317005435
Computer Graphics Lab Group 6

Final Project: 3D Recreation of CJ's House, from the game Grand Theft Auto: San Andreas

Index

User Manual

I. Introduction 3

II. Download, compile and run 3

III. Interactions 6

Technical Manual

I. Project description 9

II. Objectives and considerations 10

III. Scope 11

IV. Limitations 12

V. Gantt Chart 12

VI. Requirements 13

VII. System design **14**

VIII. Functional Specifications **14**

IX. Cost analysis **16**

X. Results 18

XI. Definitions and abbreviations 20

XII. References 21

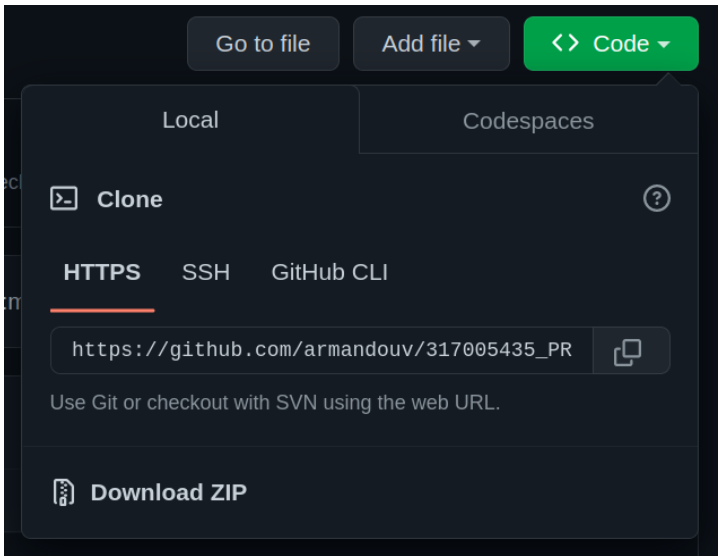
User manual

I. Introduction

This user manual presents a brief tutorial on downloading, compiling and executing the project, as well as the necessary instructions to interact with the graphical environment of CJ's house, from the video game Grand Theft Auto: San Andreas. This includes how to manipulate the camera in the environment, as well as the corresponding keys to carry out the implemented interactions.

II. Download, compile and run

To download the project, you need to go to the repository web page (https://github.com/armandouv/317005435_PROYECTOFINAL2023-1_GPO6). Subsequently, you must click on the Code button, and then on the download ZIP option.



Once downloaded, we unzip the file, and we see that we have all the files present in the repository.

At this point, we have two options to run the project:

1. Compile and run the project

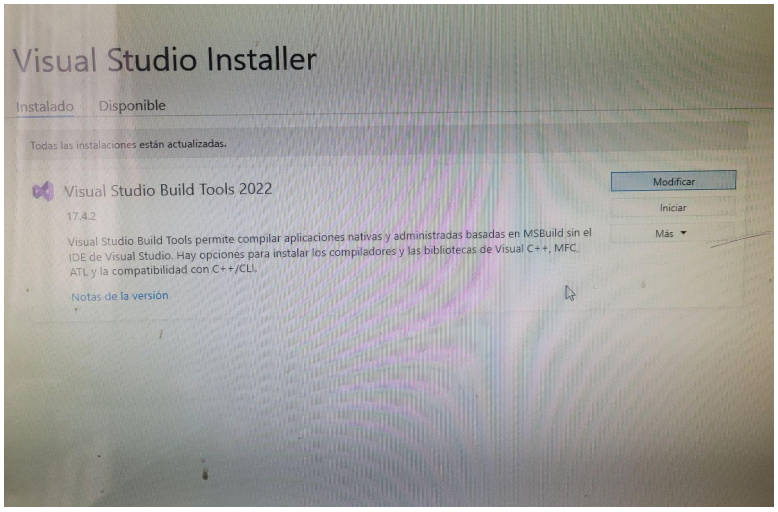
If you want to compile the project, you need to have installed:

- The Visual Studio C++ development tools, which can be installed as follows:
    - First, you have to install Git, as well as the installer of the development tools, using the following commands:

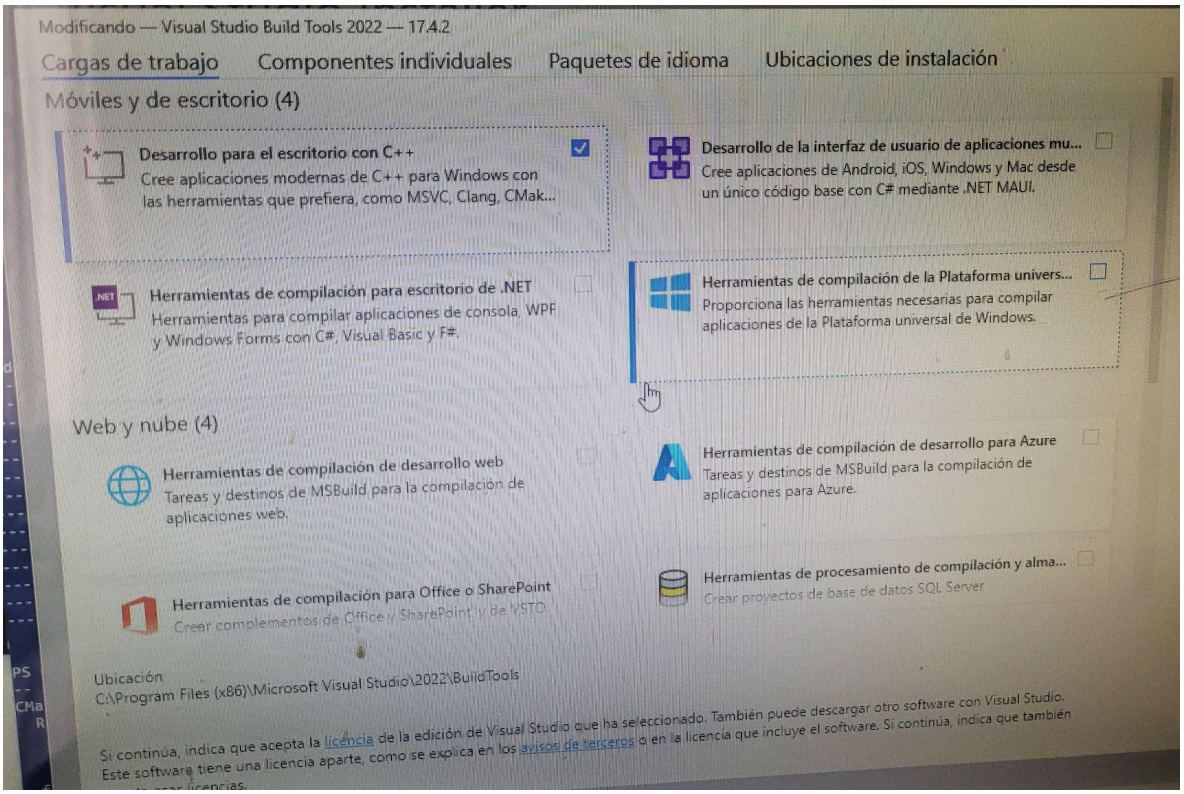
winget install -e --id Git.Git

winget install Microsoft.VisualStudio.2022.BuildTools

- Afterwards, you need to run the installer for Visual Studio (or Visual Studio Installer), and select the Modify option.



- Finally, we select the option Desktop development with C++, and click on Modify. We wait for the installation to finish.



- The CMake tool, which, if not installed in the previous step, can be installed using the Powershell command:

winget install -e --id Kitware.CMake

Once these tools are installed, the following commands should be run from the **development Powershell for Visual Studio**, just installed. From the root directory of the project, generated using the following commands. It is important to mention that this process can take a few minutes:

cd src

cmake .

cmake --build . --config Release -j4

Once this is done, we can run the program as follows:

./proyecto_final.exe

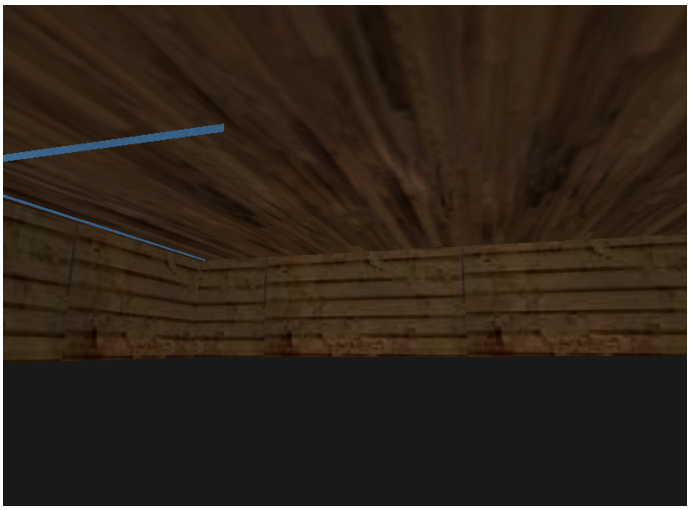
2. Using the provided executable

In the root directory of the project, there is a file named build.zip with the project executable. To do so, it is necessary first to unzip said file to a new build directory, and then execute the following commands:

cd build

./proyecto_final.exe

Once the program is executed, a window will be displayed with the environment ready for use:



III. Interactions

You can interact with the virtual environment as follows:

Camera and general controls

Move forward - *Arrow key up or W.*

Move back - *Arrow key down or S.*

Move left - *Left arrow key or A.*

Move right - *Right arrow key or D.*

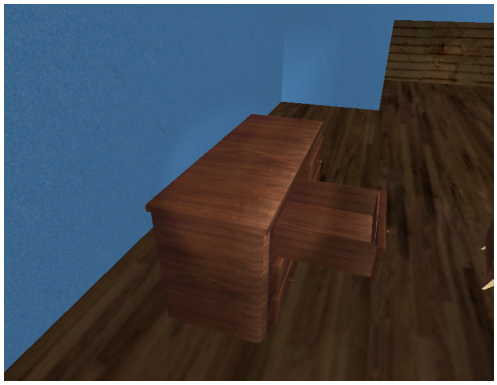
Rotation - *Mouse*

Exit - *Esc key*

Environment

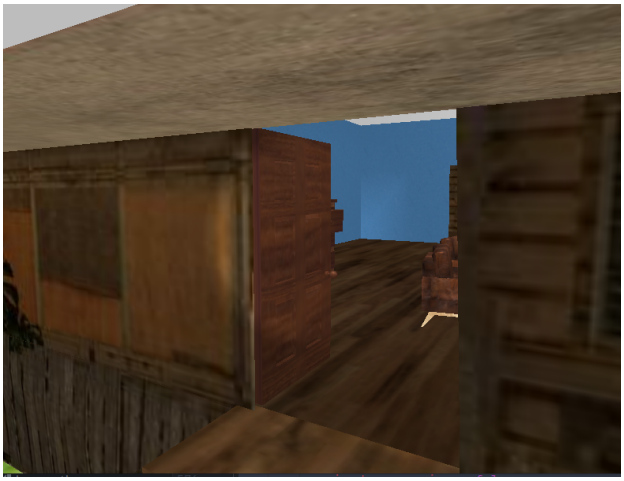
Turn ceiling lamps on and off - *Space key.*

Open and close drawer - *M key.*



Move chair down and out of the table - *N key.*

Open and close door - *B Key.*



Technical manual

I. Project description

A façade and a space must be selected for its 3D recreation in OpenGL, which can be real or fictitious. In addition, reference images of said spaces must be presented for comparison and establishment. In the reference image, 7 objects must be visualized that the student will recreate virtually. These objects must be as similar to the reference image, as well as the setting included. In addition, a minimum of 5 animations must be made: 3 simple and 2 complex.

The project must be delivered individually, with a user manual that explains each interaction within the recreated virtual environment and a technical manual that contains the project documentation that includes objectives, Gantt chart, project scope, limitations and documentation. Of code. On the other hand, the link of the project must be shared in a repository on GitHub.

The space chosen to recreate is CJ's house, from the video game Grand Theft Auto: San Andreas. The proposed reference images are:

Façade

Room

The elements to be represented, in addition to the setting, are:

- Living room table and back table
- Small armchair (x2)
- Large armchair
- Dressing
- Chair (x2)
- Frets on top of the rear table (x2).
- Door
- Fly
- Ceiling lamp
- Pot with
- Spider

In the initial planning, only 7 objects were considered, but they were increased to improve the completeness of the project.

