National Autonomous University of Mexico

Faculty of Engineering

Final Project

Graphical computing

Name: Herrera Godina Diana Celeste

316161927

Group: 4

Professor: Eng.Carlos Aldair Roman Balbuena

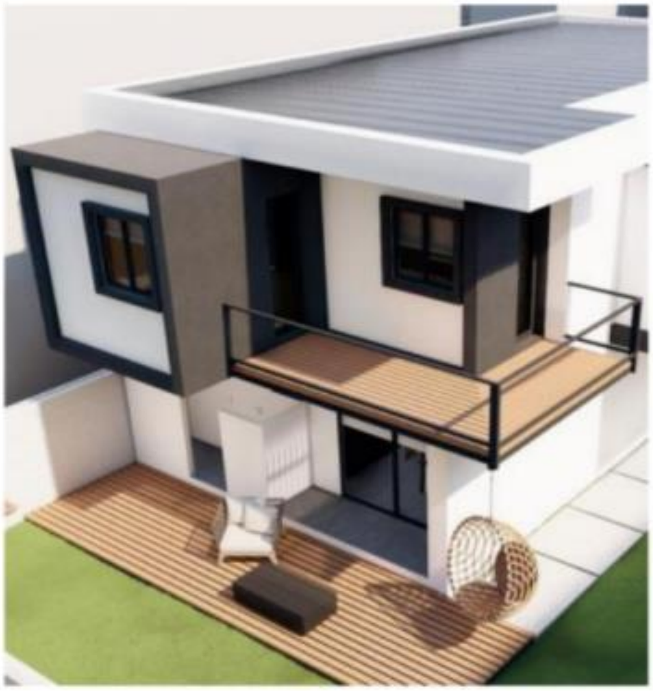
Index

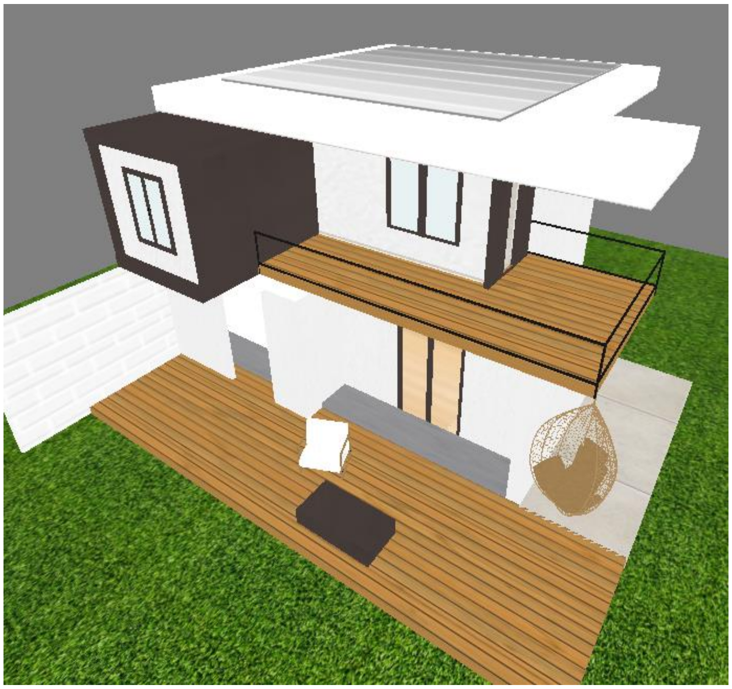
Final Project

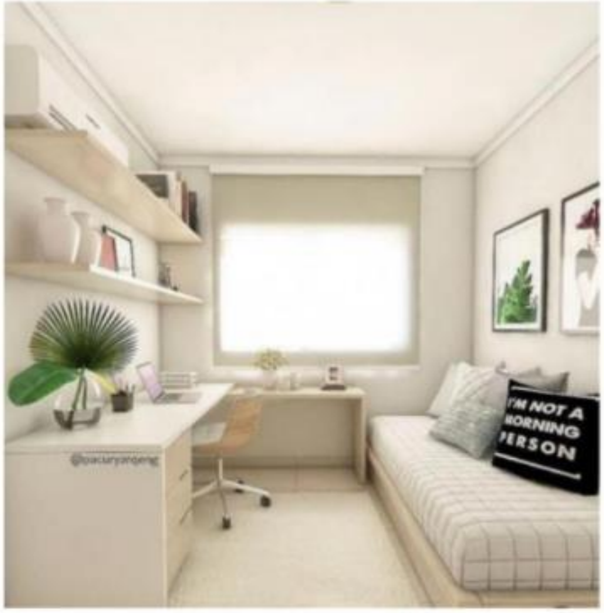
Objective

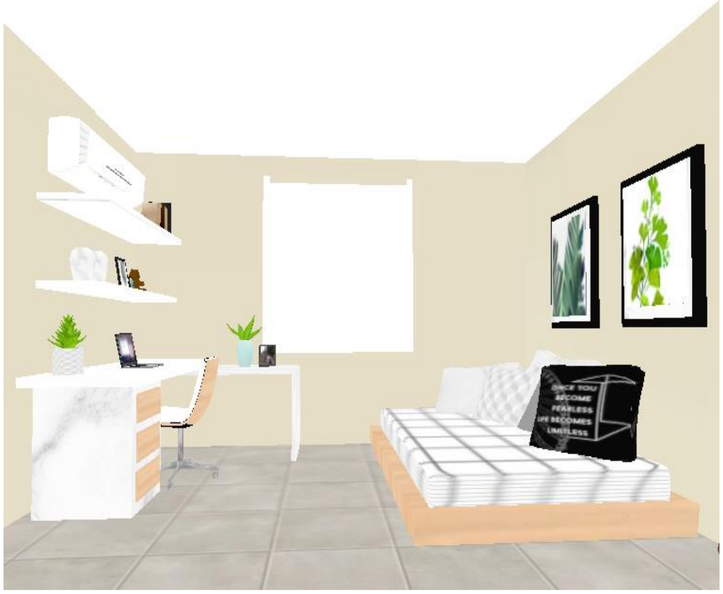
The student must apply and demonstrate the knowledge acquired throughout the course.

Facade and rooms to recreate:

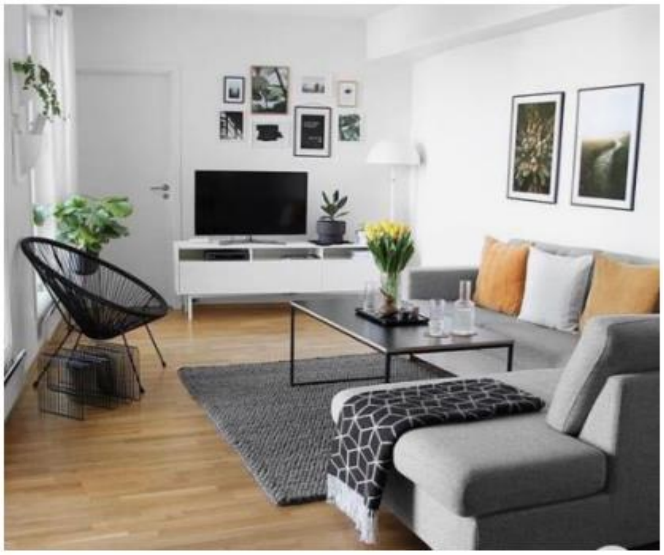


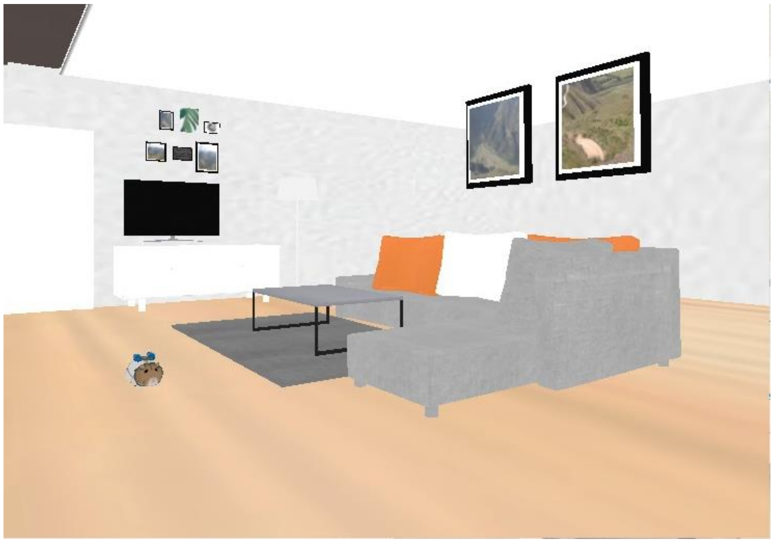




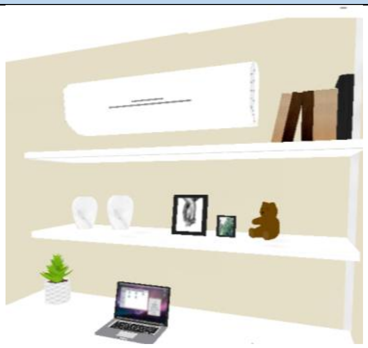
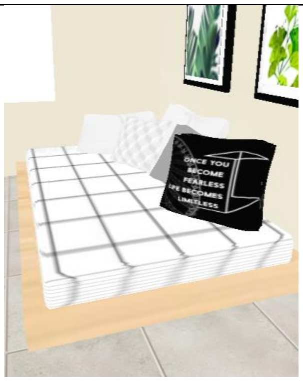


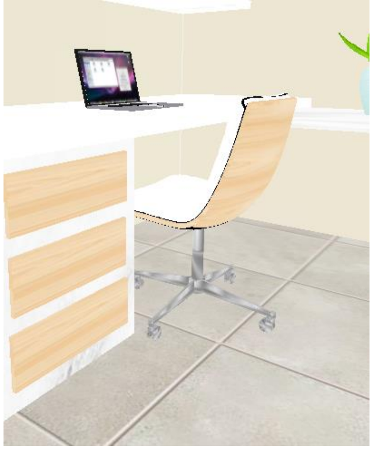
Cuarto 2:





Objects to recreate inside the rooms:

