Universidad Nacional Autónoma de México

Facultad de Ingeniería

Materia:

Computación grafica e interacción humano-computadora

Alumno:

318285553

Profesor:

Carlos Aldair Roman Balbuena

Semestre:

2026-1

Grupo:

05

Entregable:

Proyecto Final (Manual Técnico)

Index

Objective

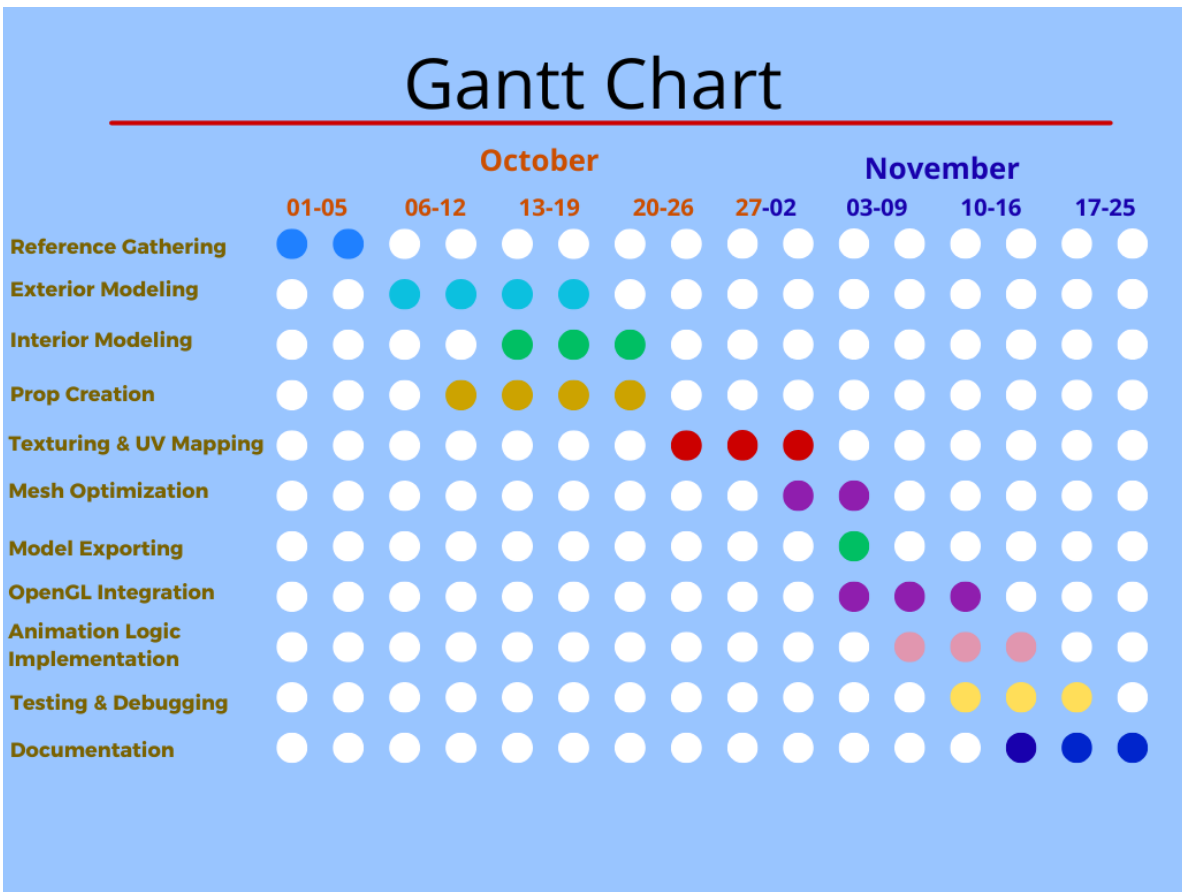
Scenario Recreation: Model and render the house from the Billy & Mandy cartoon using the provided tools, including 2 rooms (living room and bathroom) with 5 objects in each room (towel, toilet, bathtub, curtain, rug, television, armchair, lamp, portrait, hanging lamp), with their respective textures and normal maps.

Lighting Implementation: Develop a Phong lighting system that includes different types of light sources to add realism to the scene.

Animation Development: Program dynamic behaviors in objects, such as lighting or movement animations.

Software Independence: Generate a final executable that is functional and portable, managing dynamic library dependencies for execution in another environment.

