UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Computación gráfica e interacción
humano computadora

Final Project

TEACHER: Carlos Aldair Román Balbuena

STUDENT: 319049792

GROUP: 05

SEMESTER: 2026-1

DELIVERY: November 25, 2025

TECHNICAL MANUAL

INDEX

Content

1.- Goals

- The student must apply and demonstrate the knowledge acquired during the course by creating a 3D recreation in OpenGL.
- Recreate the facade of the protagonist's house from the game "Grand Theft Auto: San Andreas".
- Implement four animations, each of which must be contextualized within the environment.
- Create keyboard inputs so the user can interact with the generated 3D environment.
- Provide the total cost of the implemented project.

2.- State of the Art

Virtual tours are digital simulations of real or fictional locations that allow users to move around and interact with an environment without being physically present. Their application has expanded beyond mere entertainment, becoming an essential tool that transforms business and educational processes by improving accessibility, reducing operating costs, and increasing user immersion.

Solutions have evolved from simple 360° images to complex 3D environments created using photogrammetry or game engines (such as Unreal Engine or Unity), offering a sense of complete presence, especially through Virtual Reality (VR) headsets.

Hotels and destinations use this technology to create virtual showcases of their facilities and services, increasing customer confidence before booking.

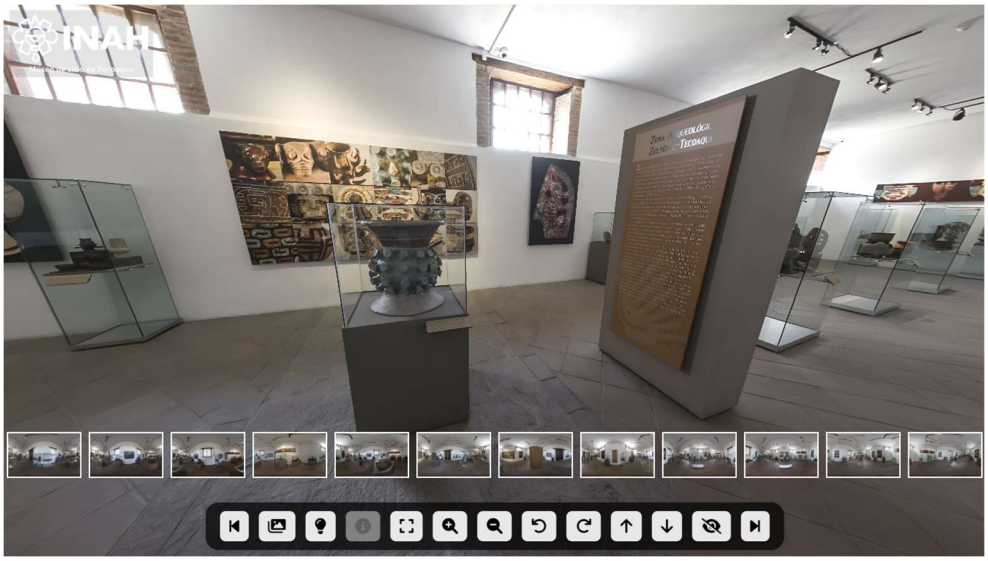

Image 1.1 – Virtual tour of the Tecoaque site museum.

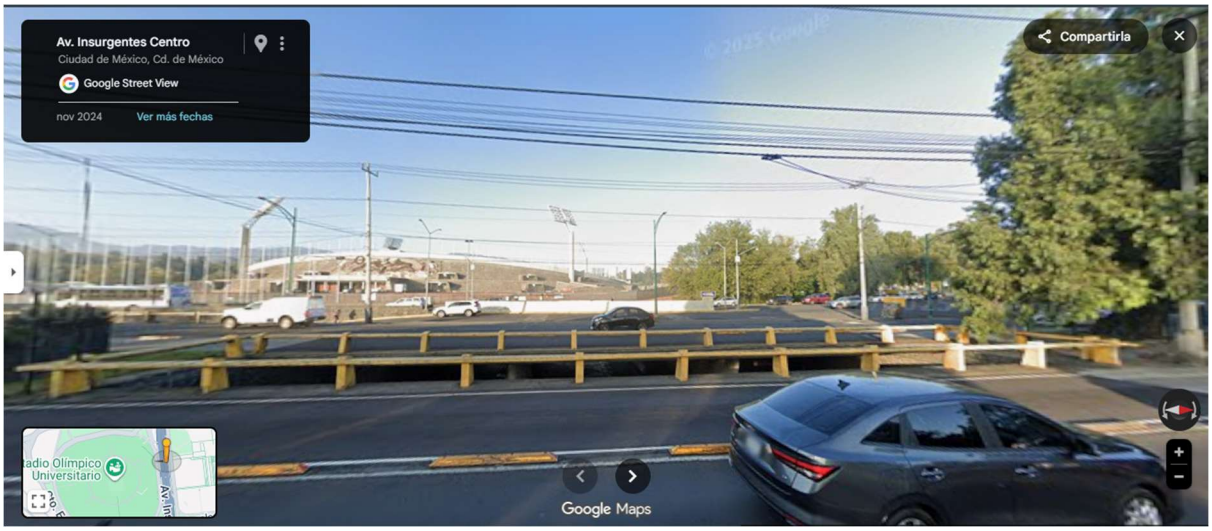

Image 1.2 – Virtual tour of the streets on Google Maps.