Objections	Reference image	Objeto en OpenGL 3D
Shelves		
Bed		

Desk		
Desk chair		
Vase		

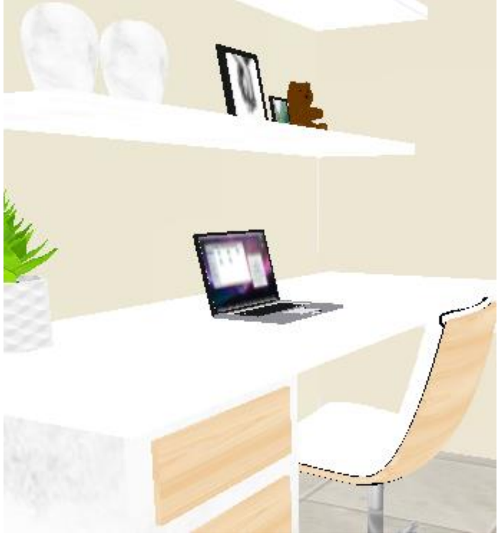
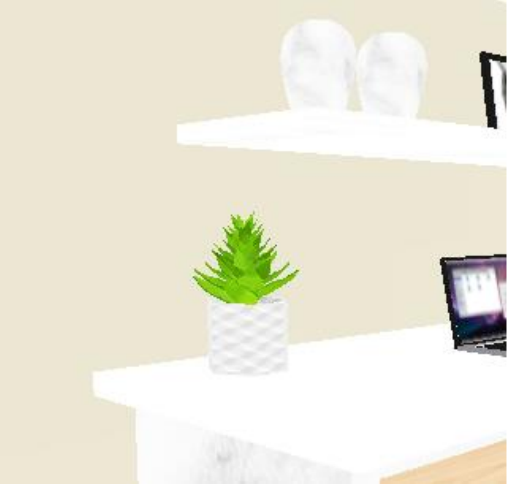