Gantt Chart



This project had a timeline from October 01 to November 25, 2025, during which different tasks were carried out, which were:

Image Search (October 01-05): The initial phase of this project was the search for models to create; the Billy and Mandy cartoon was chosen as inspiration for the execution of this project. In this search, the necessary images were obtained to model the facade, living room, bathroom, and objects.

Creation of Exteriors (October 13-23): During this period, Blender was chosen as the modeling tool for the project. Different tests were carried out to understand how Blender works, and the exterior modeling of the house began, which included: windows, doors, chimney, and walls.

Creation of Interiors (October 10-23): Continuing with the modeling in Blender, the interiors of the house were modeled, where the door frames, the separation by rooms, and the floors and ceilings of each one were created.

Application of Textures (October 23 - November 02): The texturing process began upon the completion of the exterior modeling to facilitate the model's visualization; the textures were designed in Paint, Gemini, and extracted from the internet.

Modeling Correction (October 28 - November 05): Once the modeling of the house was finished, errors such as vertex duplication, location errors, or object deformations were reviewed. Additionally, each of the object faces was triangulated, and scale and rotation corrections were made.

Exporting Models (November 04-05): The models were exported from Blender using an .obj format, with the objective of visualizing them in OpenGL.

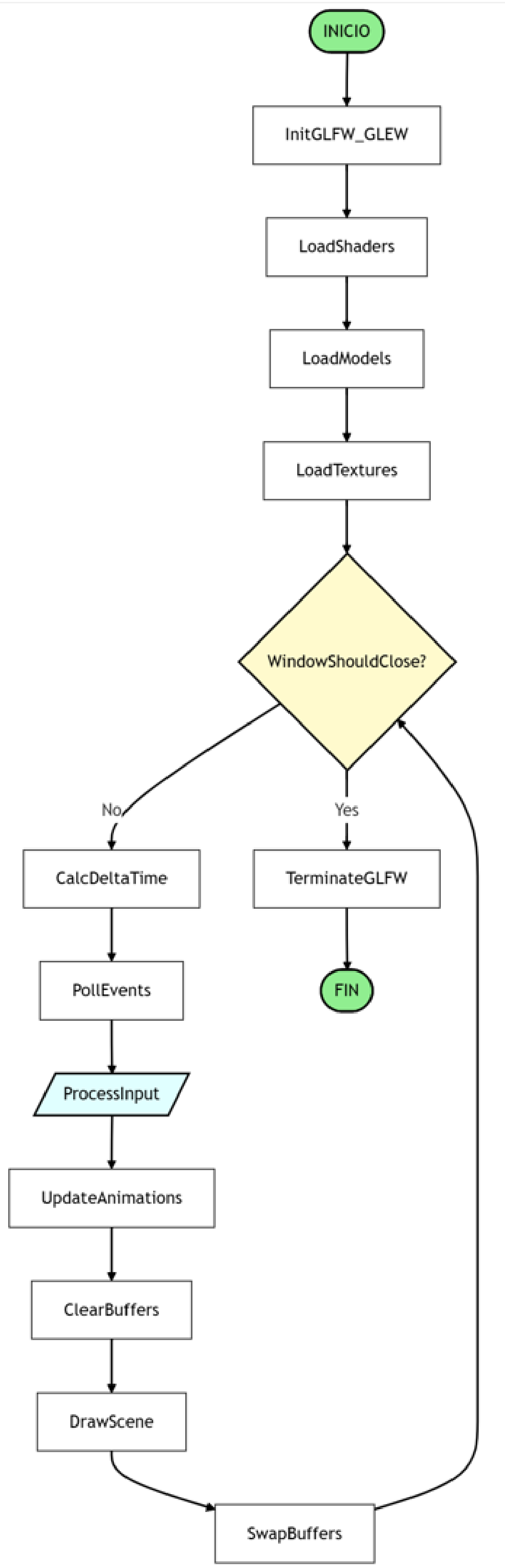
Integration in OpenGL (November 05 - 12): The created models were adapted so they could be visualized correctly in terms of texture, location, and position, in addition to the models obtained from the internet.

Animation Programming (November 07 - 16): The animations performed, such as the lamp movement, turning on the lamp, and door opening, were carried out with the objects extracted from Blender. Subsequently, for the animation of Billy walking and the specter coming out of the television, the models were extracted from the internet and loaded into the code with their due textures and position corrections.