3.- Gantt Chart

	06-oct	7-OCT | 9-OCT	10-OCT | 17-OCT	18-OCT | 27-OCT	28-OCT | 5-NOV	6-NOV | 12-NOV	13-NOV | 15-NOV	16-NOV | 18-NOV	19-NOV | 24-NOV	25-nov
Inicio de proyecto	1 Dia									
Decision y aprobacion de fachada a implementar		3 Dias								
Visualizacion de contenido y entendimiento del software blender			8 Dias							
Implementacion de fachada en blender				10 Dias						
texturizado de la fachada en blender					9 Dias					
Integracion de la fachada de blender en Open GL						7 Dias				
Descarga y carga de modelos en Open GL							3 Dias			
Implementacion de animaciones solicitadas								3 Dias		
Creacion de documentacion (manuales y repositorio)									6 Dias	
Fin del proyecto										1 Dia

Image 1.3 - Gantt chart of the activities carried out during the project.

4.- Software flowchart and animation pseudocode

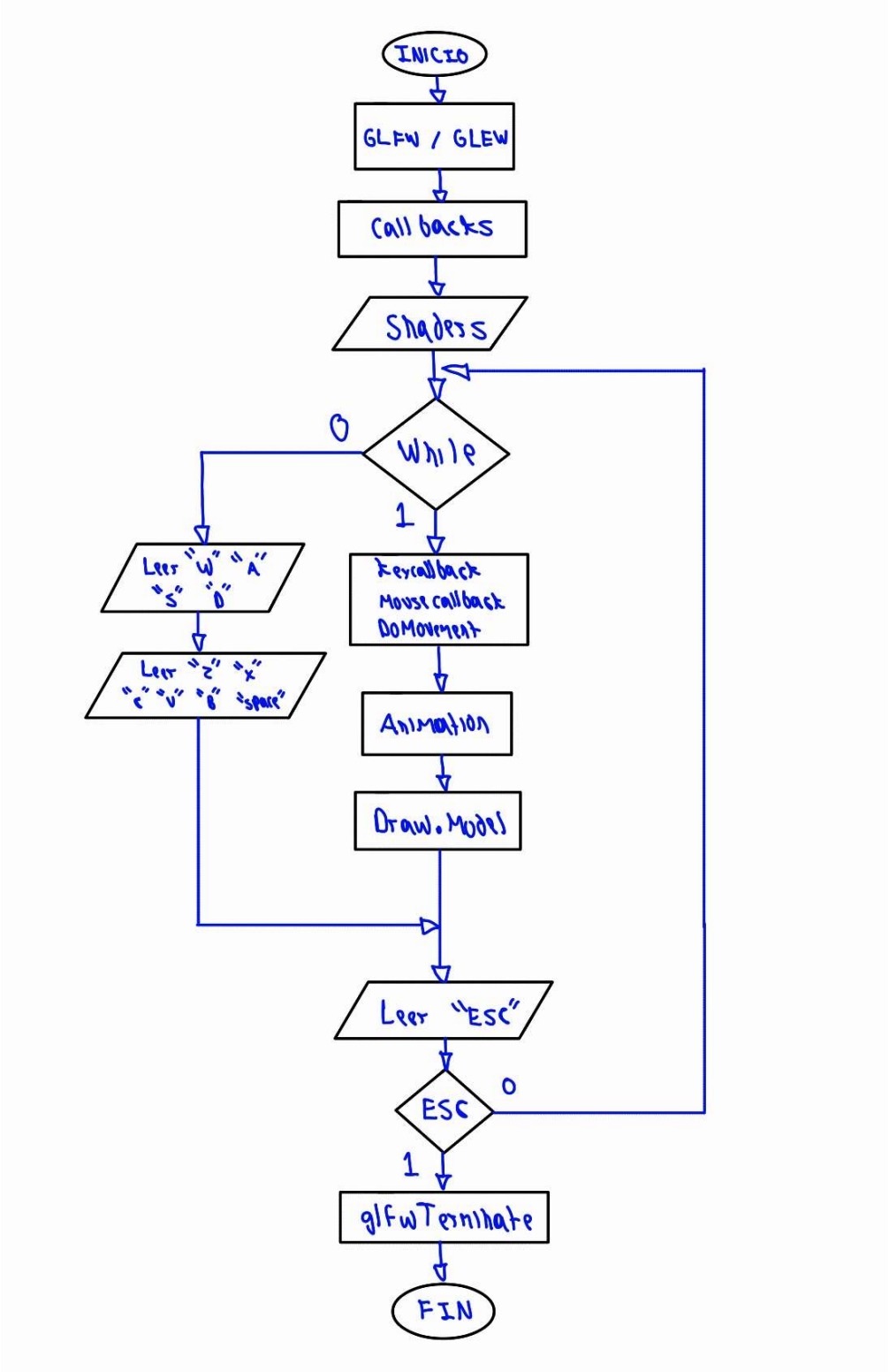

Image 1.4 – Project execution flowchart.

