

---

Laboratorio de Computación Gráfica e Interacción Humano-Computadora

Semestre 2024-1

PROYECTO FINAL  
**MANUAL TÉCNICO**

Equipo 6  
Integrantes del equipo:

# cuenta	Nombre
417006480	Dionicio Serna Marisela
315138151	Hernández Sánchez Brenda

Fecha de entrega de proyecto: 22 de noviembre de 2023

---

# Índice

<b>Objetivos</b>	<b>3</b>
<b>Información técnica</b>	<b>3</b>
• Hardware	3
• Software	
<b>Referencias para la obtención de modelos</b>	<b>4</b>
<b>Descarga e Instalación del proyecto</b>	<b>4</b>
<b>Planeación del proyecto</b>	<b>5</b>
• Diagrama de Gantt	5
• Cronograma de actividades	5
• Costo del proyecto	7
<b>Documentación del código</b>	<b>8</b>
• Cámara	8
• Carga de modelos	9
• Dibujo de modelos	10
• Carga de Shaders	10
• Iluminación	11
<b>Entorno Virtual</b>	<b>11</b>
• Vista inicial del proyecto	11
• Sala 1	14
• Sala2	15
• Sala 3	16
• Zona infantil	17
• Lago	17
<b>Animaciones</b>	<b>18</b>
<b>Musica</b>	<b>19</b>
<b>Conclusiones</b>	<b>20</b>
• Comentario individual (En ingles)	20

## Objetivo

Recrear un escenario gráfico de un Museo haciendo uso de las herramientas aprendidas a lo largo del curso de Computación Gráfica. Con lo anterior se pretende poner en práctica todas las herramientas vistas en clase desde el modelado hasta animaciones.

## Información técnica

- **Hardware**

El proyecto final fue realizado bajo una serie de especificaciones tanto de software como de hardware.

- Características mínimas:

- Procesador: Intel Core i3 de Quinta Generación o AMD Ryzen 3
- Memoria RAM: 8 GB
- Almacenamiento: 5 GB
- Sistema Operativo: Windows 7 o superior.

- Características recomendables:

- Procesador: Intel Core i5 a partir de Octava Generación o AMD Ryzen 5
- Memoria RAM: 16 GB
- Almacenamiento: 5 GB
- Sistema Operativo: Windows 10 o superior.

- **Software**

- OpenGL, C++ en Visual Studio 2022
- Blender 3.3
- 3ds Max 2023
- GIMP

## Referencias para la obtención de modelos

Algunos modelos que contiene el programa fueron creados desde 0, sin embargo fue necesario apoyarse de diversas páginas de internet para obtener otros, las cuales nos brindan en diferentes formatos.

Algunas de las páginas utilizadas para la obtención de modelos con extensión OBJ fueron:

- <https://www.turbosquid.com/>
- <https://free3d.com/es/modelos-3d/obj>

- <https://www.artec3d.com/es/3d-models/art-and-design>
- <https://sketchfab.com/feed>

## Descarga e Instalación del proyecto

Para instalar el proyecto debemos dirigirnos a la carpeta de Drive del siguiente link.

<https://drive.google.com/drive/folders/1ew4jdVaryHaE5sL9uPcw8Pr2oYMBfBU3?usp=sharing>

o del siguiente repositorio de GitHub

[https://github.com/MariselaDS/ProyectoFinal\\_CGIHC.git](https://github.com/MariselaDS/ProyectoFinal_CGIHC.git)

Ahí deberemos descargar el archivo “PF\_CGIHC.zip”, descomprimimos el archivo, entramos a la carpeta si lo que queremos es abrir el proyecto en Visual Studio y seleccionamos *PF\_CGIHC.sln*, si lo que queremos es usar el programa avanzamos a la carpeta “Debug” y ejecutaremos el archivo .exe.