Testing and Adjustments (November 10 - 20): Tests were carried out to verify that the position of the objects was correct and had their due adjustments; additionally, each of the errors present, such as lighting and textures, were corrected.

Documentation (November 10 - 25): This task was carried out once the practical part of the project was completed, to have a comprehensive view of everything that was accomplished.

Flowchart



INICIO
InitGLFW_GLEW
LoadShaders
LoadModels
LoadTextures
WindowShouldClose?
No
Yes
CalcDeltaTime
TerminateGLFW
PollEvents
FIN
ProcessInput
UpdateAnimations
ClearBuffers
DrawScene
SwapBuffers

Scope

The project is developed in a real-time visual graphic application, focused on the recreation of a specific scenario: the Billy and Mandy house, and the implementation of computer graphics techniques.

This system allows the end user to perform different actions within the virtual environment:

Free navigation: Exploration of the scenario in first person through keyboard and mouse movement with freedom of movement in the different axes.

Environment interaction: Ability to modify the state of specific objects through keyboard events.

Animation visualization: Observation of dynamic behaviors, both cyclical (characters walking, lamp oscillating, etc.).

The software implements a graphics engine based on OpenGL with the following technical capabilities:

Resource loading: Import of 3D models, images, and animated objects; Phong lighting pipeline, supporting one directional light, multiple point lights with quadratic attenuation, and 1 Spotlight type light coupled to the camera.

Animation system: Calculation of model matrices in CPU for procedural animations (translation, rotation, scale).

Skeleton-based animation: Implementation of Hardware Skinning in the Vertex Shader for the deformation of meshes based on bones.

Limitations

The system presents some technical and functional restrictions such as:

Platform and software restrictions: The application is compiled for Windows environments; however, it is not compatible with other environments such as macOS or Linux.

Object physics: The objects lack a solid structure that prevents users from passing through or allows viewing them from the inside.

Assimp Sensitivity: ASSIMP is sensitive to formats other than JPG, which can generate display errors or missing textures.

Measurements: The facade that was implemented did not have real measurements for the construction of the house, so an estimated calculation of the measurements and of each of the objects had to be made.

Modeling: The lack of experience in tools like Blender complicates the creation of models in an advanced way, which limits the ability to achieve better shapes and correct scales.

Methodology

The project's primary methodology was the Waterfall Model. This linear approach allowed for orderly progress through well-defined phases, ensuring that each stage was completed before proceeding to the next:

Phase 1: Requirements Analysis:

Functional Requirements: The needs for a navigable, interactive 3D environment with both simple and complex animations were identified.

Theme Selection: The scope was visualized based on the Billy & Mandy series, determining the list of 10 objects across 2 rooms: Living Room (sofa, television, lamp, hanging lamp, and portrait) and Bathroom (toilet, bathtub, curtain, towel, and rug) to model and comply with the rubric.

Phase 2: System Design:

Scene Design: A facade and interiors were designed in Blender as realistically as possible, constructing each component element such as windows and doors.

Class Design: The program was structured by separating logic into specialized classes for Camera, Shader, Model, and ModelAnim.

Interaction Design: Key mappings were defined for navigation using (WASD) and for triggering events with different keys.

Phase 3: Implementation:

Configuration: Setup of a virtual environment, initializing windows with GLFW and loading extensions with GLEW, creating a rendering engine with GLSL shader programming to support Phong lighting texturing (Directional, Point, and Spotlight).

Animation Logic: Implementation of mathematical algorithms for geometric transformations (model matrices).

Phase 4: Testing:

Visual Validation: It was verified that the lighting was correct and that objects were visualized in the best possible way.

Animation Testing: Animation tests were performed, verifying the fluidity of transitions and the correct execution of the complex sequence, as well as assessing the fluidity of each animation in accordance with the use of deltaTime.

Phase 5: Deployment and Maintenance:

The final phase focused on generating a deliverable product; a technical manual and a user manual were created, documenting everything developed during the project.

Tools and Resources Used:

Visual Studio 2022: This was the main development environment, where all the project code was written and compiled. Here, the necessary external libraries (GLFW, GLEW, Assimp) were configured so that OpenGL would function correctly. Additionally, it served to generate the final executable, and its debugging tools helped detect errors in the code when the program did not run.

