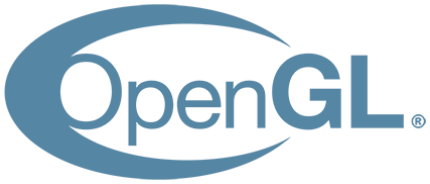
Universidad Nacional Autónoma de México

Faculty of Engineering

Final Project

Graphic Computing Laboratory



Name: Herrera Godina Diana Celeste

316161927

Group: 12

Professor: Ing.Carlos Aldair Roman Balbuena

Index

Technical Manual:

Final Project

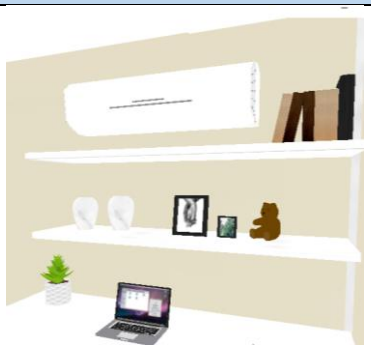
Objetive

The student must apply and demonstrate the knowledge acquired throughout course.

facade and room to recreate

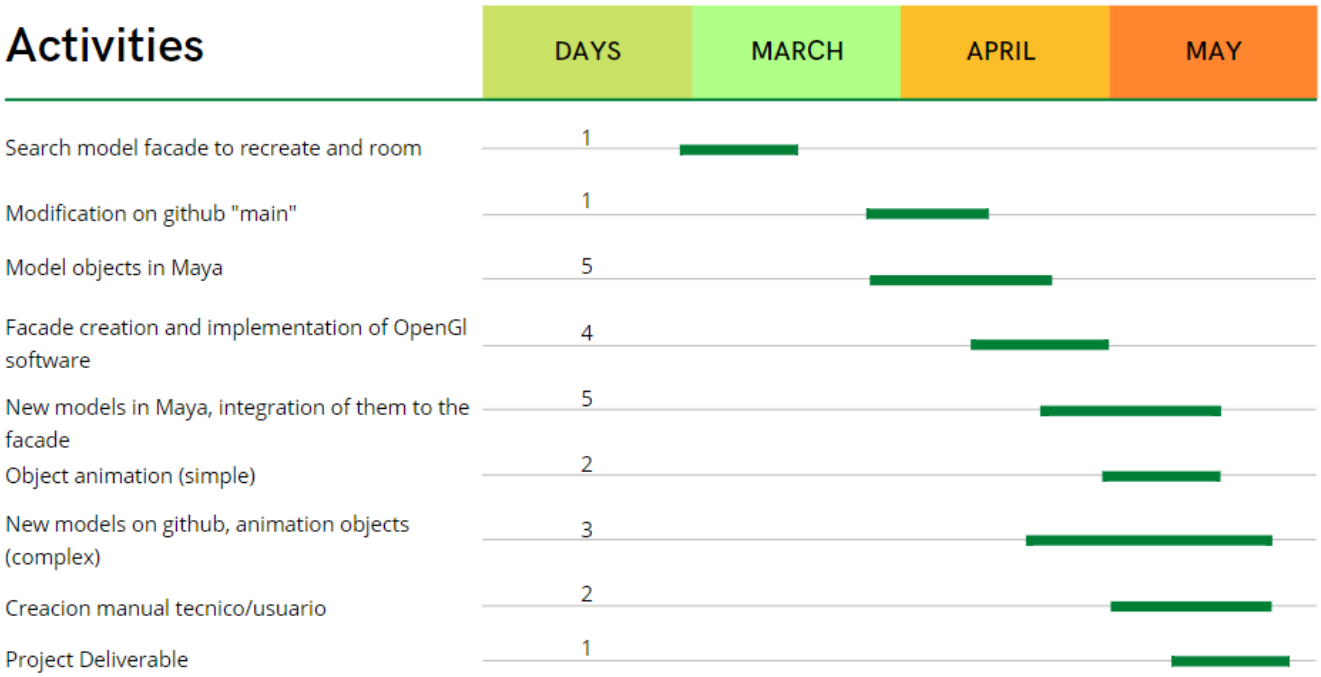
Facade and room in 3D (OpenGL Software)