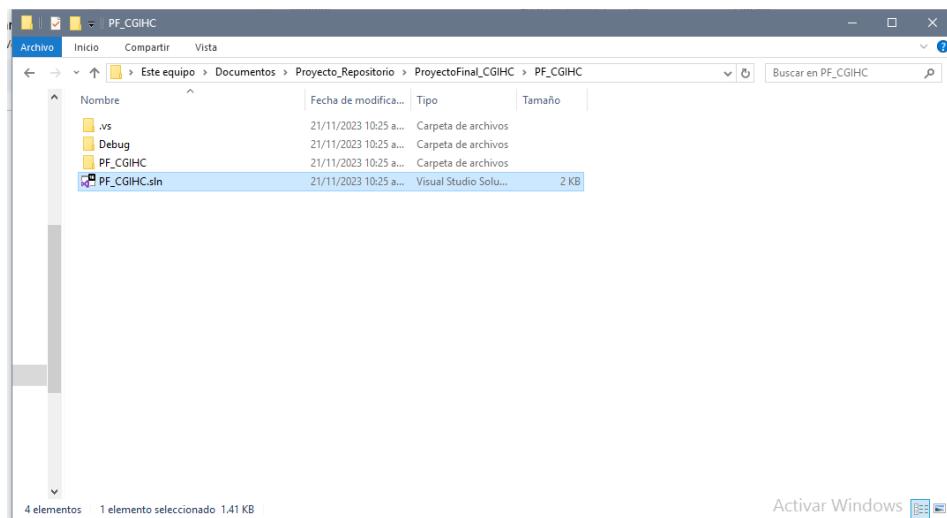
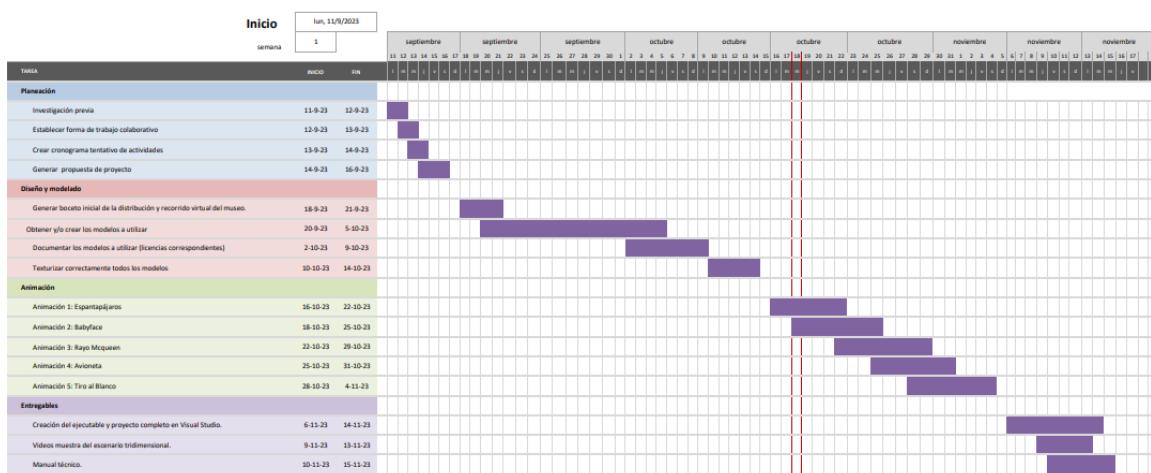


Figura 1: Carpeta con todos los archivos del proyecto para Visual Studio

# Planeación del proyecto

- **Diagrama de Gantt**

Planeación tentativa del proyecto



*Figura 2: Diagrama ilustrativo de actividades*

- **Cronograma de actividades**

Inicio: 6 de septiembre de 2023	Finalización: 16 de noviembre de 2023
---------------------------------	---------------------------------------

Fase	Objetivo	Actividades	Duración
<b>Planificación</b>	Definir alcance del proyecto y generar propuesta basada en los requerimientos solicitados.	Investigación previa necesaria para comprender los requerimientos (respecto a modelado, animación, etc.)	1 semana Del 22/09/2023 al 27/09/2023
Responsable: Néstor López González		Establecer forma de trabajo colaborativo.  Crear cronograma tentativo de actividades.  Generar documento de propuesta de proyecto.	