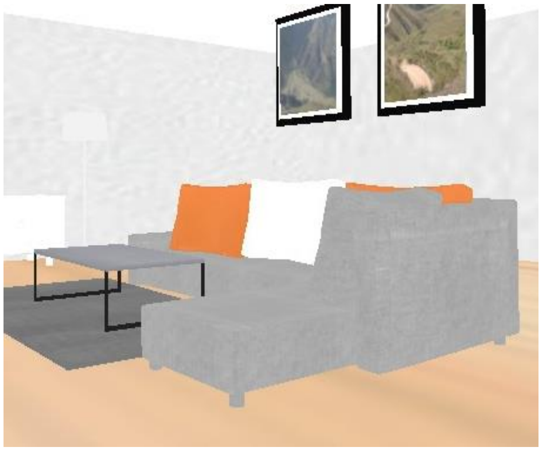
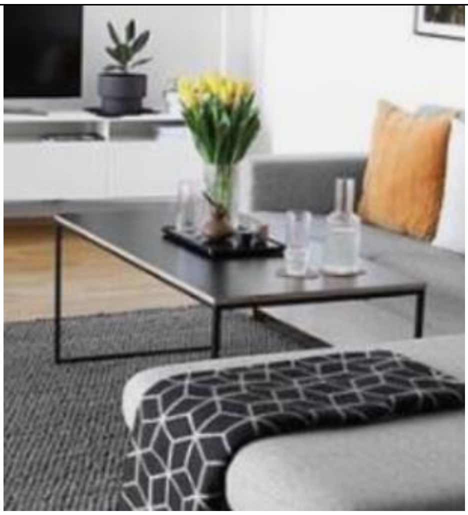
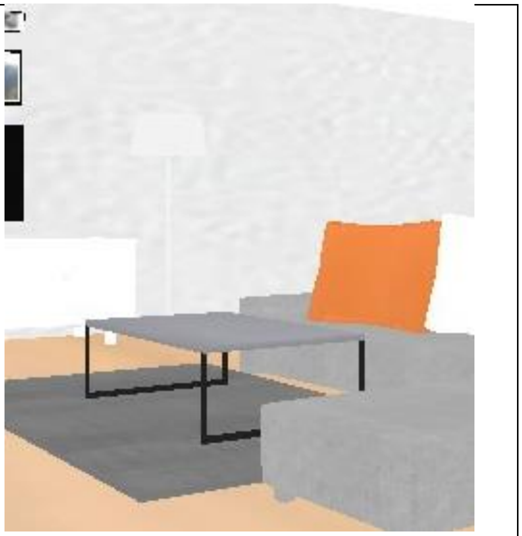
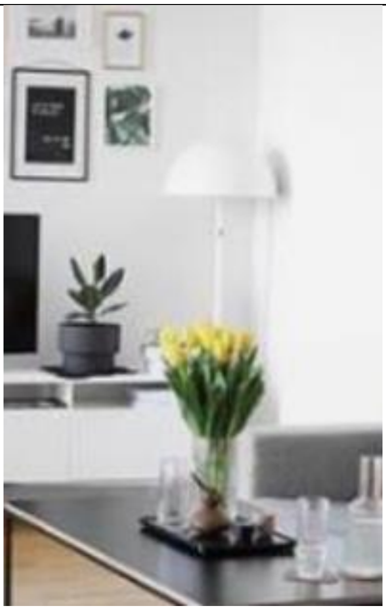
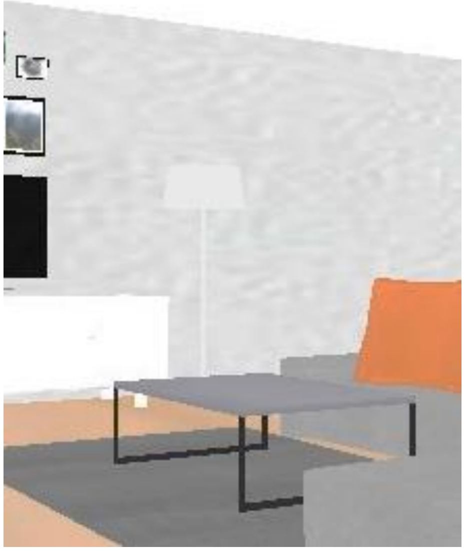