The proposed animations are:

- Opening and closing a door (simple)
- Opening and closing a dresser drawer (simple)
- Moving a chair under and out of a table (simple)
- Spider hanging from a web (complex)
- Fly flying (complex)

II. Objectives and considerations

- The student must apply and demonstrate the knowledge acquired throughout the course, which includes:
    - Geometric and hierarchical modeling.
    - Knowledge of a specific purpose modeling software such as Maya.
    - textured.

    - Basic transformations.
    - Simple and complex animation.
    - Loading and adaptation of models
    - Lighting and shading
- The recreated virtual environment must be as similar as possible to the presented reference image.
- A minimum of 5 animations must be created, having 3 simple and 2 complex. In addition, each animation must have context and coherence.

III. Scope

The project aims to be a recreation of the virtual environment already presented, which is easy to use and install on any computer (see the user manual), for which it includes:

- Functional virtual environment of the façade and selected rooms.
- Setting in the selected room and façade.
- 12 objects modeled.
- Interactions with the environment.
- Attractive and coherent animations.
- Intuitive camera handling.
- Fluency in the execution of the program.

On the other hand, the project does not include:

- Detailed recreation of the environment outside the façade.
- Complex object models (not considered low poly).
- Using a Skybox.
- Animations using advanced techniques.

IV. Limitations

- The recreated environment was made using an approximate scale, since there was no detailed plan with the appropriate sizes, so it is likely that it is not perfect, which influences the distribution and size of the objects.
- It is possible for the user to walk through parts of the environment where no object of interest is found.
- It was not possible to use very complex models, and it was decided to use those with low polygons to avoid performance problems.
- Although the animations work correctly, it is possible to optimize some of them to improve their performance, since they can be inefficient when scaling the project.
- If the application is deployed, the end user can access the Shaders used and not used, as well as the models; so there is no control over which files an end user can and cannot see.

V. Gantt Diagram

Diagrama de Gantt

Tarea	Duración											
	Septiembre			Octubre	Noviembre							
	14	20 - 26	27	26	1	8	13	14	20	23	27	29
Presentación de proyecto y requerimientos												
Selección de fachada y habitación												
Entrega de imagen de referencia												
Creación de modelo de mesa												
Creación de modelo de sillón grande												
Creación de repositorio de Github												
Agregar instrucciones de compilación												
Automatizar dependencias												
Arregla dependencia de GLEW												
Creación de modelo de tocador												
Arregla luz ambiental												
Creación de modelo de silla												
Creación de modelo de maceta												
Creación de modelo de trastes												
Creación de fachada												
Animación de cajón												
Animación de silla												

Tarea	Diciembre					
	1	2	4	5	6	7
Presentación de proyecto y requerimientos						
Selección de fachada y habitación						
Entrega de imagen de referencia						
Creación de modelo de mesa						
Creación de modelo de sillón grande						
Creación de repositorio de Github						
Agregar instrucciones de compilación						
Automatizar dependencias						
Arregla dependencia de GLEW						
Creación de modelo de tocador						
Arregla luz ambiental						
Creación de modelo de silla						
Creación de modelo de maceta						
Creación de modelo de trastes						
Creación de fachada						
Animación de cajón						
Animación de silla						
Creación de modelo de mosca						
Creación de modelo de araña						
Animación de mosca						
Animación de araña						
Creación de modelo de puerta						
Animación de puerta						
Creación de modelo de lámpara de techo						
Agrega point lights						
Crea ejecutable						
Elimina movimiento de point lights						
Elaboración de documentación						
Entrega de proyecto						

VI. Requirements

The formal requirements of the project are the following:

- Have a reference image of the façade and room, and specify the 7 objects to be modeled.
- Create a 3D OpenGL recreation of a space that can be real or fictitious. The house of CJ, from the video game Grand Theft Auto: San Andreas, was chosen.
- A synthetic camera must be included.
- A user manual and a technical manual must be produced, in English and Spanish.
- At least 3 simple and 2 complex animations must be implemented, which must have context and be coherent.
- The objects and setting should be as similar as possible to the reference image.
- The project must be placed in a repository on Github.