Objects to recreate inside the room:

Objects	Reference image	Object in OpenGL 3D
Shelves		
Bed		
Desk		
Chair desk		

Vase		
Laptop		
Plant		

Gantt Chart



Scope of the Project

The project seeks to model and recreate objects as realistic as possible (modern house) based on a facade found on the internet, a modern room and in this way apply our skills acquired in the computer graphics laboratory course, adapting the elements and taking it to the GitHub version platform, create animations within this environment, the activities that were carried out in order as they were modeled, it was required to upload to GitHub in order to load the models and have the versions within the platform.

Implementation: This project can be implemented for the development of an application or simply if it is required to model a space that can later be built or something technological implemented and visualized in a better way.

Deliverables: It is required at the end to have an executable file to deliver this project, in addition to the user and technical manuals.

Limitations: the resources that were counted to carry it out were few due to the type of computer, this in the end complicated the execution of the program (OpenGL) a bit, already loading all the models within the main main, to carry out this project it took a few months to go changing what is required. The expected results seek to meet the objective, show the knowledge acquired throughout the semester.

Requirements: To do it, it is required to have a computer capable of handling the software, OpenGL, Maya, Gimp.

Other requirements: Have basic knowledge of the GitHub platform, thus create a repository to have version control and thus in the revision of the project can download and later approve our final project.

Code Documentation

Used libraries:

Libraries	Description
GLEW	C/C++ extension loader library determines which OpenGL extensions are supported by the platform.
GLFW	Provides a simple API to create windows, contexts and surfaces, receive inputs and events.
GLM	Mathematical implementation of C++.
SOIL2	Load textures for the models.
Assimp	Import 3D models, in our case Maya, obj models to later read the information and move it to OpenGL.

GL includes:

Includes	Description
Shader.h	It is used to retrieve the code that contains the vertex and fragment, assigning memory locations.
Camera.h	Synthetic camera is used to move within the environment.
Model.h	Import the models used "obj" loading the coordinates of the vertices of said objects to import them, load the textures.
stb_image.h	Load images as textures, loading the files to be integrated into the project.

Shaders

Shaders	Description
anim.frag y anim.vs	It helps us and provides information to animate within the shader.
lighting.vs y lighting.frag	Has control of coordinates as well as texture, modifies model arrays. The frag helps us with light based on its components (spectacular, diffuse, all about materials).
lamp.vs y lamp.frag	Frag accesses the colors, the vertex checks the "VAO" memory locations with the information of the vertices as well as the matrices, checks said information.

Functions:

Function	Description
DoMovement	If there are animations inside KellCallback they change state checking if the animation is active.
KeyCallback	It captures the information from the keyboard, processes this information for the events.
animacionR	Function for the animation of the mouse (toy) with parabolic shot. In addition to the animation of the shower (using sine of time).
MouseCallback	It captures the information of the mouse in order to rotate and move in the environment.

Load Models:

```
// Carga de modelos
  Model fachada((char*)"Models/Fachada/fachada2.obj");
  Model desk((char*)"Models/Desk/desk.obj");
  Model cuadro((char*)"Models/Cuadros/cuadros.obj");
  Model cama((char*)"Models/Bed/bed.obj");
  Model silla((char*)"Models/Silla/silla.obj");
  Model aire((char*)"Models/Aire/aire.obj");
  Model repisa((char*)"Models/Repisa/repisa.obj");
  Model planta((char*)"Models/Planta/planta.obj");
  Model lap((char*)"Models/Laptop/laptop.obj");
  Model patin((char*)"Models/Patineta/patineta.obj");
  Model regadera((char*)"Models/Regadera/regadera.obj");
  Model libro((char*)"Models/Libro/book.obj");
  Model raton((char*)"Models/Juguete/juguete.obj");
```

The path was specified for each specific model within the “Models” folder, indicating the name of the obj so that the model.h file can load them.

