

Universidad Nacional Autónoma de México
Facultad de Ingeniería

Computación Gráfica e Interacción Humano-Computadora

PROYECTO FINAL

Grupo 01

Equipo 12

Alumno: Rosales Mendoza Armando José

México, Ciudad de México. 08 de diciembre del 2021

Objetivo

Se deberá realizar un recorrido virtual del interior de un espacio habitacional donde se incluirán elementos generados por cada alumno en específico.

Además de una ambientación tridimensional para simular lo mejor posible el ambiente virtual.

Se deberá realizar el ambiente virtual lo más completo posible contemplando principalmente el interior, aunque deberá contemplar el exterior por lo menos de una fachada principal, colocando diversos elementos.

Especificaciones

El alumno o equipo propondrá el interior a representar.

La propuesta deberá ser entregada con anticipación para obtener su visto bueno antes de ser construida mediante un documento de propuesta de proyecto.

El documento deberá incluir una descripción general del proyecto a entregar, imágenes de referencia del proyecto a proponer, esquemas de la propuesta de la distribución de cada espacio a representar, propuesta de animaciones a realizar y planeación tentativa de la realización del proyecto donde se debe incluir el precio aproximado del mismo.

El proyecto deberá ser entregado en formato digital, donde se colocará el proyecto de visual studio, al menos 3 videos diferentes del proyecto, mostrando su correcto funcionamiento, un manual de usuario, cronograma de actividades realizadas para la conclusión de este y el precio aproximado del proyecto.

El manual de usuario deberá estar en formato .pdf y debe contener capturas de pantalla que ejemplifiquen cada elemento que describen en el manual y al final capturas de pantalla del proyecto.

El cronograma de actividades deberá estar lo más detallado posible en función del trabajo individual y del equipo. Se otorgarán puntos adicionales si se entrega evidencias de haber utilizado herramientas colaborativas para el desarrollo del software pudiendo ser Jira, Trello, Git, GitHub, entre otros.

Además, el proyecto deberá estar en un repositorio exclusivo donde se evaluará cada avance que se vaya realizando en él.

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. Sala de la casa



Figura 3. Comedor



Figura 4. Habitaciones



Figura 5. Baño

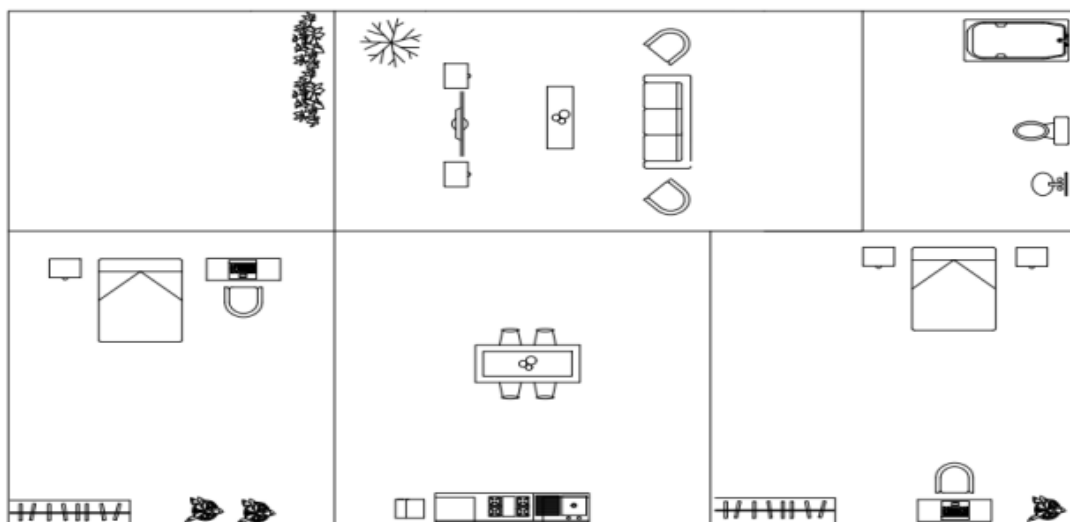


Figura 6. Distribución de la casa

Contenido

En esta parte se mostrará parte del desarrollo del proyecto y como primer punto se muestra el modelado y texturizado de algunos objetos