Blender: This tool functioned for the creation of each of the models, including the construction of the house structure, its windows, and each of the doors. It was also used to model the objects found inside, which were: armchair, television, frame, floor lamp, hanging lamp, cup, bathtub, towel, rug, and curtain. Each of these objects had its own texture applied in Blender, correcting the UV maps and assigning materials. Additionally, specific tools were used such as beveling, basic transformations, face extrusion, loop cuts, merging vertices by distance, triangulating faces, and finally, exporting the models in .obj format.

Gemini and ChatGPT: The use of these tools served a support role in various areas: they helped generate ideas for some textures, resolve doubts about C++ programming syntax, and the correct export of .obj files. They also served to explain in detail specific Blender functions that I was unaware of and to optimize the logic of some objects that were already constructed.

Paint: This program helped to edit and create new textures, as well as adapt images extracted from the internet or generated by AI. It was useful for saving images in the appropriate formats and ensuring they had the correct dimensions so there would be no problems when loading them into OpenGL.

YouTube: This tool was used as the platform where the video demonstration of the house was uploaded once finished. It also served as a source of knowledge where tutorials were watched that helped understand the functioning of Blender, as well as each of the functions used to improve the modeled objects.

Cost Analysis

Production cost

Field	Amount (MXN)	Duration	Description	Subtotal	Evidence	Actual Expense
Lead Developer Salary	$12,308	2 months	Jr. Programmer Salary.	$24,616.00	Reference 10	No (Time Investment)
3D Asset Licensing (Billy & Ghost)	$5,500.00	One-time Payment	Estimated cost for purchasing commercial rights from freelance modelers for the characters.	$5,500.00	Reference 1 y 2	No (Assets used for academic purposes)
Technical Consulting & Shaders	$6,000.00	One-time Payment	Payment for the graphics engine (ModelAnim classes, complex Shaders) developed by third parties (Senior Dev/Professor).	$6,000.00	Reference 11	No
Computer Equipment	$16,499.00	2 months	Development equipment wear and tear (3-year depreciation).	$916.61.00	Reference 12	Yes
Services & Licenses	$1,000.00	2 months	Electricity, Internet, and AI tools (ChatGPT/Gemini) for texturing.	$2,000.00	Reference 13, 14 y 15	Yes
Office Rent	$4,100.00	2 months	Office workspace for the developer.	$8,200.00	Reference 16	No

Total production cost: $47,232.61

Using a 30% profit margin for taxes, technical support, and profit, results in an estimated sale price of:

$61,402.39

The final price of $61,402.39 MXN was established considering the technical complexity of this development. Being a project built directly in C++ and OpenGL (instead of using commercial engines like Unity or Unreal which facilitate the work), the value increases for the following reasons:

Optimized Performance: Since the software does not depend on a heavy graphics engine, the final executable is much lighter and more efficient.

Engineering Work: The price covers the manual programming of complex components, such as custom shaders and bone animation logic (skeletons), which requires specialized programming knowledge, not just design skills.

Licenses and Rights: This price includes the payment to the original creators of the models (Billy and the Specter) and to the developers of the libraries.

Features

This project features functions developed in the C++ language and OpenGL; its main characteristics are:

Scenario Recreation: The house from the Billy & Mandy cartoon was recreated, modeling both the exterior facade and the interiors.

Objects: The scenario includes original 3D models (created in Blender) such as the house, bushes, doors, armchair, lamps, television, portrait, cup, bathtub, curtain, towel, and rug. Adapted external models were also integrated, such as the Specter and the character Billy.

Full Texturing: All models have their own textures and UV maps to add color and detail to materials such as wood, fabric, and vegetation.

Lighting:

Phong Model: A lighting system was implemented that allows objects to react to light and shadows on their faces.

Light Sources: The scenario operates with two types of simultaneous light sources:

Directional Light: Simulates the sun.

Point Lights: Lamp bulbs that illuminate in all directions within a established range.

Interaction and Camera:

Free Navigation: The user can move freely throughout the house and garden using a first-person synthetic camera controlled by the mouse and WASD keys.

Environment Control: Interactions were programmed so the user can open and close room doors, as well as turn the lamp light on and off.

Animations:

Skeletal Animation (Skinning): Loading of bone-animated models was implemented using the character Billy, who performs a walking cycle in the garden.

Code-based Animations: Mathematical animations were created for static objects, such as the lamp wobbling and the rotation of doors when opening.

Special Events: A complex sequence was programmed on the television, where pressing a key changes its texture to that of a portal, and a 3D model (the ghost) emerges with movement.