<b>Diseño y modelado</b>  Responsable: Brenda Hernández Sánchez	Obtener todos los elementos 3D (correctamente texturizados) que se utilizarán en cada uno de los espacios propuestos.	Generar boceto inicial de la distribución y recorrido virtual del museo.  Obtener y/o crear los modelos a utilizar.  Documentar los modelos a utilizar (licencias correspondientes).  Texturizar correctamente todos los modelos.	4 semanas  Del 29/09/2023 al 25/10/2023
<b>Animación</b>  Responsable: Marisela Dionicio Serna	Implementar 5 animaciones por medio de diferentes técnicas y/o herramientas disponibles.	Animación 1: Espantapájaros  Animación 2: Babyface  Animación 3: Rayo Mcqueen  Animación 4: Avioneta  Animación 5: Tiro al blanco	3 semanas  Del 27/10/2023 al 15/11/2023
<b>Entregables</b>  Responsable: Néstor López González	Documentación del entorno virtual tanto en modo usuario como a nivel técnico.	Creación del ejecutable y proyecto completo en Visual Studio.  Videos muestra del escenario tridimensional.  Manual técnico.	2 semanas  Del 17/11/2023 al 26/11/2023

- **Costo proyecto**

La estimación de costos asociados a nuestro proyecto de museo “Mundos animados” se consideran varios elementos. A continuación proporcionamos una lista con los elementos a considerar por tres meses:

- Costos de Software: El software que se empleará en el proceso de desarrollo es:

Software			
	Unidades	Precio Unitario	Precio Total (3 meses)
3Ds MAX	3	\$2594 / mes	\$23,346
Modelos	0	\$0	\$0
<b>Total</b>			<b>\$23,346</b>

Inversión en Modelos y Licencias: Los modelos que serán utilizados son de licencia libre por lo tanto no tienen costo.

- Gastos de Desarrollo: Dado que contamos con un equipo conformado por dos desarrolladores, estimamos el costo por hora por desarrollador en \$175 MXN. Suponiendo que cada desarrollador trabaja aproximadamente 6 horas al día durante los 2 meses y 2 semanas del proyecto, el gasto total en desarrollo se estima en:

$$2 \text{ desarrolladores} \times \$175 \text{ MXN/hora} \times 6 \text{ horas/día} \times 5 \text{ días/semana} \times 12 \text{ semanas} = \$126,000 \text{ MXN.}$$

Recursos humanos			
	Unidades	Precio Unitario	Precio Total (3 meses)
Salarios (Programadores, diseñadores, animadores)	2	\$21000 / mes	\$126,000

- Gastos en Servicios Complementarios: Se prevé un costo suplementario para cubrir servicios de electricidad, acceso a Internet, y transporte público para las reuniones presenciales. Este gasto se calcula en \$6,000 MXN, teniendo en cuenta los aumentos en los costos de servicios públicos y las posibles alternativas necesarias por el cierre parcial de varios de estos servicios.

En consecuencia, el gasto total previsto para el proyecto de computación gráfica en 2023 se desglosa de la siguiente manera:

- Costos software: \$ 23,346 MXN
- Modelos: \$ 0 MXN
- Desarrollo: \$126,000 MXN
- Complementarios: \$6,000 MXN

**Total:** \$ 155,346 MXN

## Documentación del código

A continuación se hablará de algunos elementos importantes implementados dentro del código, con el objetivo de mostrar al lector las metodologías utilizadas.

- Cámara

```

43
44 // Cámara
45 Camera camera(glm::vec3(0.0f, 50.0f, 220.0f));
46 float MovementSpeed = 0.1f;
47 float lastX = SCR_WIDTH / 2.0f;
48 float lastY = SCR_HEIGHT / 2.0f;
49 bool firstMouse = true;
50

```

*Figura 3: Código de implementación de cámara*

El proyecto cuenta con una cámara que muestra la vista principal del proyecto. La posición de ésta se controla con las siguientes teclas y la dirección con el movimiento del mouse:

- Tecla /arriba: mover cámara hacia delante
- Tecla /abajo: mueve la cámara hacia atrás
- Tecla /izquierda: mueve la cámara hacia la izquierda
- Tecla /derecha: mueve la cámara hacia la derecha

