NATIONAL AUTONOMOUS UNIVERSITY OF MEXICO
FACULTY OF ENGINEERING
ELECTRICAL ENGINEERING DIVISION
COMPUTER ENGINEERING
COMPUTER GRAPHICS AND HUMAN-COMPUTER INTERACTION

Final Project 2
User manual
Computer Graphics and Human-Computer Interaction

Student. 319079485

Professor. ENG. CARLOS ALDAIR ROMAN BALBUENA

Group 5

Semester 2026-1

Due Date: 25 de November 2025

Index

Introduction

This project consists of the development of an interactive 3D simulation built with C++ and OpenGL 3.3, which recreates a walking steampunk house equipped with a hierarchical animation system, proximity-interactive objects, an autonomous drone, and a cinematic tour reproducible via keyframes.

The main objective is to demonstrate the learning of computer graphics techniques: 3D modeling, texturing, lighting, interpolation animation, hierarchical transformations, synthetic camera, and GPU resource management.

During execution, the user can:

Freely explore the post-apocalyptic environment.

Activate the house's animations using keys.

Interact with objects that react to proximity.

Execute a pre-recorded automatic tour.

The scene includes a desert environment skybox, a drone with autonomous circular flight, doors that open based on distance, a rotating satellite dish, and four articulated mechanical legs with movements.

The project's implementation involves three-dimensional computer graphics concepts.

The rendering pipeline begins with the definition of three-dimensional geometry using vertices, normals, and texture coordinates stored in GPU buffers.

Programmable shaders process this information: the vertex shader transforms vertex coordinates from local space to screen space using model, view, and projection matrices, while the fragment shader calculates the final color of each pixel by applying Phong lighting models and texture mapping.

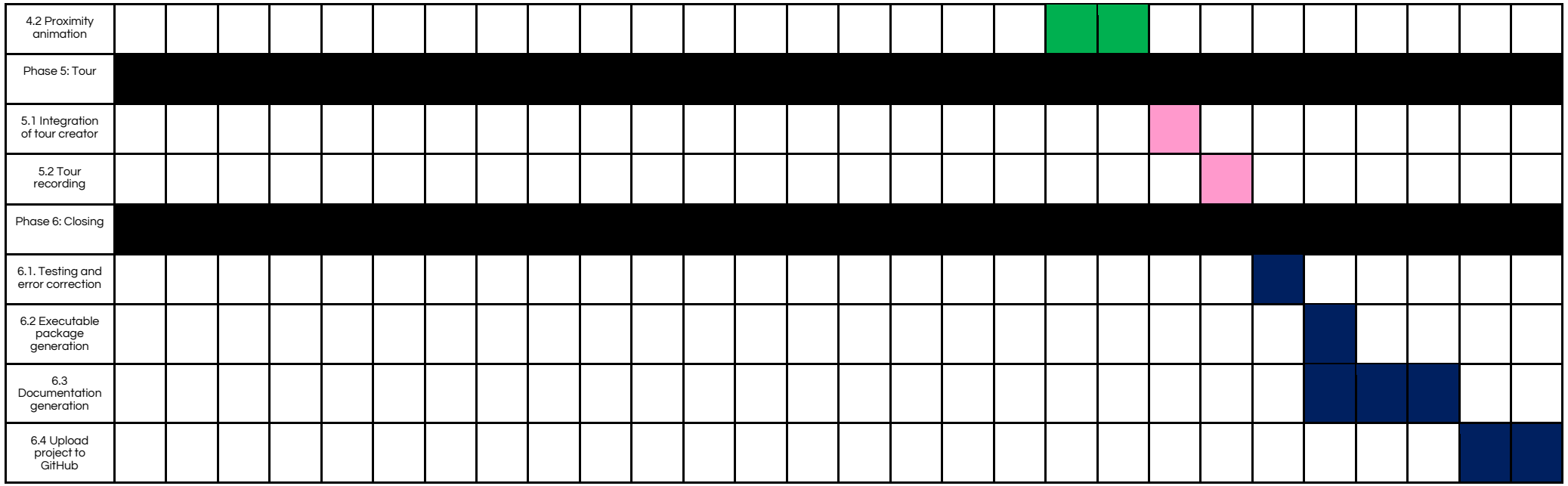
The lighting system implements three types of light sources: directional light simulating the sun with parallel rays, point lights that emit radially from specific positions with quadratic attenuation, and a spotlight-type flashlight with a defined opening angle.