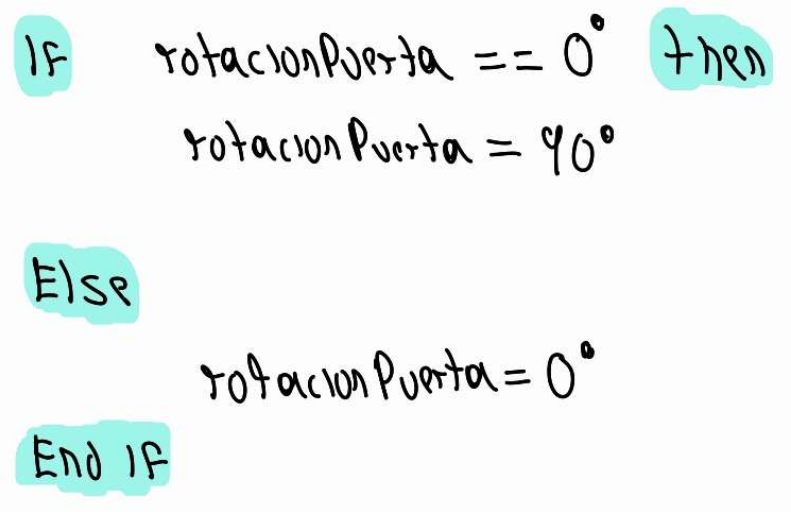

Image 1.5 – Pseudocode for the animation “Open or close door”.

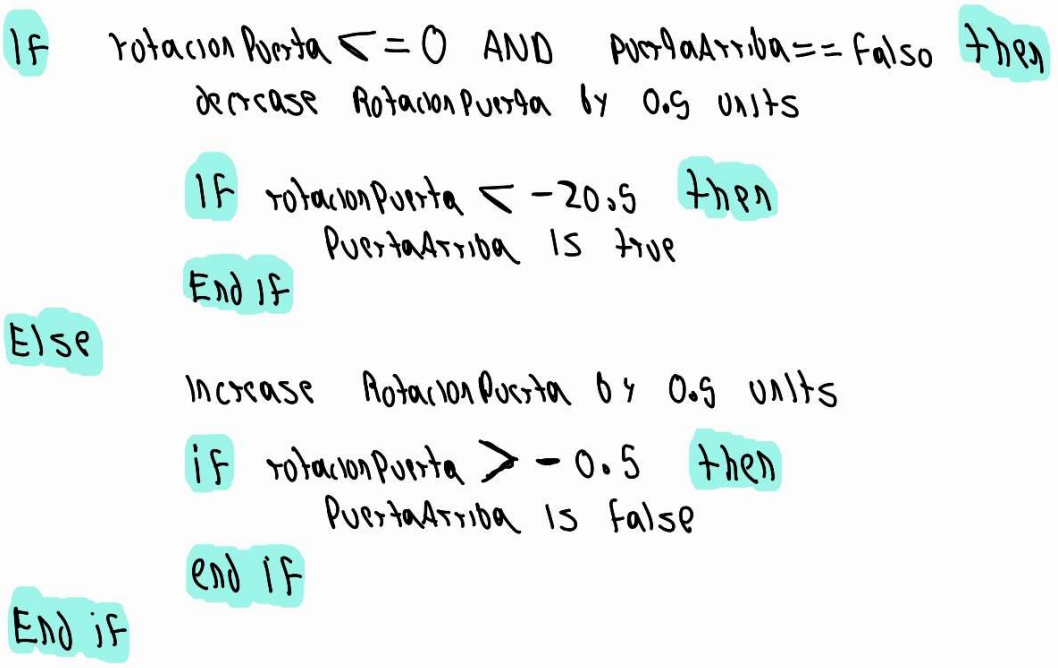

Image 1.6 – Pseudocode for the animation “Open or close the garage door”.

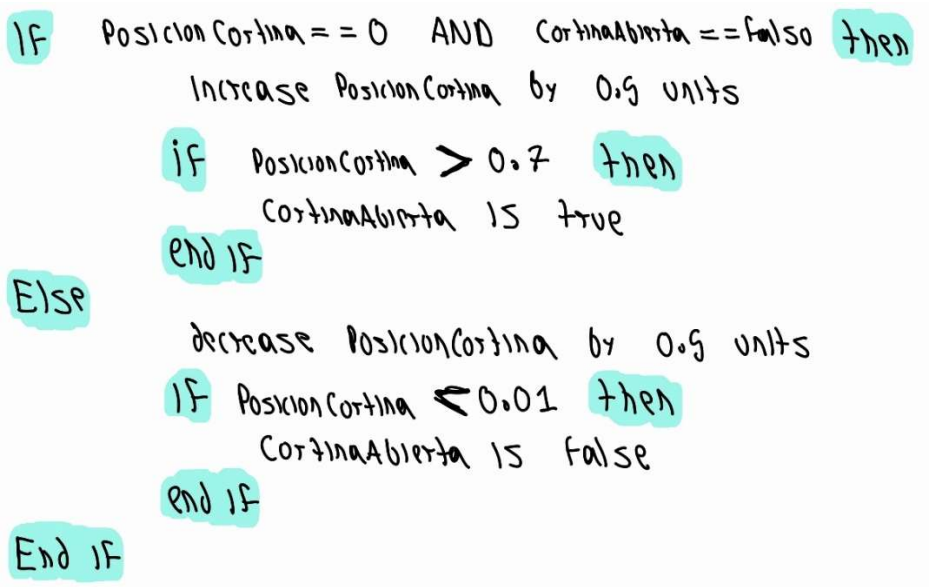

Image 1.7 – Pseudocode for the animation “Open or close curtain”.