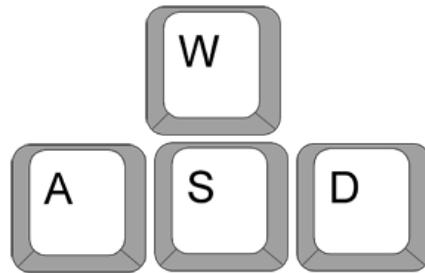


Figura 4 : Teclas para manejo de cámara

- Carga de modelos

De la siguiente forma se cargaron los modelos realizados en el software de modelado. Se encuentran separados por carpetas dentro de resources\objects. El código implementado es el siguiente:

```

296 Model piso("resources/objects/piso/PisoArboles.obj");
297 Model baby("resources/objects/babyface/babyface.obj");
298 Model sullivan("resources/objects/sullivan/sullivan.obj");
299 Model lago("resources/objects/Lago/lago.obj");
300 Model fuente("resources/objects/Fuente/Fuen.obj");
301 Model museo("resources/objects/museo/museo.obj");
302 Model banca("resources/objects/Silla/Silla.obj");
303 Model caballo("resources/objects/tiroAlBlanco/tiroalblanco.obj");
304 Model stitch("resources/objects/Stitch/Stitch.obj");
305 Model lilo("resources/objects/Lilo/lilo.obj");
306 Model mcqueen("resources/objects/McQueen/mcqueenCarroceria.obj");
307 Model llanta("resources/objects/McQueen/Wheel.obj");
308 Model aladin("resources/objects/Aladin/DP_Aladdin_T.obj");
309 Model nala("resources/objects/Nala/DP_Nala.obj");
310 Model espantapajaros("resources/objects/espantapajaros/espantapajaros.obj");
311 Model yubaba("resources/objects/yubaba/yubaba.obj");
312 Model avion("resources/objects/porcoAvion/porcoAvion.obj");
313 Model chihiro("resources/objects/Chihiro/SG_Chihiro.obj");
314 Model ohmu("resources/objects/Ohmu/SG_Ohmu.obj");
315 Model princesa("resources/objects/Princesa/SG_PrincesaMononoke.obj");
316 Model bugs("resources/objects/BugsBunny/BugsBunny.obj");
317 Model pato("resources/objects/PatoLucas/DaffyDuck.obj");
318 Model coyote("resources/objects/Coyote/WB_Coyote.obj");
319 Model correCaminos("resources/objects/CorreCaminos/WB_CorreCaminos.obj");
320 Model taz("resources/objects/Taz/WB_Taz.obj");
321 Model marvin("resources/objects/Marvin/WB_Marvin.obj");
322 Model scooby("resources/objects/Scooby/WB_Scooby.obj");
323 Model vilma("resources/objects/Vilma/WB_Vilma.obj");
324 Model carroPicapiedras("resources/objects/CarroPicapiedras/WB_CarroPicapiedras.obj");
325 Model pedro("resources/objects/Pedro/WB_Pedro.obj");

```

Figura 5: Código para cargar modelos.

- Dibujo de modelos

Después de cargar los modelos, indicamos por medio de código el dibujado de los mismos, en esta sección se aplicaron las herramientas de transformación: traslación, rotación y escala. Este procedimiento fue aplicado en múltiples ocasiones a lo largo del proyecto

```

723     // Escenario
724     // -----
725     staticShader.use();
726     staticShader.setMat4("projection", projection);
727     staticShader.setMat4("view", view);
728
729     model = glm::mat4(1.0f);
730     model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
731     model = glm::scale(model, glm::vec3(1.0f, 1.3f, 1.5f));
732     staticShader.setMat4("model", model);
733     piso.Draw(staticShader);
734
735     model = glm::translate(glm::mat4(1.0f), glm::vec3(10.0f, 0.2f, 5.0f));
736     model = glm::scale(model, glm::vec3(0.13f));
737     staticShader.setMat4("model", model);
738     museo.Draw(staticShader);
739
740     model = glm::translate(glm::mat4(1.0f), glm::vec3(35.0f, 0.0f, -48.0f));
741     model = glm::scale(model, glm::vec3(1.0f));
742     staticShader.setMat4("model", model);
743     lago.Draw(staticShader);
744
745
746     model = glm::translate(glm::mat4(1.0f), glm::vec3(-65.0f, 0.9f, -110.0f));
747     model = glm::scale(model, glm::vec3(3.0f));
748     staticShader.setMat4("model", model);
749     banca.Draw(staticShader);
750
751     model = glm::translate(glm::mat4(1.0f), glm::vec3(-65.0f, 0.9f, -145.0f));
752     model = glm::scale(model, glm::vec3(3.0f));
753     staticShader.setMat4("model", model);

```

*Figura 6: Código para el dibujo de modelos.*

- Carga de Shaders

Al igual que los modelos, los shaders se cargan de forma similar, con la diferencia de que ahora se especifica la ruta con el formato correspondiente.

```

271     // build and compile shaders
272     // -----
273     Shader staticShader("Shaders/shader_Lights.vs", "Shaders/shader_Lights_mod.fs");
274     Shader skyboxShader("Shaders/skybox.vs", "Shaders/skybox.fs");
275     Shader animShader("Shaders/anim.vs", "Shaders/anim.fs");