Each light contributes ambient, diffuse, and specular components to the final color of the surfaces according to their material and orientation relative to the source.

Objetivos

The main objective of this project is to develop a real-time interactive 3D scene using C++ and the OpenGL graphics API.

1.- Gantt Chart	10/10	15/10	17/10	20/10	21/10	22/10	23/10	24/10	25/10	26/10	27/10	28/10	29/10	30/10	31/10	01/11	02/11	03/11	05/11	06/11	07/11	08/11	09/11	10/11	11/11	12/11	13/11	14/11
Phase 1: Planning and Setup																												
1.1 Choice of reference and objects																												
1.2. Environment Configuration (VS, Git)																												
1.3. Library Setup (GLFW, GLEW, GLM)																												
Phase 2: Basic Environment																												
2.1. Shader Loading (Classes)																												
2.2 House facade modeling																												
2.3 House interior modeling																												
2.4 House objects modeling																												
2.5 Texture Implementation																												
Phase 3: Scene Creation																												
3.1 Synthetic camera implementation																												
3.2 Model and texture loading																												
3.3 Basic Lighting																												
3.4 Skybox implementation																												
Phase 4: Animation																												
4.1 Hierarchical animation																												


4.2 Proximity animation
Phase 5: Tour
5.1 Integration of tour creator
5.2 Tour recording
Phase 6: Closing
6.1. Testing and error correction
6.2 Executable package generation
6.3 Documentation generation
6.4 Upload project to GitHub

2.- User Manual

2.1 Project Objectives

Welcome to the "Final Project" project. This is an interactive real-time 3D environment that simulates a walking robot-house, with a "steampunk" inspired style in a post-apocalyptic landscape.

The objective of this simulation is to allow free exploration of the walking house's environment.

You will be able to move through the scene, interact with objects that react to your presence, and activate an automated cinematic tour to see the scene in action, all rendered in real-time.

2.2. Technologies Used

This project was built using a set of tools for computer graphics development:

- 3D Modeling Software: Blender. Used to create, edit, assemble, and export all 3D models (.obj) used in the scene (the house, the drone, the legs, etc.).
- Programming Language: C++. It is the main language on which all the program's logic runs.
- Graphics API: OpenGL (Open Graphics Library). It is the interface that allows the program to communicate with the graphics card (GPU) to draw all the 3D graphics.

External Libraries

- GLFW Library: Used to create the application window, receive keyboard (W, A, S, D, P, L, etc.) and mouse input.
- GLEW Library: Allows the code to access modern OpenGL functions.
- GLM Library (OpenGL Mathematics): Used for all vector and matrix mathematical operations (translating, rotating, and scaling 3D objects).

- Assimp Library (Open Asset Import Library): Used to load 3D models (the .obj files for the house, drone, legs, etc.) and translate them into a format that OpenGL understands.

- stb_image.h and SOIL2 Libraries: Used to load image files (like .jpg or .png) that are used as textures for the models and the skybox.

2.3 System Requirements

This project is a viewer/demo developed in C/C++ with OpenGL (GLFW + GLAD + GLM) and is executable on Windows.

Below are the minimum and recommended requirements for both running and compiling the code.