Implemented Animations

1.- Door Animation: For the door movement, a complex function was not used; instead, a basic linear interpolation was employed. The logic consists of increasing or decreasing the rotation angle until a limit is reached.

Logic: A flag is used to determine whether to open or close.

$$Angle = Angle_{current} + (Speed\ x\ deltaTime)$$

Implementation: Three transformations were applied in a specific order to simulate a hinge:

Translation to origin: The door is moved so that its axis of rotation, which is the hinge, is at the center point.

Rotation: Rotation is applied on the Y-axis using the calculated angle.

Translation back: The door is returned to its original position in the house.

2. Lamp Animation: To make the lamp wobble smoothly and continuously, the trigonometric Sine function was used.

Logic: By using execution time as a variable, the value oscillates smoothly between -1 and 1.

$$Angle = Amplitude + sine(Time\ x\ Velocity)$$

This generates a swaying movement between -5 and 5 degrees, creating the wobbling effect on the lamp.

3. Specter Animation: The appearance of the ghost coming out of the TV is a procedural animation that combines two simultaneous movements:

Linear Advance (Z-axis): The ghost moves out of the screen towards the front.

$$Pos_z = Pos_{ini} + (Time\ x\ Velocity)$$

Lateral Oscillation (X-axis): To make it look like it is floating, a left-to-right movement was added using Sine.

$$Pos_x = sine(Time\ x\ 8.0)x0.3$$

4. Billy Animation: This is the most complex animation in the project because, unlike the previous ones that move the entire object, this one deforms the character's mesh. The model has an invisible "skeleton" (bones). Each vertex of Billy's model is assigned to one or more bones.

Operation: The ModelAnim class reads the .dae file and extracts the keyframes. It calculates how much time has passed and interpolates the position of the bones between two movements. It sends an array of matrices to the Shader. The Vertex Shader multiplies each vertex by its corresponding bone matrix, allowing Billy to walk by deforming his legs and arms naturally.

Code Documentation

This project was elaborated in the OpenGL environment; it is composed of a main file where we initialize each of the components. In this section, we have the main logic of the animations and user interaction.

Shader.h : This helps to extract the reading, compilation, and connection of the .frag and .vs files.

Camera.h: This class is fundamental as it helps to create a synthetic camera, with which we can navigate in first person through the created models. Additionally, it processes keyboard and mouse input to generate a view matrix.

ModelAnim.h/meshAnim.h: These classes are a contribution from Professor Carlos Aldair Roman Balbuena, managing the loading of animated 3D models such as the Billy animation (movimiento.dae); this class is based on the bone hierarchy and calculates their transformation in each frame.

Shaders

This project is composed of different shaders which fulfilled a specific function in the project.

Lighting: It is a main shader that fulfills the function of illuminating objects following the Phong illumination model and textures in all types of light sources (directional, focal, and flashlight). Additionally, they facilitate the loading of static 3D models such as those created by own authorship (house, bushes, doors, armchair, lamps, television, portrait, cup, bathtub, curtain, towel, rug) and those extracted from various sources (Specter and Billy). These use Assimp libraries to read the files and assign their textures.

modelLoading: It was used for the basic texture of the house and the doors.

anim: Applies the functioning of the animated character's skeleton skin by multiplying the vertex position by the matrix position, performing a weighted sum of the transformations of the bones affecting the vertex.

lamp: This shader draws a solid color and generates a light used for debugging.

Glossary of functions

Function	Type	Parameters	Description
main	int	void	It is the main function of the program. Here, the libraries are initialized, models, textures, and shaders are loaded, and the main loop (Game Loop) where everything is drawn is executed.
KeyCallback	void	GLFWwindow* window, int key, int scancode, int action, int mode	This function helps to detect when we press a key just once. It serves to activate "switches" like opening doors, turning on the light, or activating the TV animation.
MouseCallback	void	GLFWwindow* window, double xPos, double yPos	It registers the movement of the mouse in the window. It serves to calculate where we are looking and update the first-person camera.
DoMovement	void	void	This function detects if we keep a key pressed. It is used for the continuous movement of the camera (WASD) so that the displacement is fluid.
Animation	void	void	Here is the mathematical logic of the animations. It calculates the opening angles of the doors, the wobbling of the lamp, and the time of the ghost animation.
ObjetoDraw	void	glm::mat4 base, glm::vec3	It is a function that helps to draw cubes (It receives the base matrix, the size, and the position to draw compound

		escala, glm::vec3 traslado, GLint uniformModel	objects without repeating so much code).