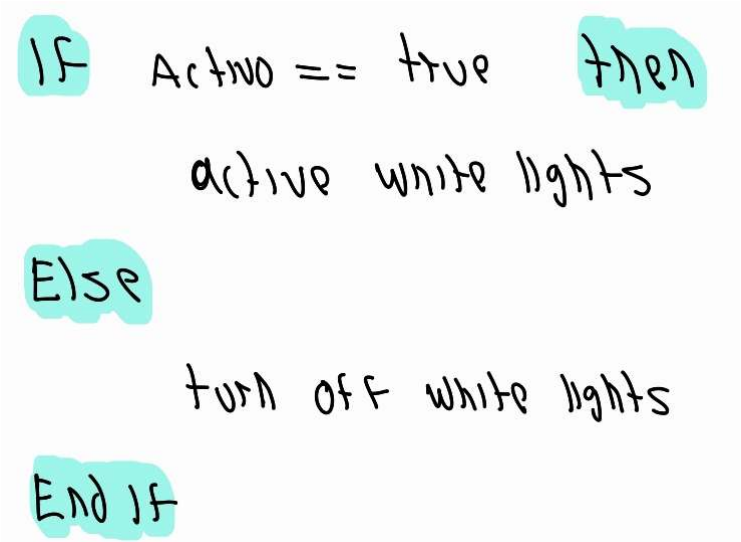

Image 1.8 – pseudocode for the animation “Turn lights on or off”.

```
Increase carriage rotation by 0.1 units
Increase timer "deltatime" times
Posicionllanta is 0.05 units.

if timer >= intervalo_cambio then
    posicionllanta is -0.05 units
    increase timer "intervalo_cambio" times.
end if
```

Image 1.9 – pseudocode for the animation “Move car”.

```
if rotacion == 0 then
    Move character on the z-axis
    rotate joint in a positive direction
Else
    Move character on the z-axis
    rotate joint in a negative direction
End if

if Personaje.z == 0.5 then
    rotate character 90° on the y-axis
End if

if rotacion.z == 90° then
    Move character on the X-axis
    rotate joint in a positive direction
Else
    Move character on the X-axis
    rotate joint in a negative direction
End if

if Personaje.X > 4.6 then
    stop animation.
end if
```

Image 1.10 – pseudocode for the animation “character movement”.

5.- Software Methodology

The methodology I applied to the project was Waterfall, a top-down approach. Different functions are developed in distinct stages, following a strict order. Before each stage, the product must be reviewed to ensure it's ready to move to the next phase. The initial requirements and specifications are not subject to change, so results aren't visible until the project is quite advanced.

I chose to implement this methodology because it best suited my needs. Since my team consisted of no more than two members, I didn't need progress meetings with other colleagues, as is common in other agile methodologies. The project was individual and depended solely on my own progress.

That's why the activities in the Gantt chart are sequential. First, I decided which facade to implement, then I learned to model in Blender, and proceeded to model the house, texture it, and load it with the OpenGL libraries. I left the process of furnishing the house with downloaded models for later. Finally, in the code, I implemented the requested animations, taking into account the available objects and the 3D space where each animation would be created.

6.- Project scope

This project only covers the exterior design of the protagonist's house in "Grand Theft Auto: San Andreas." In this case, I focused on developing the front view of the house, specifically the white door and the small hallway on the left side, as well as the garage section belonging to the protagonist's house.

Image 2.1 – reference to the created facade.

Image 2.2 – Model of the facade implemented in the project.

For the interior of the house, I only developed two rooms, the kitchen, and the living room. The kitchen contains the most important visual elements, such as the stove, refrigerator, and dish cabinet, which were created in OpenGL. The other elements were modeled in Blender or downloaded from Sketchfab and loaded into OpenGL.

Image 2.3 – kitchen reference on the facade.

Image 2.4 – Model of the kitchen made in the project.

The last room implemented was the living room. This was the most difficult room to furnish because it had the most items, and many of them were time-consuming to model. Therefore, I decided to create only the easiest objects using OpenGL. In this case, I modeled the plant located at the entrance to the kitchen, a table, a chair, and the framed photograph containing a landscape from the game using OpenGL. The other photographs and tables for the plant were created in Blender, and the remaining objects are models downloaded from Sketchfab.

Image 2.5 – reference to the facade room.

Image 2.6 – Model of the room created in the project.

7.- Limitations

The limitations of the project are obvious, as I decided to omit the upstairs rooms. Inside the house, I only included the decorative stairs, since they lead nowhere and bump into the walls.

For this project, I decided to focus solely on the two rooms on the first floor, as they are the first ones we encounter in the game, the most popular, the easiest to remember, and also the most recognizable in the house for those who have played the game.

Image 3.1 – decorative staircases implemented.

Image 3.2 – Stairs in the game leading to the upper floor, NOT implemented in this project.

Image 3.3 – The top part of the house is empty, just to decorate the exterior of the facade.

Image 3.4 – Upstairs rooms that were NOT implemented in the project.

Another factor to consider was the computer I used to develop the project. While my computer has decent resources, it lacks a dedicated graphics card. This means my processor and RAM are primarily used for graphics processing, which slows down the project compilation process.

Due to this limitation, I experienced delays in loading OpenGL models, and adding downloaded models took up to a minute and a half. This slowed down the project's development.