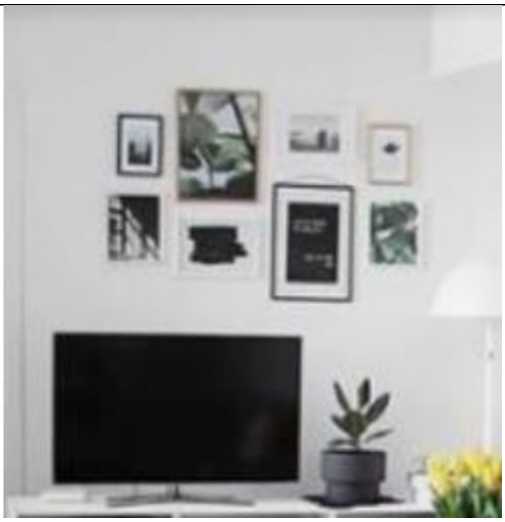
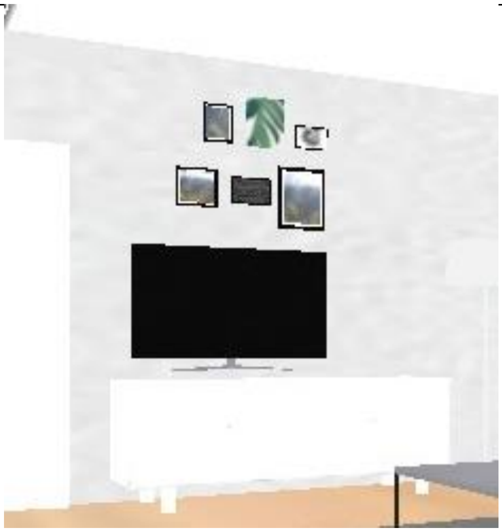
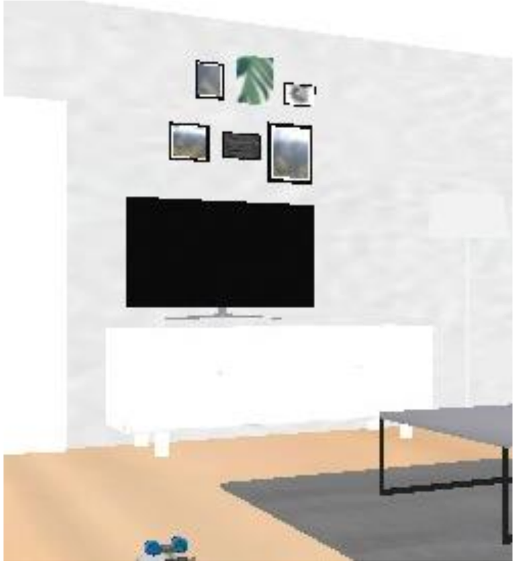
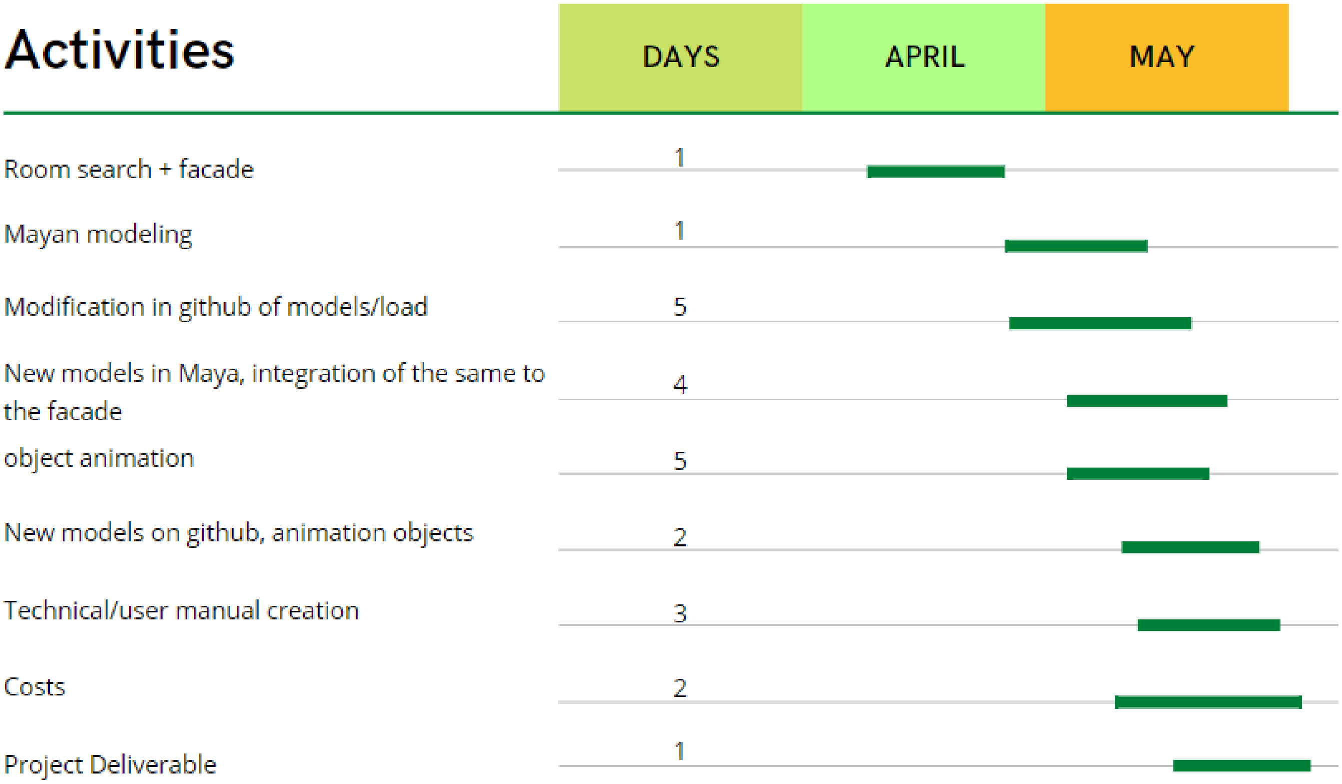
Laptop		
Planta		
Sofa		