Simple animations:

animation desk chair

variable animation	Tipe	Function
acsilla	bool	Check if the function is activated
transilla	float	translate function, activates it
rotsilla	float	Function to rotate, activates it

Skate board animation

Variable animation	Tipe	Function
acpat	bool	Check if the function is activated
transpat	float	translate function, activates it
rotpat	float	Function to rotate, activates it

Book animation

Variable animation	Tipe	Function
transbookX	float	Translate function for X-axis
transbookY	float	Translate function for Y-axis
transbookZ	float	Translate function for Z-axis
rotbook	float	Function to rotate, activates it
acbook	bool	Check if the function is activated

In these simple animations they were only linear, using functions: rotation, translation and scaling where these parameters of the matrix and its position vector (x,y,z) are required.

Complex animations:

Watering can animation:

```
void animacionR()
{
    // Movimiento de la regadera usando la variable tiempo para oscilar
    if (circuitoReg1)
    {
        if (recorrido1)
        {
            movKitYReg += 0.5f;
            if (movKitYReg < 1.5)
            {
                recorrido1 = false;
                recorrido2 = true;
            }
        }
        if (recorrido2)
        {
            rotKitReg = -60;
            movKitZReg -= 0.5f;
            if (movKitZReg < -2.0)
            {
                recorrido2 = false;
                recorrido3 = true;
            }
        }
        if (recorrido3)
        {
            movKitZReg = sin(tiempo + 5);
            movKitXReg = sin(tiempo);
        }
    }
```

Watering

Variable animation	Tipe	Function
movkitXReg	float	Move on the X axis
movkitYReg	float	Move on the Y axis, it is activated so that the watering can does not reach the ground
movkitZReg	float	Move on the Z axis
rotKitReg	float	Function to rotate, activate it so that it has a degree of inclination and it seems that the shower is at an angle that reaches the plants.
tiempo	float	Variable for time, to apply the sine function and can move undulatingly.

circuitoReg1	Bool	tour function
Recorrido1	Bool	Variable for travel 1
Recorrido2	Bool	Variable for route 2
Recorrido3	Bool	Variable for tour 3; apply the sine function of time + the required rotation