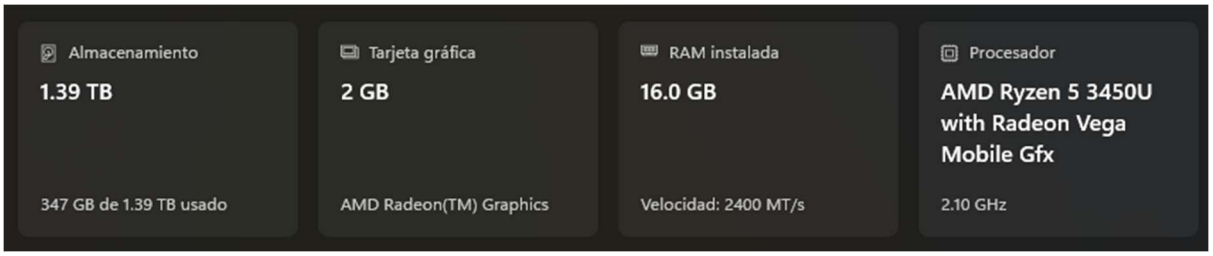

Imagen 3.5 – limited resources of the computer I used to carry out the project.

8.- Choice of Software used.

Blender

To model the house's facade, I used Blender. I decided to use it to model and texture the entire facade I created. Although other modeling programs exist, I preferred Blender because it's free, easy to install, and doesn't require creating an account before using it. There was also plenty of online content that provided the fundamentals of how to use the program. Furthermore, my friends who were also taking this course were going to use Blender to model their own facades, so we could support each other with any problems that might arise.

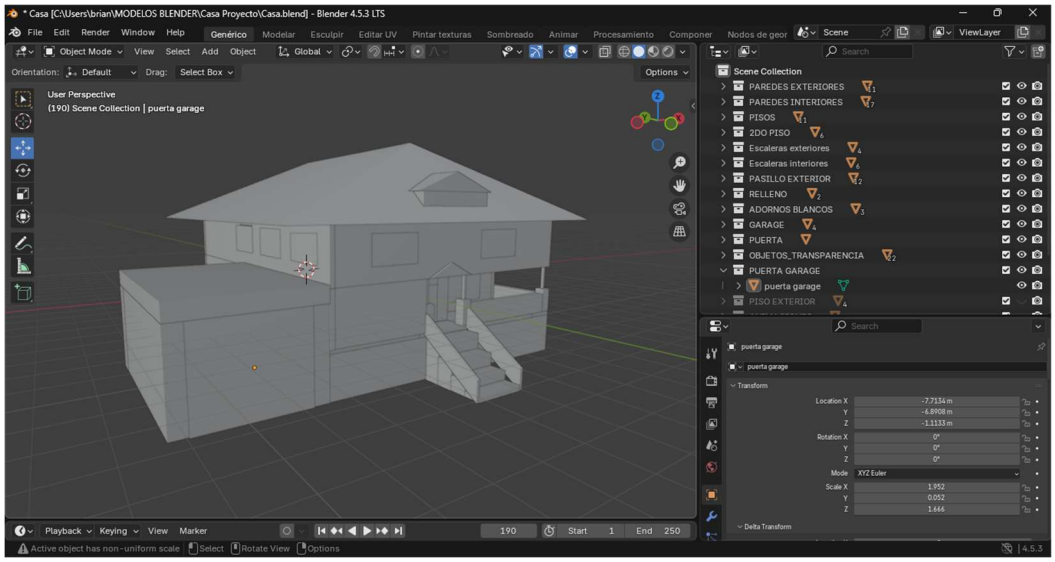

Image 3.6 – Modeling the facade in Blender.

Gemini PRO Image Generator

To texture a facade or any other model, you need images of a certain pixel size, so downloading images from the internet can be time-consuming, and you don't always find images with the specifications you need. That's why I decided to use Gemini PRO's image generator. I simply described the texture image I wanted and also provided an image from the game to give it a better understanding of what I wanted it to create.

The only problem I encountered was that Gemini took increasingly longer to generate the texture images I requested, and sometimes the image it generated was completely different from what I asked for or what I provided as a reference.

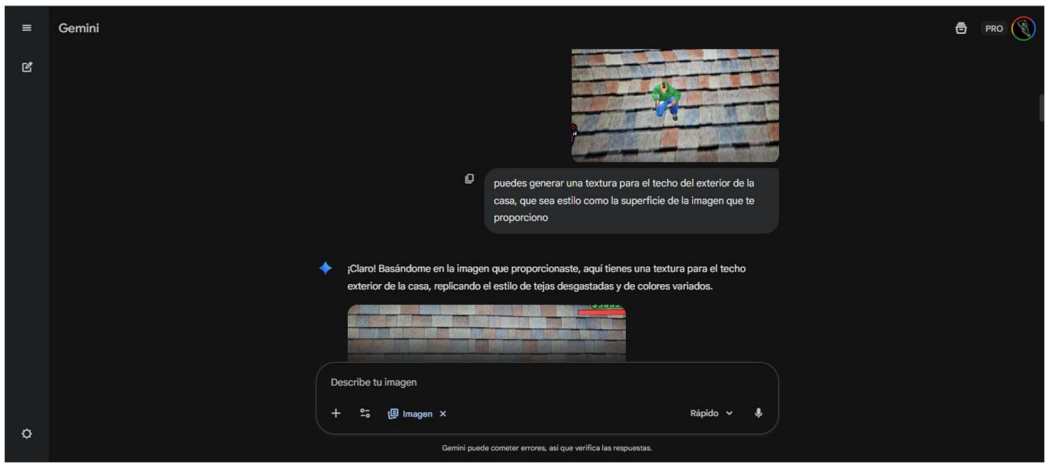

Image 3.7 – generating textures with Gemini PRO.