Table		
Lamp		
Pictures		

LED TV		

Gant diagram



Activities	DAYS
Room search + facade	1
Mayan modeling	1
Modification in github of models/load	5
New models in Maya, integration of the same to the facade	4
object animation	5
New models on github, animation objects	2
Technical/user manual creation	3
Costs	2
Project Deliverable	1

Project Scope

The project seeks to model and recreate objects as realistic as possible (modern house) from a façade found on the internet, a modern room and in this way apply our skills acquired in the infographic laboratory course, adapting the elements and taking it to the GitHub version platform, creating animations within this environment, the activities that were carried out in order as they were modeled, it was required to upload to GitHub to be able to upload 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 then be built or implemented something technological 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 on to carry it out were few by the type of computer, this in the end complicated a little the execution of the program (OpenGL), already loading all the models within the main main, to carry out this project took a few months to change what is required. The expected results seek to meet the objective, to show the knowledge acquired throughout the semester.

Requirements: To do so 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 review of the project to be able to download and subsequently approve our final project.

Costs

In the realization of this project will be seen the salaries of the people that are required to be able to carry it out with a total of days of work. Taking into account that the project took a few days (1 month approximately). The maximum duration of the day will be: eight hours the day, seven the night and seven and a half hours the mixed.

The minimum amount per hour was taken into account for wages:

salary for the programmer → **$ 86.15** per hour

project leader salary → **$76. 92** per hour

business leader salary → **$80** per hour

designer salary → **$55.38** per hour

Work done	**Hours worked**	**Working days**	**Total→ salary**
Programmer	120	15	$10,338
Project Leader	56	7	$4,307. 52
Commercial Leader	32	4	$2,560
Designer	128	16	$7,088.64

Total $24,294.16→

Servicios/ for one month:

Electric power → $200

Internet → $800

Water → $200

Total, of services for one month = $1,200

Human resources

Considering slack of 7% and a profit of 50%

$$RH = (1.5)(24{,}294.16) + (0.7)(24{,}294.16) = \ 53447.152$$

Support/ monthly tools:

Stationery → $100

Extra models → $200

Total, Support Tools = $300

Net payment (PN):

$$PN = RH + herramientas\ de\ apoyo + servicios = \ 53447.152 + 300 + 1{,}200 = 54{,}947.152$$

Total project price →$\mathbf{54,947.152}$, with an additional cost for delay in project payments, must be started with a total 20% of the total price.

Code documentation

Libraries used:

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

GL includes:

Includes	Description
Shader.h	It is used to retrieve the code containing the vertex and fragment, allocating memory locations.
Camera.h	The synthetic camera is used to move within the environment.
Model.h	Import the models used "obj" loading the coordinates of the vertices of these objects to import them, load the textures.
stb_image.h	Upload images as textures, uploading the files to integrate them into the project.

Shaders

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

Functions:

Functions	Description
DoMovement	If there are animations within KellCallback, they change states by checking if the animation is active. .
KeyCallback	Captures keyboard information, processes this information for events.
animation	Function for animation of the mouse (toy) with parabolic shooting. In addition to the animation of the shower (using time sine).
MouseCallback	Function for animation of the mouse (toy) with parabolic shooting. In addition to the animation of the shower (using time sine).