Árbol de Navidad



Figura 7. Árbol en maya

Bota navideña

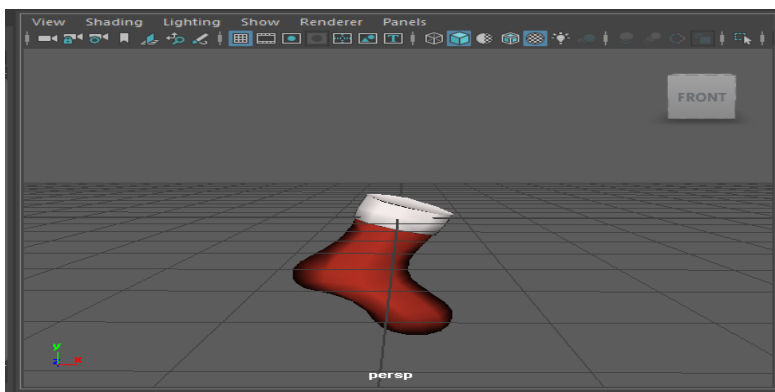


Figura 8. Bota modelada

Campanas

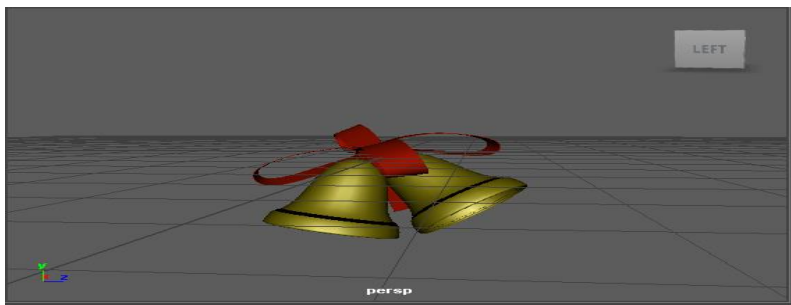


Figura 9. Campanas navideñas en maya

Chimenea

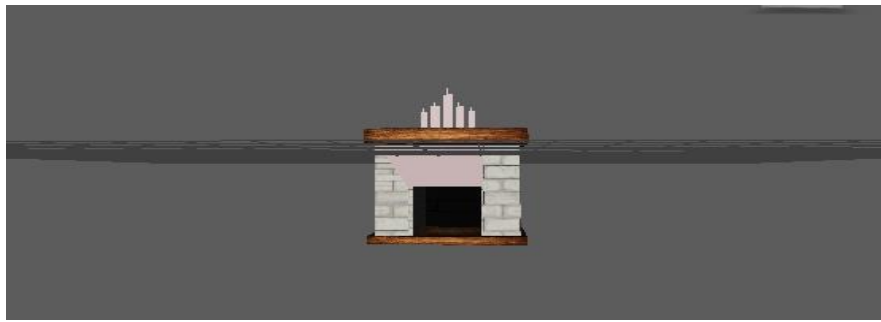


Figura 10. Chimenea

Regalos

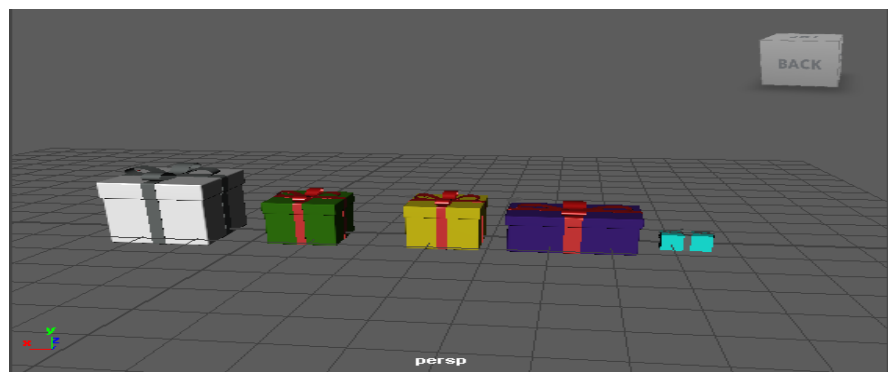


Figura 11. Regalos en maya

Santa



Figura 12. Santa Claus en maya

Sofá

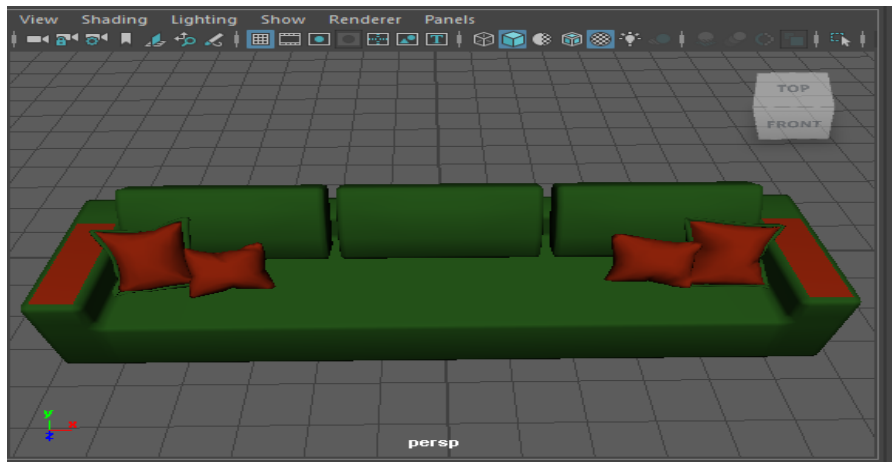


Figura 13. Sofá en maya

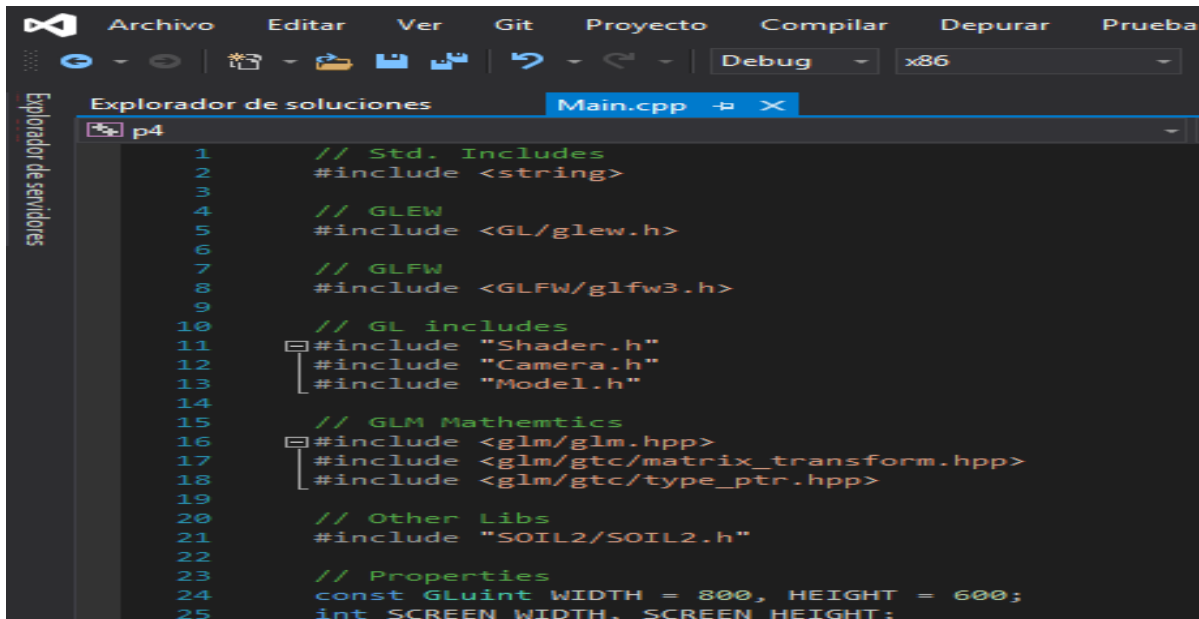
Fachada



Figura 14. Fachada en maya

Documentación

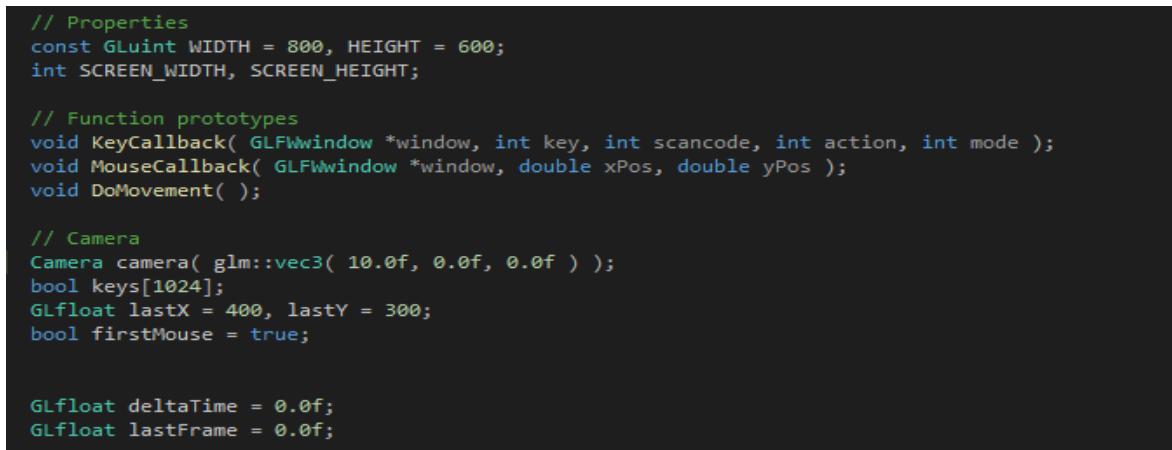
En estas primeras líneas de código se definen los encabezados que se ocuparan más adelante



```
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 15. 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 16. 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 17. Código 3

En esta siguiente sección seguimos con código que crea la ventana

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

Figura 18. Código 4

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

```
////////////////////////////////////  
// Load models  
  
Model Fachada((char*)"Models/fachada/fachada01.obj");  
Model Fachada2((char*)"Models/fachada/fachada02.obj");  
Model Fachada3((char*)"Models/fachada/fachada03.obj");  
Model Bota((char*)"Models/bota/bota.obj");  
Model Arbol((char*)"Models/fachada/arbolnav.obj");  
  
////////////////////////////////////
```

Figura 19. 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 20. Código 6

Seguimos con el dibujo de nuestros modelos, así como las transformaciones que les aplicamos, en nuestro caso dividimos la casa en tres partes la primera y la segunda muestran parte de la estructura de la casa y la tercera muestra los muebles, esto se hace debido a que el equipo no aguanta la ejecución completa de este