- An executable file must be delivered.
- A cost analysis should be done.

VII. System design

From the src directory of the project, the different modules can be differentiated, both in the different source code files and in the corresponding subdirectories. The main modules that make up the system are the following:

- **Module Models:** It contains the 3D models used in the project, and they are located in the Models directory. It is worth mentioning that each model is in obj format, with its respective texture and mtl files. Subsequently, in the main module the models will be loaded for each one when required.
- **Shaders module:** In it are the Shaders that are used in the project, as well as some extras for future expansion of the project. not all of them are being implemented, however they are decided to be retained for future reference. The main shaderused in the project is the lightingShader, which calculates the appropriate lighting for the recreation of the space.
- **Auxiliary files module:** Here all the files with .h extension present in the project are considered, whose objectives are to implement some necessary auxiliary functionalities, such as the implementation of camera management, model loading, shaders, textures, etc.
- **Main module (main):** In the main module, all the main logic of the project is implemented, which includes the configuration of OpenGL for rendering, loading and initialization of the necessary libraries, creation of windows, loading of shaders, models and textures. Additionally, animations are implemented, basic transformations are performed, environment lighting is configured, imported models are drawn using the lightingShader, and interactions with the environment are implemented by specifying the appropriate keys or mouse input to define the behavior.

VIII. Functional specifications

As mentioned, the main module implements all the main logic of the project, which includes:

- OpenGL configuration for rendering
- Loading and initializing the necessary libraries
- Creation of windows

- Loading of shaders, models and textures
- Basic transformations for adaptation of models
- Implementation of lighting the environment
- Drawing imported models using the lightingShader
- Implementation of animations
- Implementation of interactions with the environment by specifying the appropriate keys or mouse input to define the behavior.

The following is a breakdown of each of the mentioned functions, within specific functions in the code:

Function Name	Specification
main()	This is the main function within the code, as the name implies. First, it starts configuring the window to use, with the help of the utilities provided by the GLFW library, and the GLEW library is initialized. Subsequently, the Shaders to be used are created and initialized, as well as the models, with the help of the auxiliary files module. The VAO and VBO are configured with the help of OpenGL, and most of the previously presented tasks are executed in the rendering loop. These include implementing the logic needed for animations, lighting setup including directional light and point lights, and adapting and drawing models using the lightingShader and basic transformations.
DoMovement()	In this function you deal with the movement of the camera by pressing the appropriate keys, using the respective Camera module.
KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mode)	Several aspects of the interaction with the environment are implemented in this function: closing the window and terminating the program, pressing the ESC key, turning on and turning off the ceiling lamps, using the space key, and activating the animations of the drawer, chair and door, with the m, n and b keys, respectively.

MouseCallback(GLFWwindow* window, double xPos, double yPos)	This function deals with rotating the camera by pressing the appropriate keys, using the respective Camera module.

IX. Cost analysis

To determine the final cost of the project, all the activities carried out are taken into account, and an objective analysis is carried out, taking into account both the value of the product generated and the time required to carry it out. A labor cost per hour of 90 Mexican pesos is considered. To determine the final price of each item, the cost of work per hour is considered and an analysis of market prices is carried out if necessary (for example, to carry out the models, prices are compared with products from the Turbosquid platform). :

- *Proposal for a reference image of the façade and room, and 7 objects to be modeled.*
  Estimated time: 4 hours.
  Estimated cost: 360 Mexican pesos.
- *Implementation and adaptation of models (work time is taken into account, as well as market prices for each model)*
  - *Adaptation*
    Estimated time: 6 hours
    Estimated cost: 540 Mexican pesos.
  - *Dining room table and back table*
    Estimated time: 1 hour
    Estimated cost: 100 Mexican pesos.
  - *Small armchair (x2)*
    Estimated time: 1.5 hours
    Estimated cost: 180 Mexican pesos.
  - *Large armchair*
    Estimated time: 2 hours
    Estimated cost: 200 Mexican pesos.
  - *Dressing*
    table Estimated time: 2 hours
    Estimated cost: 230 Mexican pesos.

