muy importante ya que fue una limitante muy importante porque no permitió desarrollar satisfactoriamente el proyecto.