Minimum (to open the scene and tour at basic quality)

- Operating system: Windows 10 (x86)
- CPU: Intel/AMD 64-bit with SSE2 support
- RAM: 4 GB
- GPU: OpenGL 3.3 compatible (e.g.: Intel HD 4000, NVIDIA GeForce 600 series, AMD GCN 1st gen) or integrated CPU graphics
- VRAM: 1 GB
- Storage: 500 MB free (binary + assets)
- Display: 1280x720
- Input: Keyboard and mouse
- Drivers: Updated graphics driver from the manufacturer (Intel/NVIDIA/AMD)

Recommended (for smoother lighting and animation)

- CPU: 4-core (Intel i5/Ryzen 5 or higher)
- RAM: 8 GB
- GPU: OpenGL 4.1+ (e.g., NVIDIA GTX 1050/1650, AMD RX 560/570, Intel UHD 620+) dedicated video card
- VRAM: 2 GB
- Storage: 1 GB free
- Display: 1920x1080 (or higher)

2.4. Interaction Guide and Controls

To run the program, download the compressed .rar file titled "319079485", unzip the file, then open the folder and run "Proyecto Final Computación Grafica.exe". Wait for it to load, and the scene will finally be displayed.

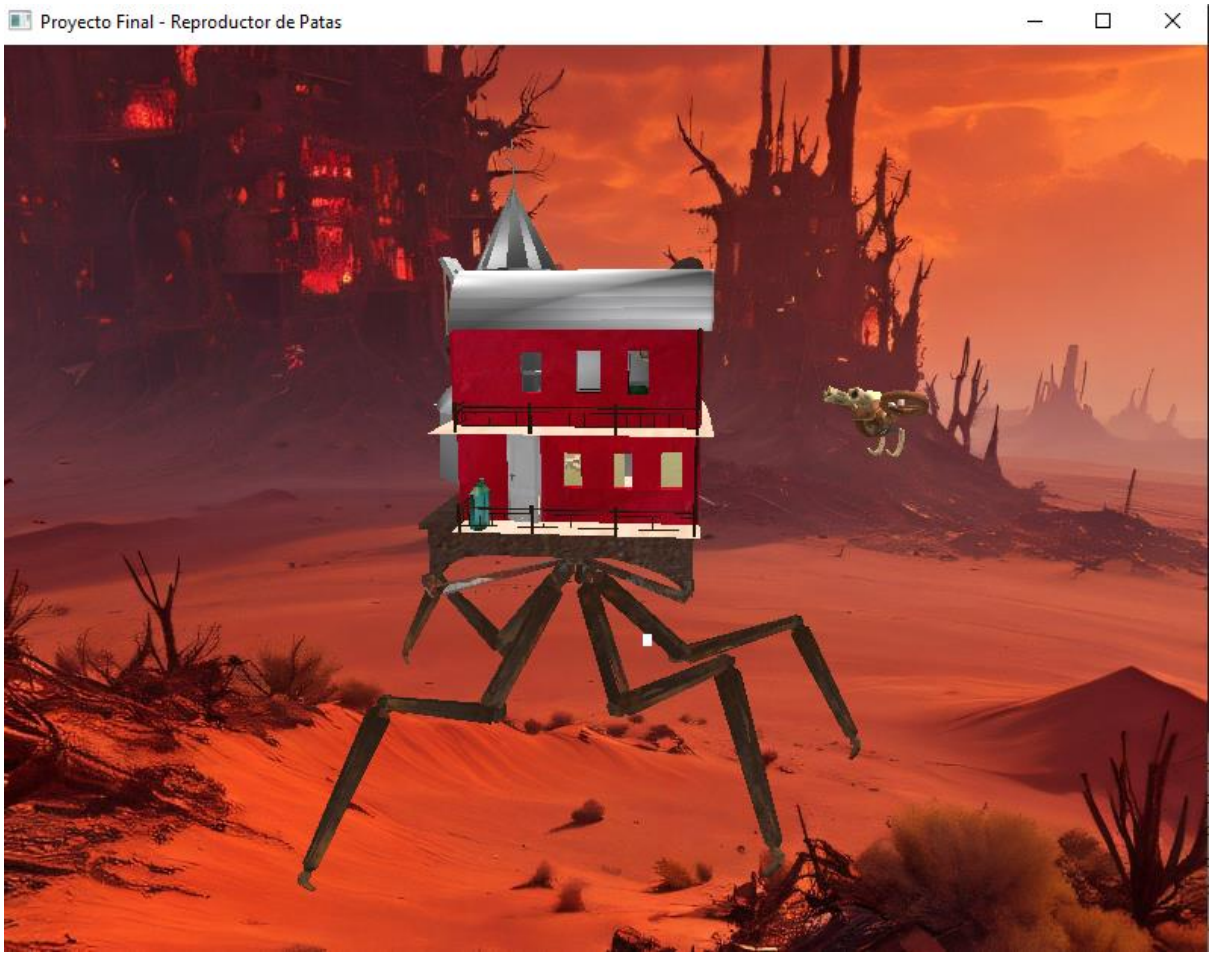
The program has two modes:

 Free Mode (manual control)

 Tour Mode (automatic playback)

The program will start in free mode (manual control); you can navigate freely through the scene.

Program home screen:



You can start the tour mode by pressing the "L" Key.

2.2.1 Free Mode Controls

In this mode, you have full control over the camera.

Action	Key
Move camera	
Move Forward	W or Up Arrow
Move Backward	S or Down Arrow
Move Left	A or Left Arrow
Move Right	D or Right Arrow
Rotate camera	Mouse
Animations	
Start automatic tour	L
Animated central light	Space
Metallic legs animation	P
Hide second-floor wall	F
Exit program	
Exit	ESC

- Activate Leg Animation: Press the P key. The four legs of the house will begin to move in their animation cycle. Press P again to stop them.

- Hide Wall: Press the F key. The front wall of the house's second floor will disappear, allowing you to see inside. Press F again to make it reappear.

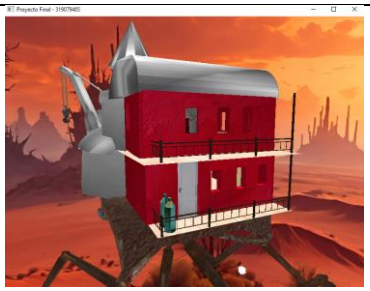
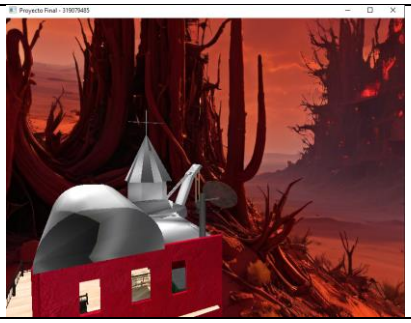
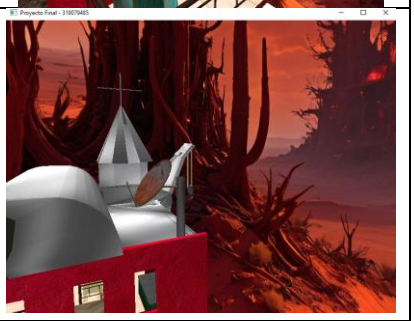
2.2.2 Tour Mode Controls

When you press the corresponding key (L), the camera will take automatic control and begin a pre-recorded tour of the scene (loaded from camera_anim.txt).

At the same time, the animation of the four legs will activate (just like pressing P). During this mode, the W, A, S, D, and mouse controls will be disabled. Press L at any time to stop the camera tour, returning manual control.

2.2.3 Automatic Interactions (Free Mode)

While exploring in free mode, the scene will react to the camera's presence:

Object	Reaction	***No reaction, camera far***	***With reaction, camera near***
Main Door	If the camera gets close to the house's main door, it will open smoothly. When you move away, it will close.		
Dish	If you fly with the camera to the roof and get close to the rusty antenna (dish), it will activate and start spinning.		

Repository GitHub:
https://github.com/soeil1/319079485_ProyectoFinalTeoria_GPO05.git

To visualize the complete operation of the house and all its implemented features, a demonstration video was prepared that shows navigation in the environment, the activation of the animations for each part

Demostrative video:
https://youtu.be/0YKs-a2w2YA