- *Chair (x2)*
  Estimated time: 1.5 hours
  Estimated cost: 150 Mexican pesos.
- *Frets on the back table (x2).*
  Estimated time: 1 hour
  Estimated cost: 100 Mexican pesos.
- *Door*
  Estimated time: 1 hour
  Estimated cost: 110 Mexican pesos.
- *Fly*
  Estimated time: 2 hours
  Estimated cost: 300 Mexican pesos.
- *Ceiling lamp*
  Estimated time: 1.5 hours
  Estimated cost: 160 Mexican pesos.
- *Pot with plant*
  Estimated time: 3 hours
  Estimated cost: 400 Mexican pesos.
- *Spider*
  Estimated time: 3 hours
  Estimated cost: 450 Mexican pesos.

Estimated cost: 2920 Mexican pesos

- *Implementation of animations*
  Estimated time: 5 hours.
  Estimated cost: 450 Mexican pesos.
- *Preparation of technical and user manual*
  Estimated time: 7 hours.
  Estimated cost: 630 Mexican pesos.

Estimated total cost of the project: 4360 Mexican pesos

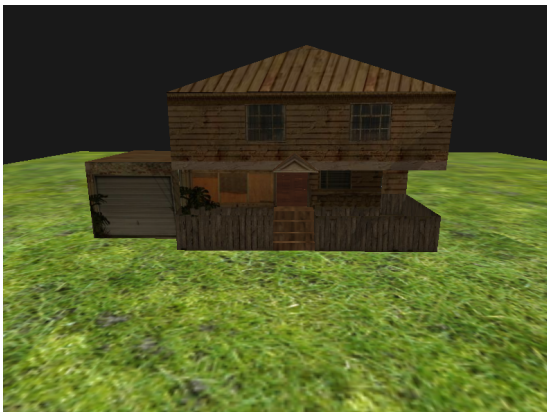
X. Results

Below is a brief comparison of the proposed reference images and the final environment obtained, as a sample of the project result:

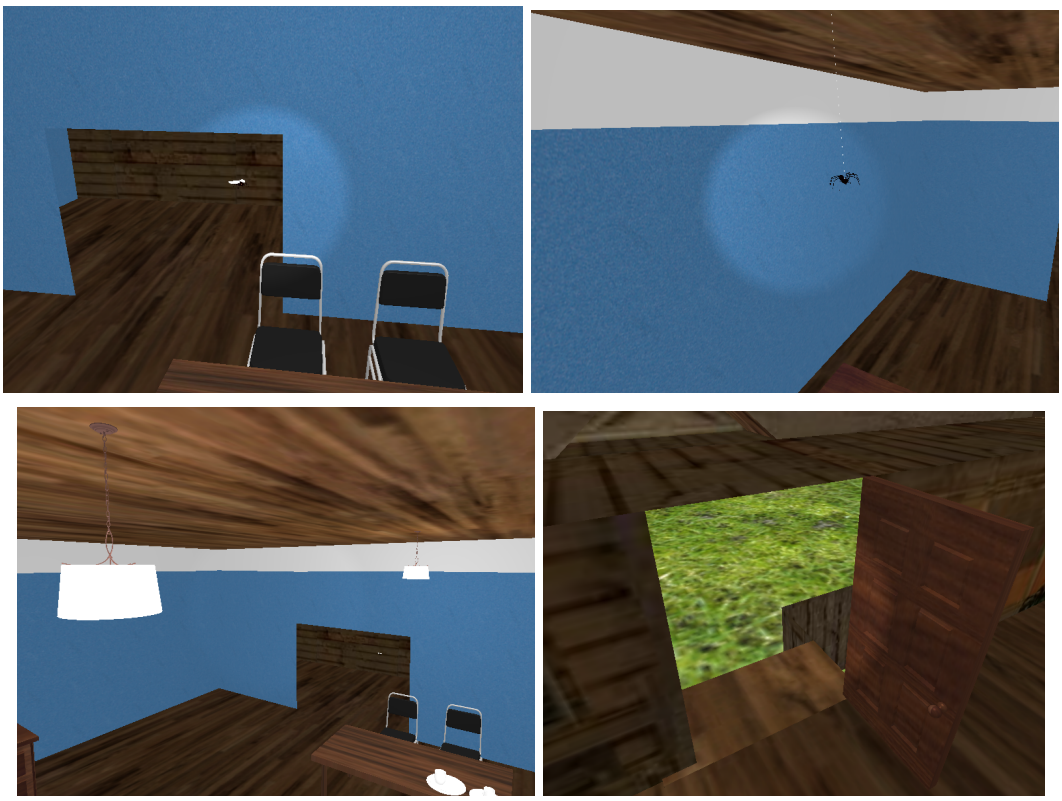
Facade

Room

Recreation





XI. Definitions and abbreviations

Skybox	It is a graphic element that allows to give, in a three-dimensional space, the illusion that this space is larger than it really is, by providing a visual effect of a stage by means of a texture applied to an object inside which the environment camera is located.
Texture	It is an image mapped in UV space that is associated with a three-dimensional mesh to give it greater realism.
.obj file	is a geometry definition format. Represents only the 3D geometry, that is, the position of each vertex, the UV position of each texture coordinate vertex, the vertex normals, and the faces that cause each polygon to be defined as a list of vertices.
.mtl	file An mtl file is an auxiliary file for obj files, containing definitions of materials that could be used by the obj file. Each material definition starts with a newmtl statement, which defines the name of the material, followed by lines specifying particular properties.
Shader	It is a program that calculates the appropriate levels of light, dark and color during the rendering of a 3D scene. Most run on the GPU, but it's not a strict requirement.
OpenGL	OpenGL is a library that provides services for rendering 2D and 3D graphics.
GLFW	*OpenGL FrameWork, or Graphics Library FrameWork*, is a lightweight utility library for use with OpenGL. Its acronym stands for Graphics Library Framework. It gives

	programmers the ability to create and manage OpenGL windows and contexts, as well as handle joystick, keyboard, and mouse input.
GLEW	It is an open source library for C/C++, whose functionality is the loading of OpenGL extensions. It provides efficient runtime mechanisms to determine which OpenGL extensions are supported by the target platform. OpenGL core and extension functionality are exposed in a single header file.
GLSL	*OpenGL Shading Language*, is the main shading language for OpenGL. It is based on C.
GLM	*OpenGL Mathematics*, is a mathematical library written in C++ for the development of OpenGL-based graphics software. It allows you to easily manipulate vectors and matrices and perform operations on them.

XII. References

Nash Vail. (2017). Understanding Linear Interpolation in UI Animation. August 28, 2022, from FreeCodeCamp. Website: https://www.freecodecamp.org/news/understanding-linear-interpolation-in-ui-animations-74701eb9957c/

Foley, J. D, Van Dam, A. (1982). Fundamentals of Interactive Computer Graphics. August 28, 2022, from Addison-Wesley. Website: https://archive.org/details/fundamentalsofin00fol

Unknown. (2014). Uniform Variables. August 28, 2022, from Cornell University. Website: https://www.cs.cornell.edu/courses/cs4620/2014fa/lectures/glsl2.pdf

Various authors. (2022). Frequently Asked Questions. August 28, 2022, from GLFW Website: https://www.glfw.org/faq.html

Various authors. (2017). The OpenGL Extension Wrangler Library. August 28, 2022, from GLEW Website: http://glew.sourceforge.net/

Orlando Saldívar Esquivel. (2021). Project Management subject template. Consulted on December 2, 2022.

Various authors. (2018). MTL Files, Material Definitions for OBJ Files. Retrieved September 26, 2022, from Florida State University. Website: https://people.sc.fsu.edu/~jburkardt/data/mtl/mtl.html

Mike Bailey. (2022). OBJ Files. Retrieved September 26, 2022, from Oregon State University. Website: https://web.engr.oregonstate.edu/~mjb/cs550/PDFs/ObjFiles.4pp.pdf

Andy Johnson. (2022). Texture and Other Mapping. Retrieved September 26, 2022, from the University of Illinois Chicago. Website: https://www.cs.uic.edu/~jbell/CourseNotes/ComputerGraphics/TextureMapping.html