```

*Figura 7: Código para la carga de shaders*

- Iluminación

En esta sección se optó por la iluminación ambiental y darle brillo a cada uno de los objetos del escenario.

```
674 // don't forget to enable shader before setting uniforms  
675 staticShader.use();  
676 //Setup Advanced Lights  
677 staticShader.setVec3("viewPos", camera.Position);  
678 staticShader.setVec3("dirLight.direction", lightDirection);  
679 staticShader.setVec3("dirLight.ambient", glm::vec3(0.9f, 0.9f, 0.9f));  
680 staticShader.setVec3("dirLight.diffuse", glm::vec3(0.4f, 0.4f, 0.4f));  
681 staticShader.setVec3("dirLight.specular", glm::vec3(0.3f, 0.3f, 0.3f));  
682
```

Figura 8: Código para la iluminación

## Entorno virtual final

- Vista inicial del museo



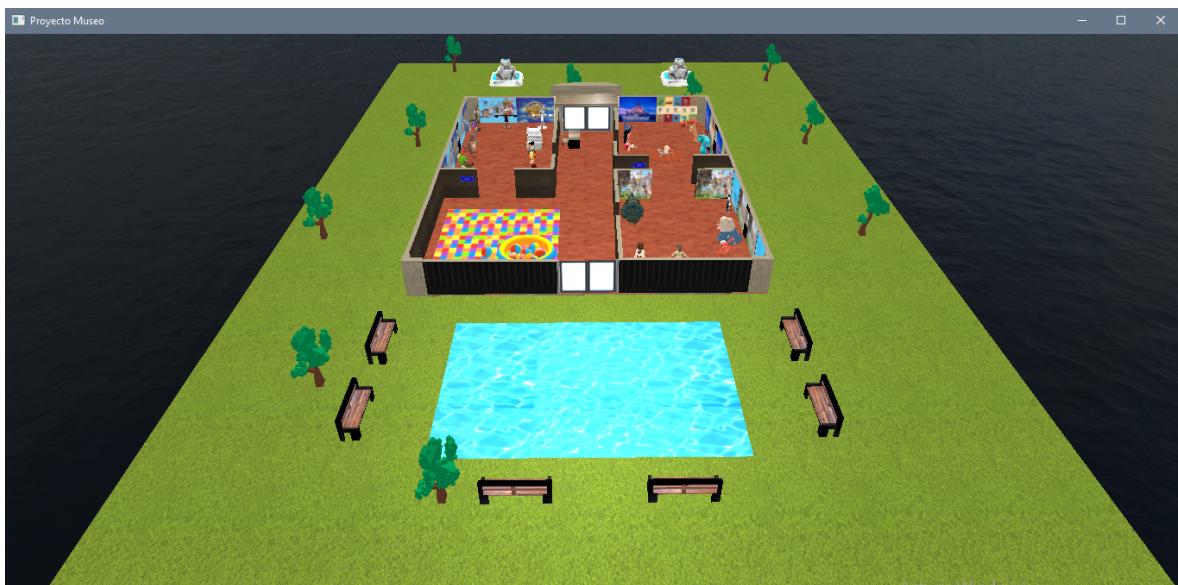
- Vista superior del museo sin techo



- Vista derecha del museo



- Vista trasera del museo sin techo



- Vista trasera del museo con techo



- Vista izquierda del museo sin techo



A continuación se muestra más a detalle cada una de las secciones que conforman nuestro escenario 3D