Loading models

```
    Loading models
Model façade((char*)"Models/Façade/façade2.obj");
Model desk((char*)"Models/Desk/desk.obj");
Model frame((char*)"Models/Tables/frames.obj");
Model bed((char*)"Models/Bed/bed.obj");
Model chair((char*)"Models/Chair/chair.obj");
Model aire((char*)"Models/Aire/aire.obj");
Model shelf((char*)"Models/Shelf/shelf.obj");
Model planta((char*)"Models/Planta/planta.obj");
Model lap((char*)"Models/Laptop/laptop.obj");
Model skate((char*)"Models/Skateboard/skateboard.obj");
Model shower((char*)"Models/Shower/shower.obj");
Model book((char*)"Models/Book/book.obj");
Model raton((char*)"Models/Toy/toy.obj");
Model mat((char*)"Models/Mat/rug.obj");
Model sofa((char*)"Models/Sofa/sofa.obj");
Model cuadros2((char*)"Models/Cuadro2/cuadros2.obj");
Model mesita((char*)"Models/Mesita/mesita.obj");
Model lampara((char*)"Models/Lamp/lampara.obj");
Model muebletv((char*)"Models/Muebletv/muebletv.obj");
Model tele((char*)"Models/Tele/tele.obj");
Model tables3((char*)"Models/Tables3/tables3.obj");
Model door((char*)"Models/Door/door.obj");
Model hamster((char*)"Models/Hamster/hamster.obj");
```

The path for each specific model within the "Models" folder was specified, indicating the name of the obj for the model.h file to load.

Simple animations:

animation desk chair

variable animation	Tipo	Function
acsilla	Bool	Check if the feature is turned on
transilla	float	translate function, activates it
rotsilla	float	Function to rotate, activates it

Animation of patineta

Variable animation	Tipor	Function
acpat	Bool	Check if the feature is turned on
transpat	float	translate function, activates it
rotpat	float	Function to rotate, activates it

Book animation

Variable animation	Tipor	Funcion
transbookX	float	Translation function for the X axis
transbookY	float	Translation function for the Y axis
transbookZ	float	Translation function for the Z axis
rotbook	float	Function to rotate, activates it
acbook	Bool	Check if the feature is turned on

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.

Animations:

Animation of the shower:

```
void animationR()
{
   Movement of the shower using the variable time to oscillate
   if (circuitReg1)
   {
      if (route1)
      {
         movKitYReg += 0.5f;
         if (movKitYReg < 1.5)
         {
            path1 = false;
            path2 = true;
         }
      }
      if (tour2)
      {
         rotKitReg = -60;
         movKitZReg -= 0.5f;
         if (movKitZReg < -2.0)
         {
            path2 = false;
            path3 = true;
         }
      }
      if (tour3)
      {
         movKitZReg = sin(time + 5);
         movKitXReg = sin(time);
      }
   }
```

Watering can

Variable animation	Tipor	Funcion
movkitXReg	float	Move on the X axis
movkitYReg	float	Mueve on the Y axis, is activated so that the shower 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 looks like the shower is at an angle that reaches the plants.
time	float	Variable by time, to apply the sine function and can move undulatingly.

circuitReg1	Bool	travel function
Route1	Bool	Variable for route 1
Route2	Bool	Variable for route 2
Route3	Bool	Variable for route 3; apply the time sine function + the required rotation