Gimp

Often, the images generated by Gemini had details I wanted to remove or modify, so I needed to edit them. For this, I used GIMP, an image editing software. I chose this software because it's free, easy to download, and doesn't require an account to use. Furthermore, it's an image editor I've used many times before, so I didn't have to look for resources or references online. It's a very powerful software that allowed me to make all the modifications I wanted to the images, such as adding an alpha channel to images that required transparency, removing watermarks from Gemini images, changing the color of textures, changing the pixel size, or simply changing the image format.

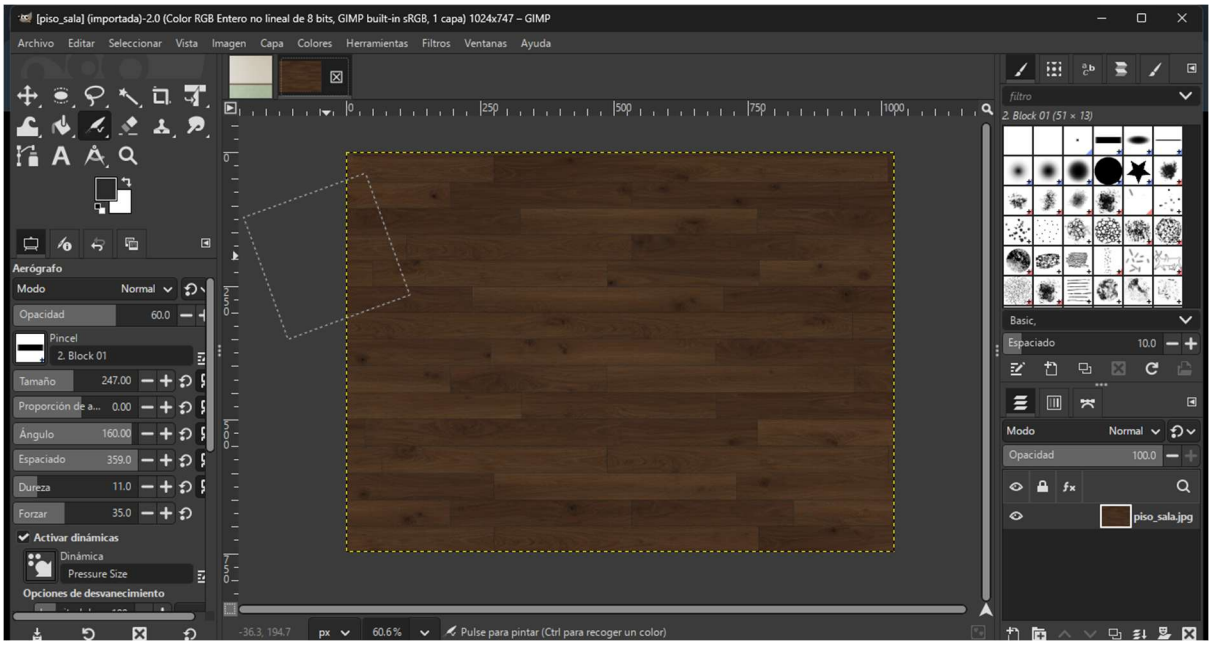

Image 3.8 – editing textures with Gimp.

9.- Project cost assessment.

To estimate the total project cost, the following aspects were considered: the programmer's salary, the computer used for the project, and the use of paid licenses. In this case, the only paid license was Gemini PRO for generating texture images, as the other software, such as Blender and GIMP, is free and does not require any license fees.

Monthly payments for internet and electricity services were also considered. Finally, the physical location where the project would be developed was taken into account; in this case, an office located in Mexico City.

Product	**Monthly Amount**	**Total amount of the product**	**Description**	**Reference**
Programmer Salary	$25,000	$50,000	This is the average salary for a C++ programmer.	https://mx.computrabajo.com/
Equipment (Computer)	$\frac{valor\ computadora}{\frac{5(depreciacion)}{12\ (meses\ del\ año)} * 2(duracion\ proyecto)}$ $158.3	$316.6	Computer used to carry out the project.	https://www.walmart.com.mx/ip/laptop-gamer-dell-inspiron-ryzen-5-16gb-1tb-256gb-ssd-vega-8-dell-3515-v3/00070452111687
Gemini Pro	$395	$790	Gemini license for texture generation.	https://gemini.google/subscriptions
Internet Plan	$500	$1,000	Internet plan rate.	https://www.totalplay.com.mx/paquetes
Electricity plan	$500	$1,000	Electricity consumption plan rate.	https://www.cfe.gob.mx/hogar/tarifas/
Work office	$3,500	$7,000	Physical office where the project was developed.	https://www.regus.com/

To estimate the final cost, the sum of the products in the "Total Product Amount" field was performed, so the total cost of the project is $60,106.6, sixty thousand one hundred six point six Mexican pesos.

10. – Code Documentation

Functions

Function Type	Function Name	Parameters it Receives	Description
void	KeyCallback	GLFWwindow *window, int key, int scancode, int action, int mode	It handles keyboard input events (keys pressed/released). It is used for camera control, lighting, and state machine actions.
void	MouseCallback	GLFWwindow *window, double xPos, double yPos	It handles mouse input events (cursor movement). It is used to control the orientation (pitch and yaw) of the synthetic camera.
void	DoMovement	None	Process the current state of the keyboard (keys[]) to smoothly apply the camera's translation movement (W, A, S, D) using deltaTime.
void	Animation	None	It contains the main logic of the animation. It is responsible for updating the position and rotation variables (dogPosX, head, rotDog, etc.) in each frame, especially during playback (play == true).
GLunit	loadTexture	Char: texture file path	Load and configure textures from image files.
int	main	None	Main function of the application. It handles the initialization of GLFW/GLEW, the configuration of shaders and buffers, the loading of models, and the rendering loop (Game loop).