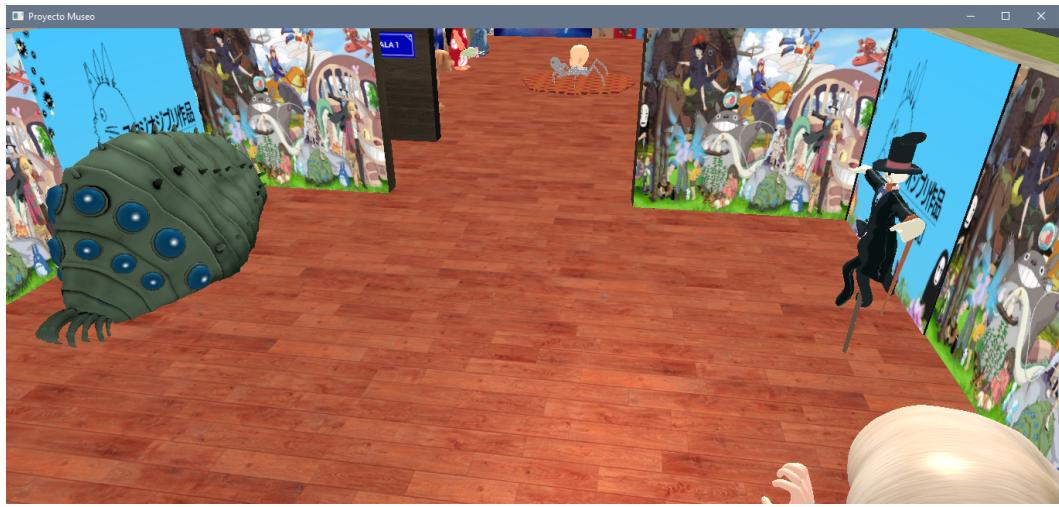
- Sala 1 Disney Pixar





- Sala 2 Studio Ghibli





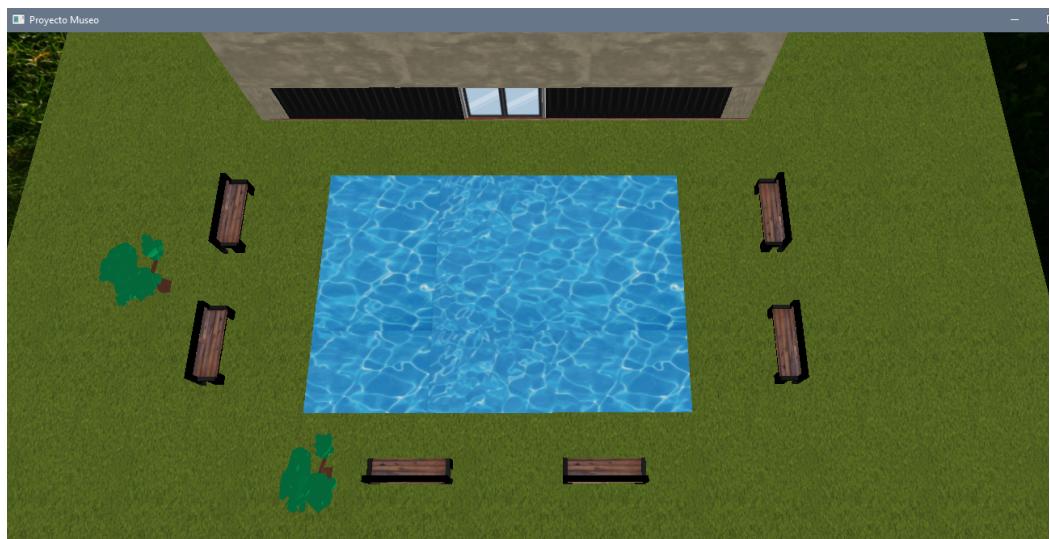
- Sala 3 Warner Bros



- Zona Infantil



- Lago



## Animaciones

Para las animaciones se utilizó la técnica de cuadros clave, en este caso del espantapájaros se guardaron 63 fotogramas. El código implementado es el siguiente.

```
328     //Inicialización de KeyFrames
329     KeyFrame[0].posX = 0.0f;
330     KeyFrame[0].posY = 0.0f;
331     KeyFrame[0].posZ = 0.0f;
332     KeyFrame[0].giroMonito = -45.0f;
333
334     KeyFrame[1].posX = 0.0f;
335     KeyFrame[1].posY = 1.0f;
336     KeyFrame[1].posZ = 1.0f;
337     KeyFrame[1].giroMonito = -45.0f;
338
339     KeyFrame[2].posX = 1.0f;
340     KeyFrame[2].posY = 2.0f;
341     KeyFrame[2].posZ = 2.0f;
342     KeyFrame[2].giroMonito = -45.0f;
343
344     KeyFrame[3].posX = 2.0f;
345     KeyFrame[3].posY = 3.0f;
346     KeyFrame[3].posZ = 3.0f;
347     KeyFrame[3].giroMonito = -45.0f;
348
349     KeyFrame[4].posX = 3.0f;
350     KeyFrame[4].posY = 4.0f;
351     KeyFrame[4].posZ = 4.0f;
352     KeyFrame[4].giroMonito = -45.0f;
353
354     KeyFrame[5].posX = 4.0f;
355     KeyFrame[5].posY = 3.0f;
356     KeyFrame[5].posZ = 5.0f;
357     KeyFrame[5].giroMonito = -45.0f;
```