```

144 //Estructura//
145
146 glm::mat4 model(1);
147 model = glm::translate(model, glm::vec3(0.0f, 0.0f, -10.0f));
148 model = glm::scale(model, glm::vec3(2.5f, 2.5f, 2.0f));
149 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
150 Fachada.Draw(shader);
151
152 //segunda seccion//
153
154 model = glm::mat4(1);
155 model = glm::translate(model, glm::vec3(0.0f, 0.0f, -10.0f));
156 model = glm::scale(model, glm::vec3(2.5f, 2.5f, 2.0f));
157 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
158 Fachada2.Draw(shader);
159
160 //tercera seccion //muebles//
161
162 model = glm::mat4(1);
163 model = glm::translate(model, glm::vec3(0.0f, 0.0f, -10.0f));
164 model = glm::scale(model, glm::vec3(2.5f, 2.5f, 2.0f));
165 glUniformMatrix4fv(glGetUniformLocation(shader.Program, "model"), 1, GL_FALSE, glm::value_ptr(model));
166 Fachada3.Draw(shader);

```

Figura 21. 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.

```

archivo  Editar  Ver  Git  Proyecto  Compilar  Depurar  Prueba  Analizar  Herramientas
Debug  x86  Depurador local de Visual Studio
Solucionador de problemas  Main.cpp*  X
p4  (Ámbito global)
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 22. Código 8

Para finalizar tenemos esta última sección de código

Figura 23. Código 9

Resultados



Figura 24. Resultados



Figura 25. Resultados

Diagrama de gant



Figura 26. Diagrama de Gant

La figura 26 muestra el diagrama de Gant que se propuso, entonces de acuerdo con lo desarrollado podemos decir que no cumplimos completamente con el proyecto ya no se terminaron las animaciones. Sin embargo cumplimos con todas las demás actividades en tiempo y forma aunque se atraso un poco en la modelación, se pudo recuperar el tiempo en las ultimas semanas.

Extras

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 ya cada vez era más difícil hacer ligeros cambios en código ya que a veces la depuración podía tardar 40 minutos y a veces visual Studio dejaba de funcionar, además que no permiten mostrar mejores graficas.

Aprendizajes

El desarrollo de este proyecto también nos dejó aprendizajes extraordinarios a los temas vistos en clase tales pueden ser el uso de programas de modelado como maya y de programación como visual Studio y también nos ayudo a repasar o mejorar algunos otros conocimientos como es el manejo de GitHub, ya que lo utilice anteriormente, pero esta vez podía explorare algunas otras características.

Costos

Primera cotizacion

Para poder realizar este apartado, realice una búsqueda en varios sitios web y a partir de la información obtenida, la cotización del proyecto debe considerar los siguientes aspectos:

- ❖ Modelado
- ❖ animaciones
- ❖ Tiempo
- ❖ Software
- ❖ Trabajo

Entonces a partir de esto, considerando que ya conocemos lo que tenemos y necesitamos

los precios son los siguientes:

Costo si yo comprar el proyecto

\$ 11,000.00 MN

Costo si yo vendiera el proyecto

\$ 13,000.00 MN

Segunda Cotización

Ahora bien, después de la realización del proyecto y de una nueva investigación los nuevos costos que ahora también consideran otros factores como dificultades y la animación en México

los precios son los siguientes:

Costo si yo comprar el proyecto

\$ 13,000.00 MN

Costo si yo vendiera el proyecto

\$ 15,000.00 MN

Conclusiones

Este proyecto nos ayudó a aplicar y reforzar todos los conocimientos aprendidos durante el curso ya que en cada práctica se realizó 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.

Referencias

Mercado, P. (2020, 31 agosto). Como Cobrar Animación. IndustriaAnimacion.com. Recuperado 31 de octubre de 2021, de <https://www.industriaanimacion.com/2018/09/como-cobrar-animacion/>

Irigoyen, D. J. P. (s. f.). Aprende a cotizar y cobrar tus proyectos de motion graphics y animación. Designare Media - Arte y Animación. Recuperado 31 de octubre de 2021, de <https://www.designare.com/2020/04/aprende-cotizar-y-cobrar-tus-proyectos.html>

Kong, A. (2019, 19 junio). Animación en México. La vida en esta industria. Alex Kong. Recuperado 8 de diciembre de 2021, de <https://www.alexkong.mx/animacion-en-mexico-la-vida-en-esta-industria/>