Variable Glossary

System and Window Variables: Help to control the initial configuration of the graphic window and the rendering time flow.

Name	**Type**	**Description**
WIDTH	const GLuint	Constant width of the window (1600 px)
HEIGHT	const GLuint	Constant height of the window (900 px)
SCREEN_WIDTH	int	Stores the actual width of the window framebuffer
SCREEN_HEIGHT	int	Stores the actual height of the window framebuffer
deltaTime	GLfloat	Variable that measures the time elapsed between the current frame and the previous one
lastFrame	GLfloat	Saves the time of the last processed frame

Input and Camera Variables: Variables used for image manipulation, as well as for each of the images or movements.

Name	**Type**	**Description**
camera	Camera	Allows navigation through the scenario simulating a first-person camera.
keys	bool	Array that records which keys are being pressed and which are not at any given moment.
lastX	GLfloat	Stores the cursor location on the X-axis of the previous frame to calculate camera movement.

lastY	GLfloat	Stores the cursor location on the Y-axis of the previous frame.
firstMouse	bool	Variable that prevents abrupt camera jumps when entering the window for the first time.
nKeyProcessed	bool	Auxiliary flag that prevents an action from repeating many times if a key is kept pressed.

Animation Variables (Doors): These variables help manage the angles and necessary states, as well as the logic required to open or close doors.

Name	**Type**	**Description**
doorE_angle	float	Stores the current opening angle of the Entrance door.
doorS_angle	float	Stores the current opening angle of the Living Room door.
doorB_angle	float	Stores the current opening angle of the Bathroom door.
doorE_open	bool	Indicates if the Entrance door should open or close.
doorS_open	bool	Indicates if the Living Room door should open or close.
doorB_open	bool	Indicates if the Bathroom door should open or close.

Animation Variables (Lamp): These variables will help control the simulated physical movement and the state of the light.

Name	**Type**	**Description**
lampWobble	float	Valor numérico que define qué tan inclinada está la lámpara Numeric value defining how inclined the lamp is.

lampAnimationActive	bool	Functions as a switch to activate or deactivate the lamp movement.
lampLightActive	bool	Logical variable that allows enabling or disabling the light control.
lampState	bool	Indicates if the lamp bulb is on (true) or off (false).

Animation Variables (TV): Variables that control the appearance of the ghost.

Name	Type	Description
tvAnimationActive	Bool	Main switch that starts the animation of the ghost exiting the TV.
tvAnimTime	float	Timer that counts the elapsed time of the animation to determine when to stop it.

State of the Art

The virtual tour developed is a simulated tracking because it is done digitally through programming in OpenGL and serves so that explorations can be carried out on the computer; the tour is done as if one were inside the cartoon in a physical space and serves to create forms, videos, and images in three dimensions. In this project, the movement of the camera can be manipulated, and objects can be seen moving or remaining static, working with this type of tour similar to how it is done in the video game industry or architecture.

Some of the main characteristics of this project are the following:

The simulation: Where a recreation of a place that is fictitious can be made, working with something that does not exist in the real world but does in animation.

Regarding technology: It works with real-time graphics rendering that allows a 360-degree visualization and was also programmed with images (textures) and models with movements (animations).

In the interactivity part: The user has the freedom to move freely throughout the house and the garden using the keyboard and mouse.

Access: It is from a computer equipment (PC) through an executable file that processes the graphics at the moment.

The main purpose: It serves to make explorations in modeled spaces, recreate images of the series, and demonstrate the functioning of lights and shadows remotely.

This virtual tour is a recreation of the Billy and Mandy scenario and is totally visual; it serves to simulate a real tour inside the house where the user can move freely, simulating movement as if walking and observing the recreation of forms, figures, and characters like Billy or the ghost where one can interact with them freely.

A virtual tour like this facilitates the recreation of a space that does not physically exist (a 2D cartoon turned into 3D) and can also be manipulated in the creation of desired elements, adding forms, colors, lighting, and movements according to the project's needs.

Conclusion

This project is a compilation of everything learned throughout the semester, creating an interactive virtual environment based on the adventures of Billy & Mandy. Through a rendering tool, the objective of integrating modeling, texturing, lighting, and animation in real-time was met.

Various fundamental OpenGL tools were used in this project. Developing a model from scratch in Blender was interesting, as it is a world that generates infinite possibilities; learning to use this tool was very positive because it helps me visualize a new area of professional development.