Variables

Variable Type	Variable Name	Description
const GLuint	WIDTH, HEIGHT	Initial dimensions of the window in pixels.
int	SCREEN_WIDTH, SCREEN_HEIGHT	Actual dimensions of the window framebuffer obtained at runtime.
Camera	camera	Instance of the Camera class that controls the view array and position of the camera in the 3D scene.
GLfloat	lastX, lastY	Last X and Y coordinates of the mouse cursor to calculate the displacement.
bool[1024]	keys	Array that stores the state (pressed/released) of all keys on the keyboard.
bool	firstMouse	Flag indicating whether it is the first mouse movement to correctly initialize the coordinates.
glm::vec3[]	pointLightPositions	Array that stores the positions in 3D space of the point lights in the scene.
glm::vec3	Light1	Vector that defines the color and intensity of a variable light in the scene (used for dynamic effects).
glm::vec3	lightPos	Position of an additional light in 3D space (usually a point light or spotlight, currently at the origin).

bool	active	Controls whether the lighting system (or a specific feature) is globally active or deactivated.
float	rotCarro	Rotation angle in degrees of the car around the Y axis.
float	rotPuerta	Rotation angle in degrees of the main door of the house.
float	rotPuertaGarage	Rotation angle or vertical position of the garage door.
float	posCortina	Horizontal movement of the curtains to open/close.
float	tamllanta	Displacement of the car's wheels for rotational effect (simulation of forward movement).
float	timer	Time accumulator for controlling intervals in discrete or cyclic animations.
float	FLegs, RLegs	Rotation angles for animating the character's legs and arms (front and back legs).
float	personajePosX, personajePosZ	X and Z coordinates of the character's position in the 3D world.
float	personajeRot	Rotation angle in degrees of the character around the Y axis.

float	posPuertaX, posPuertaZ	Additional position adjustments for the main door.
const float	INTERVALO_CAMBIO	Constant time interval (0.1 seconds) to control the rhythm in step animations.
bool	AnimCarro	Flag that activates/deactivates the car animation.
bool	AnimPuerta	Flag that activates/deactivates the main door animation.
bool	AnimGarage	Flag that activates/deactivates the garage door animation.
bool	AnimCortina	Flag that activates/deactivates the animation of the curtains.
bool	AnimPersonaje	Flag that activates/deactivates the character's animation.
bool	puertaArriba	Status indicating whether the garage door is up (open).
bool	cortinaAbierta	Status indicating whether the curtains are open.
bool	step	Auxiliary flag to control the alternating step in the character's animation (e.g., to alternate the movement of the legs).

GLfloat	deltaTime	Time elapsed between the current and previous frame in seconds (for FPS independent movement).
GLfloat	lastFrame	Time in seconds of the last rendered frame (used to calculate deltaTime).
float[]	vertices	Array with vertex data for generic cubes or planes (position, normal, UV coordinates).
GLuint[]	indices	Array with indices for efficient triangle rendering using EBO.
GLuint	VBO	Vertex Buffer Object - Stores vertex data in GPU memory.
GLuint	VAO	Vertex Array Object - Stores the configuration of vertex attributes (pointers, stride, offset).
GLuint	EBO	Element Buffer Object - Stores the indices for element-based rendering.
Shader	lightingShader	Phong's lighting shader program for objects (the complex shader).

Shader	lampShader	Simple shader program to visually represent light sources.
glm::mat4	projection	Perspective projection matrix to transform 3D coordinates to 2D (Clip Space).
glm::mat4	view	View matrix that defines the position and orientation of the camera.
glm::mat4	model	Model matrix for transformations of individual objects before they are drawn.
glm::mat4	modelTemp, modelTemp1, modelTemp2	Temporal matrices used for the composition of transformations in hierarchical modeling.
float	velocidad	Base speed for movements and animations (used to modulate deltaTime).
Model	Piso_Exterior, Skybox, Casa, etc.	Instances of 3D models loaded from OBJ files for rendering.
GLuint	madera, marco, hoja, etc.	Identifiers of OpenGL textures loaded from image files.

11. – Conclusions

In this final project, I managed to integrate each of the concepts we covered in each of the theory and lab classes. To integrate all the functionalities, I started with the state machine code provided in the lab, as I considered it the most complete for adding or removing features. In carrying out the project, I not only used OpenGL, but I also had to use other 3D modeling tools like Blender, and image manipulation software like GIMP. To create the images I used as textures, I had to use artificial intelligence, specifically the Gemini image generator. I provided images taken with my phone from the game, and Gemini generated the textures.

Finally, to avoid having to model the entire environment myself, I downloaded models from Sketchfab. Many of these were not well textured, and I had to manipulate their UV coordinates so they would display correctly in OpenGL. Finally, I am grateful that the theory and laboratory professors allowed free implementation of the project, since this project not only involves technical aspects of the tools used, but also depends on the creativity that the student puts into their project.