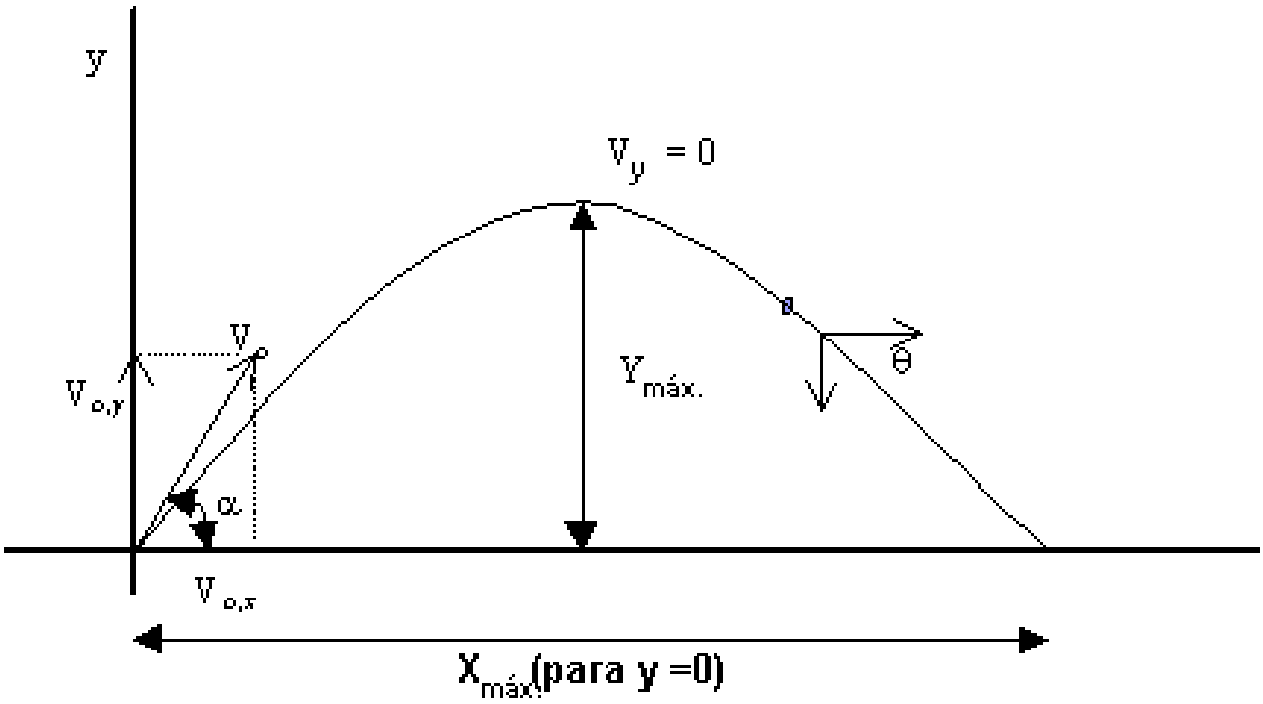
Toy mouse (car) animation:

```
  //little mouse movement using parabolic shot
    if (circuitoRat1)
    {
        if (recorrido4)
        {
            rotKitRat = 0;
            v0y += 0.5f;
            v0z += 0.5f;
            if (v0z > 1.5 || v0y > 1.5) //calculate the maximum height of the
{
                recorrido4 = false;
                recorrido5 = true;
            }
        }
        if (recorrido5)
        {
            v0y -= 0.5f;
            v0z += 0.5f;
            if (v0z > 3.5 || v0y < 0.0)   //height increases when it falls
            {
                recorrido5 = false;
                recorrido6 = true;
            }
        }
    }

    if (circuitoRat2)
    {
        if (recorrido6)
        {
            rotKitRat = 180;     //returning the mouse to its initial position
            v0z -= 0.5f;
            if (v0z < 0.5)
            {
                recorrido6 = false;
                recorrido4 = true;
            }
        }

}
```

For the animation of the parabolic shot (mouse)



Use one and max for this function so that the mouse does not go to the ground or to the ceiling, Voy and a VoX were required to be able to move, since when it falls it will change its speed, more strength is required to return to the ground and have the original position, in this case the mouse at the moment of falling will return with a path on the ground "imitation a toy car".

Toy mouse:

Variable animation	Tipe	Function
v0y	float	It moves in the Y axis in the parabolic shot, it is the maximum height in this case.
v0z	float	Move in the Z axis, in the parabolic shot
rotKitRat	float	Rotate so that the mouse returns to its initial position.

circuitoRat1	bool	In this route, both x and y are increased to make it advance upwards, making a curve and reaching Ymax.
circuitoRat2	bool	Return path function for the mouse.
Recorrido4	Bool	Identify and rotate the mouse to the maximum height to increase.
Recorrido5	Bool	Increase the speed at which the toy mouse falls so that it reaches the other side as it descends the curve past the maximum height it can move.
Recorrido6	Bool	Move the mouse 180° to return to a point so that it can return to its original position.

Project conclusions:

The objectives of the project were fulfilled since we can implement everything that we saw throughout the course in OpenGL and in some other modeling tools in addition to an image editor. This project also helps us to implement it in real life. that we can model any thing or engineering project, we were also able to make models and animate based on them, these animations required previous knowledge seen in the course. In addition, we were able to texture the models and upload them to Open gl in order to obtain the results of the facade and the room. Remembering all the knowledge we applied in the semester.

Enlace github:

https://github.com/DianaCelesteHerrera/316161927_Proyecto_Gpo12.git