Mastering graphics pipelines is somewhat complex, but they allow control over visualization, achieving detailed lighting effects using multiple light sources that provide depth and realism. On the other hand, animations help observe how mathematical functions can create non-linear trajectories and dynamic graphic behaviors.

Finally, this project helped me see 3D graphics in a different way, especially when analyzing older animations where I can now detect geometry, lighting, or texturing errors. However, over time, new tools have helped us improve graphically to the point of having visual projects where the line between realism and what is generated by a graphics engine is increasingly thin.

References

Link where the ghost model used in the project was obtained:

1.-Cruz Ponce, A. A. (2021, 17 de abril). Fantasma [Modelo 3D]. Sketchfab. https://sketchfab.com/3d-models/fantasma-8928e3d13773497291605e5ef1fc0aed

Link where the billy model used in the project was obtained:

2.-calvinwil5782. (2023, 13 de noviembre). Billy [Modelo 3D]. Sketchfab. https://sketchfab.com/3d-models/billy-04a755991fee4d99ba6027a80e6aeb65

3.-Integración de GitHub Desktop con el Proyecto de Computación … [Video]. (n.d.). YouTube. https://www.youtube.com/watch?v=XZYgHmnB1vU

4.-Animación Básica en OpenGL [Video]. (n.d.). YouTube. https://www.youtube.com/watch?v=Iuybf6bTzhc

5.- Carga de Texturas en OpenGL Introducción a las UV s [Video]. (n.d.). YouTube. https://www.youtube.com/watch?v=21jOb3w_T8k

6.-Animación por Keyframes en OpenGL: Movimiento y Rotación [Video]. (n.d.). YouTube. https://www.youtube.com/watch?v=5l1A6Lxa3GQ

7.-IEP. (2022, 7 de julio). Metodología Waterfall: Modelo de gestión de proyectos en cascada. IEP. Recuperado de https://iep.edu.es/metodologia-waterfall/

8.-Universidad de Valladolid. (s. f.). Metodología Waterfall (modelo en cascada). Universidad de Valladolid. Recuperado de https://uvadoc.uva.es/bitstream/10324/71366/1/TFG-G6918.pdf

9.-Rominaescalonab. (s. f.). 59 ideas de Billy y Mandy [Tablero de imágenes]. Pinterest. Recuperado de https://cl.pinterest.com/rominaescalonab/billy-y-mandy/

10.- Indeed. (2025, 14 de noviembre). *Sueldo de Programador/a junior en México* [Página web]. https://mx.indeed.com/career/programador-junior/salaries

11.- Indeed. (s. f.). Sueldos de CONSULTORIA TI en México: ¿Cuánto paga CONSULTORIA TI? [Página web]. https://mx.indeed.com/cmp/Consultoria-Ti/salaries

12.- Liverpool. (s. f.). Laptop gamer HP Victus 15-FA2701WM 15.6 pulgadas Full HD Intel Core i5 NVIDIA GeForce RTX 4050 16 GB RAM 512 GB SSD [Página de producto]. https://www.liverpool.com.mx/tienda/pdp/laptop-gamer-hp-victus-15-

fa2701wm-15-6-pulgadas-full-hd-intel-core-i5-nvidia-geforce-rtx-4050-16-gb-ram-512-gb-ssd/1179817655?gclsrc=aw.ds&…

13.- Grupo Industronic. (2025, 9 de junio). Costo de electricidad en México: tarifas y cómo reducir la … [Página web]. https://grupoindustronic.com/costo-de-electricidad/

14.- Telmex. (s. f.). Paquetes de Internet para hogar [Página web]. https://telmex.com/web/hogar/mkt/paquetes-de-internet?utm_source=performance_max&utm_medium=cpa&utm_campaign=performance&utm_id=ads001&utm_term=80megas_netflix_sincosto_cdmx&…

15.- Google. (s. f.). AI y herramientas de productividad – Google One AI [Página web]. https://one.google.com/ai?utm_source=g1&utm_medium=paid_media&utm_campaign=sem_aiplans&…

16.- El Economista. (2024, 20 de julio). ¿Cuánto cuesta rentar una habitación con servicios incluidos en la CDMX? EconoHábitat. Recuperado de https://www.eleconomista.com.mx/econohabitat/Cuanto-cuesta-rentar-una-habitacion-con-servicios-incluidos-en-la-CDMX-20240719-0078.html

Final Project Results