12. – References

Blender Flow [@BlenderFlow]. (s/f). *Como aplicar texturas fácil !!* | | *Blender 3.1* [[Object Object]]. Youtube. Recuperado el 15 de noviembre de 2025, de https://www.youtube.com/watch?v=ik9Ldysa6zE

Casa de los Johnson. (s/f). Grand Theft Encyclopedia; Fandom, Inc. Recuperado el 15 de noviembre de 2025, de https://gta.fandom.com/es/wiki/Casa_de_los_Johnson

DansterDev [@DansterDev]. (s/f). *¡Modelar edificios con interior fácil y sencillo en blender! Explicado paso a paso!* [[Object Object]]. Youtube. Recuperado el 15 de noviembre de 2025, de https://www.youtube.com/watch?v=Q1lilIIriAs

de Costas, A. [@AlejandrodeCostas]. (s/f-a). *Cómo exportar modelo a OBJ con texturas en Blender - Tutorial guardado en formato Wavefront OBJ 3D* [[Object Object]]. Youtube. Recuperado el 15 de noviembre de 2025, de https://www.youtube.com/watch?v=Oes-ecXv8nc

de Costas, A. [@AlejandrodeCostas]. (s/f-b). *TUTORIALES BLENDER - Cómo hacer un mapeado de texturas - Mapeo UV Unwrapping en español Parte 12* [[Object Object]]. Youtube. Recuperado el 15 de noviembre de 2025, de

https://www.youtube.com/watch?v=Ek-Y62D9W4k

Interactivas y Computación Gráfica, T. [@ArturoVMS]. (s/f). *Integración de GitHub Desktop con el Proyecto de Computación Gráfica* [[Object Object]]. Youtube. Recuperado el 15 de noviembre de 2025, de https://www.youtube.com/watch?v=XZYgHmnB1vU

Metodologías de desarrollo de software: ¿qué son? (s/f). Santander Open Academy. Recuperado el 15 de noviembre de 2025, de https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html

Nisakai [@nisakai_420]. (s/f). *GUÍA RÁPIDA: ¿Cómo modelar en Blender 3D? 🤓 | Tutorial de blender en español 👌* [[Object Object]]. Youtube. Recuperado el 15 de noviembre de 2025, de https://www.youtube.com/watch?v=_f_6QN_G3bk

Downloaded Templates

Grand Theft Auto San Andreas - Balla - Download Free 3D model by Vesauq• (@Trevoncuz2.0). (2025, junio 15). : https://sketchfab.com/3d-models/grand-theft-auto-san-andreas-balla-cd393ab3131c4160a31be25a5c771ae8

Books and magazines - Download Free 3D model by Naira (@naira001). (2023, diciembre 27).: https://sketchfab.com/3d-models/books-and-magazines-d0b76eada5bd495abcdfb2b20e6f7ee6

Ddiaz Design (Director). (2025, enero 4). *1988 Lamborghini Countach 5000 QV - Download Free 3D model by Ddiaz Design (@ddiaz-design)*. : https://sketchfab.com/3d-models/1988-lamborghini-countach-5000-qv-884741220ecc44a886f9232f7888a7f1

Deharo, Y. (Director). (2018, julio 26). *20 Liquor bottles - Download Free 3D model by Yannick Deharo (@YannickDeharo)*. : https://sketchfab.com/3d-models/20-liquor-bottles-4729b1cfa5b74db68924190242f8ac76

Dish Rack with Dishes - Download Free 3D model by HippoStance. (2020, mayo 6). : https://sketchfab.com/3d-models/dish-rack-with-dishes-ce69014964624fbb808a028e5c55d88b

Elena, F. F. (Director). (2020, septiembre 3). *Saintpaulia flowers low-poly - Download Free 3D model by Elena FF (@elenaferfor)*. : https://sketchfab.com/3d-models/saintpaulia-flowers-low-poly-6f78eeed71b74755b3fc14f1ac5924ca

Mesa minimalista para TV - Download Free 3D model by Draxter_pendragon. (2025, julio 16). : https://sketchfab.com/3d-models/mesa-minimalista-para-tv-badfe2c2c5a7494cac5bd052145f8f94

Old Television from 90's - Download Free 3D model by Zgon (@Z-gon). (2022, febrero 22). : https://sketchfab.com/3d-models/old-television-from-90s-5642074408e94782a7f2ee545e846b45

Plate and cup - Download Free 3D model by morrrtu1o. (2023, noviembre 19). Serrano, V. H. F. (Director). (2020, diciembre 17). : https://sketchfab.com/3d-models/plate-and-cup-7c2c9df4bcfb4dfbaa98836af4333f9e

Latas_Coursera - Download Free 3D model by Victor Hugo Franco Serrano (@VictorHugoFrancoSerrano). : https://sketchfab.com/3d-models/latas-coursera-3768161dbb7b4f91907934c7fcca7961

Shekh, A. (Director). (2020, octubre 17). *Pine tree - Download Free 3D model by Andriy Shekh (@sheh5262)*. : https://sketchfab.com/3d-models/pine-tree-e52769d653cd4e52a4acff3041961e65

Sillas - Download Free 3D model by Elbolillo (@Elbolilloduro). (2021, abril 9). : https://sketchfab.com/3d-models/sillas-c9e85c037309413299b30dd666ab27c7

Sillón Le Mans - Download Free 3D model by Pablo.Portela. (2020, agosto 13). : https://sketchfab.com/3d-models/sillon-le-mans-79e6c35685a24da6899d884fa470df54

Three_Seater_Sofa - Download Free 3D model by Nelesh_surve. (2025, abril 12). : https://sketchfab.com/3d-models/three-seater-sofa-bedd882faf354f87a28abcf93c718a3d