Figura 9: Código de animación por cuadros clave espantapájaros

Teclas para activar animaciones

Tecla	Animación
P	Animación espantapájaros
P	Animación Tiro al Blanco
P	Animación Babyface
Barra espaciadora y R	Animacion Rayo Mcqueen



Figura 10: Animación personaje (Mixamo)

## Música

El siguiente código fue la implementación de la música, escucharemos al oprimir la tecla L “Yo soy tu amigo fiel” de Toy Story

```
1026
1027    [ ] //Song
1028    if (key == GLFW_KEY_L && action == GLFW_PRESS)
1029    {
1030        PlaySound(TEXT("SongToyStory.wav"), NULL, SND_ASYNC);
1031    }
1032
1033
```

Figura 11: Código para activar música

## Conclusiones

- Marisela:

Below I will describe the activities that I carried out during the development of this project.

Firstly, when the project requirements were made known to us, I conducted research before making the initial project proposal. This research consisted mainly of searching for information about 3D modeling (the programs that can be used to manage models, ways to texture a model, etc.), searching for the animation techniques that are covered during the course of the subject and the possible animations that can be implemented with such methods. Subsequently, I carried out a search for 3D models on various internet pages based on the different theme ideas proposed by all the members of the team. With the results of that search, taking into account the quantity and quality of the models found, the choice was made on the theme of the museum.

The second activity I did and the one that took the most time was the entire process of preparing the 3D models to be used on the stage. Before proceeding with the application of the textures to each model, I did some steps such as converting between file formats to be able to import them into 3DS Max and correcting the images to be used as textures since many of these did not meet the characteristics necessary so that the model could be successfully loaded into Visual Studio. There were some complications with several models downloaded from the internet, since they did not have the corresponding texturing file or were made up of too many vertices.

The third activity consisted of the implementation of some of the proposed animations. In order to do this I had to cut the 3D models that we wanted to animate and place their centers and pivots correctly to achieve the desired animation. Finally, in Visual Studio I built the figure with all its parts using hierarchical modeling and the animation was implemented.

The development of this final project, in each of its stages, gave me the opportunity to put into practice the concepts studied in the theory class.

- Brenda:

The development of this project allowed me to put into practice the concepts involved in creating a three-dimensional scenario, from creating the models to programming them in OpenGL. I can say that in this process I had a more determined vision of what I planned to do. I know the essentials in Blender to create models from scratch, but my goal was to learn how to use 3ds Max. The structure of the museum was a challenge for me, as applying textures was the most complicated part I faced. When exporting the model, I didn't consider them all, but after several attempts, I realized where the error was coming from and was able to solve it. The scales were not square, and thanks to GIMP, I resolved it. I became familiar with the software so that handling it for what I intended to present was easier and more agile for me. The most tedious part was in the code because this is where we load the models and apply the transformation tools if they don't match in scale and rotation within the scene. At first, I thought that creating animations for the proposed models would be very difficult. At that moment, I had no minimal information to have an idea. It was in laboratory classes where I learned the importance of keyframes, drawing variables, and the use of arrays to move each part that make up a character, save them, and then play them back. Learning to use GitHub was another important thing I achieved with the completion of this project, cloning a repository to my local computer and from there, using the console to upload, update, or delete files. This virtual scenario leaves me very satisfied. Seeing days of work in a project that didn't present many complications to compile, load models, and animate is an achievement for me, after seeing the final result.

## **Archivos del proyecto**

- Repositorio de GitHub

[https://github.com/MariselaDS/ProyectoFinal\\_CGIHC.git](https://github.com/MariselaDS/ProyectoFinal_CGIHC.git)

- Videos

[https://drive.google.com/drive/folders/1R5rrLSOe-5M\\_ckSvRCpWkf2VhRw8A9Wq?usp=drive\\_link](https://drive.google.com/drive/folders/1R5rrLSOe-5M_ckSvRCpWkf2VhRw8A9Wq?usp=drive_link)