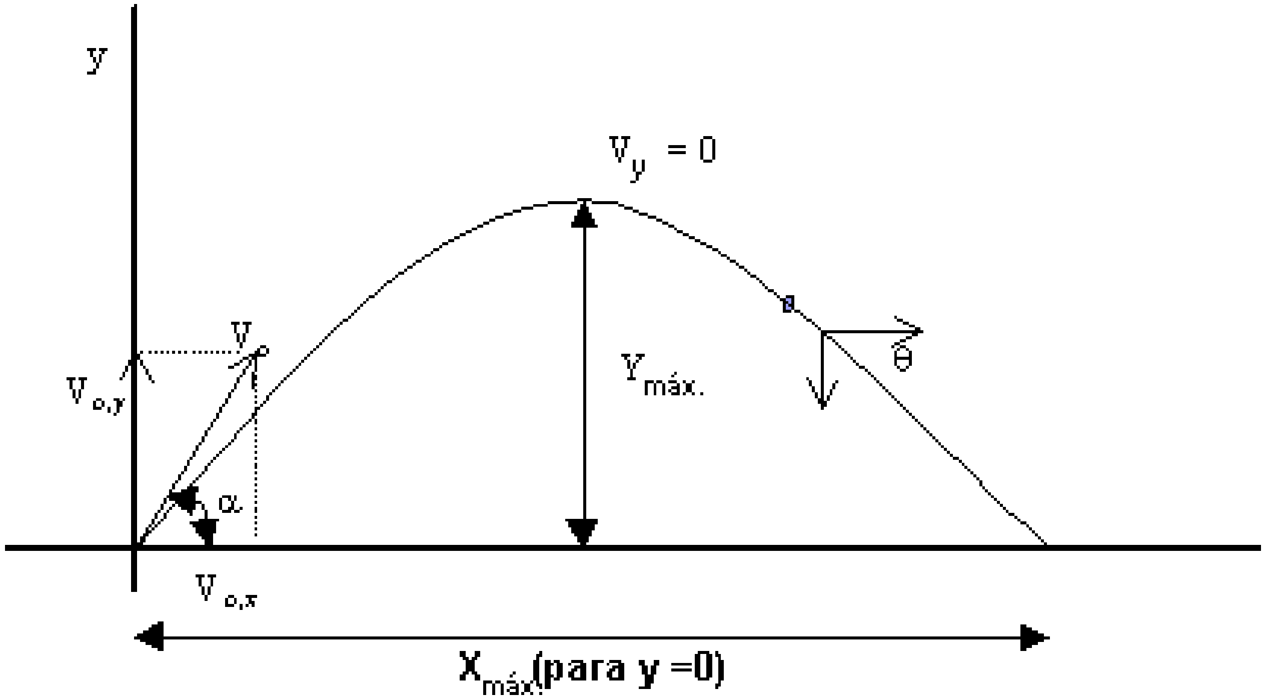
Toy mouse animation (car):

```
little mouse movement using parabolic shot
   if (circuitRat1)
   {
      if (route4)
      {
         rotKitRat = 0;
         v0y += 0.5f;
         v0z += 0.5f;
         if (v0z > 1.5 || v0y > 1.5) // calculate the maximum height of the          {
            path4 = false;
            path5 = true;
         }
      }
      if (route5)
      {
         v0y -= 0.5f;
         v0z += 0.5f;
         if (v0z > 3.5 || v0y < 0.0)   // height increases when it falls
         {
            path5 = false;
            path6 = true;
         }
      }
   }

   if (circuitRat2)
   {
      if (tour6)
      {
         rotKitRat = 180;     // returning the mouse to its initial position
         v0z -= 0.5f;
         if (v0z < 0.5)
         {
            path6 = false;
            path4 = true;
         }
      }

}
```

For parabolic plane animation (mouse)



Use ymax and xmax for this function so that the mouse does not go to the ground or the ceiling, voy and a VoX were required to be able to move, since when falling it will change speed, more force is required to return to the ground and have the original position, in this case the mouse at the time of falling will return with a trajectory on the ground "imitation of a toy car".

toy mouse:

Animationvariable	Tipor	Funcion
v0y	float	It moves on the Y axis in the parabolic plane, it is the maximum height in this case.
v0z	float	Move on the Z axis, in the parabolic plane

rotKitRat	float	Rotate to return the mouse to its starting position.
circuitRat1	Bool	In this route both x and y are increased to make it advance upwards, making a curve and reaching Ymax.
circuitRat2	Bool	Return path function for the mouse.
Route4	Bool	Identify and rotate the mouse to the maximum height to increase.
Route5	Bool	Increase the speed at which the toy mouse falls so that it reaches the other side as the curve descends beyond the maximum height it can move.
Route6	Bool	Move the mouse 180° to return to a point so that it can return to its original position.

Conclusions of the project:

This project helped me to realize the importance of graphic computing since when making 3D modeling and using programming to move our modeled objects, I could see the complexity of making a game, application, any project of this nature and especially the documentation is very important. I was able to apply my knowledge that I learned throughout the course since we could see modeling, texturingand animation. In this project

I had to take a lot of knowledge learned in the course such as texturing in a correct way, UV maps, since I had to load the images in powers of two and also program well, place the functions in correct order as seen in class, move, rotate, etc. Geometric transformations are of great help, they are geometric operations that allow you to create a new figure from one already made previously to this new figure we can call it homologa. The textures used required a correct size in order to perfectly cover the requested surface.

With respect to the models I could notice these represent geometric entities, full of triangles and figures, joining them can form any figure that we require, either lines, polygons or circumferences, when we model we must be careful to triangular since these figures require to be united in order to visualize it computationally.

Whenmaking cost estimates, we can analyze if we need at any time to sell the project we must take into account many factors.

This project also helps us to implement it in real life since we can model anything or engineering project, we could also 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 to obtain the results of the façade and the required rooms.

github link:

https://github.com/DianaCelesteHerrera/316161927_Proyecto_Gpo04.git