National Autonomous University of Mexico.

Faculty of Engineering.

Final Project.

Subject: Computing graphics and human-machine interaction.

Professor: Eng. Carlos Aldair Roman Balbuena.

Student:

Maceda Patricio Fernando
ID. 316054416

Semester: 2022-2.

Date: May 27, 2022.

Content.

Objective.

The student will do the final project of computer graphics theory and human-computer interaction, in which he will apply the concepts seen throughout the course. For the final laboratory project, a real housing complex is already modeled along with 5 animations, which was done in a team of 4 people. For the requirements of the theory project, two rooms furnished, and animations were added.

Requirements.

- The student must create a virtual environment, which will represent two rooms, made up of five elements to recreate which have already been approved by the teacher.
- Integrate synthetic camera.
- Must contain at least 4 animations.
- Project documentation must be delivered in both Spanish and English (Gantt chart, user manual where project objectives and interaction are set out, and technical manual).
- A cost analysis of the project must be added.
- The delivery of documentation must be done in digital format.
- You must submit an executable file.
- The realism of the space will be evaluated.
- The project must be on GitHub.

Introduction.

For this project, it was chosen to model the Villa Olímpica housing unit, located in Tlalpan in Mexico City, specifically on the nine-hectare property between Av. Insurgentes Sur and Av. Periférico, very close to Ciudad Universitaria.

> The buildings were built by the Banco Nacional de Obras y Servicios Públicos to house the Athletes of the Olympiad of Mexico, it consists of twenty-nine buildings that have a total of 5,044 rooms and 2,572 bathrooms in 904 apartments of the total of the buildings, of them 24 were destined for the male competitors and 3 for the female athletes, the remaining two were used for the press (Edificios de México, (s. f.)).

This place was chosen initially because a teammate lives there, which made it easier to model the objects that were placed in the project. Taking advantage of the fact that the buildings have the same shape, it would only be necessary to design a building and duplicate it as many times as desired, in addition, this place allows us to place various elements that allow us to cover the requirements imposed by the teacher. It should be noted that we gave ourselves the freedom to modify some objects in a way that could be consistent with our vision, such as reducing the number of buildings, the placement of an OXXO store and a court.

The rooms to model were obtained from the internet, they are student-themed rooms which have a bed and a few pieces of furniture, the reference images on which I base myself to model the rooms and the elements are attached.

Finally, we will show the design with the layout of the elements that we deliver in the project proposal.

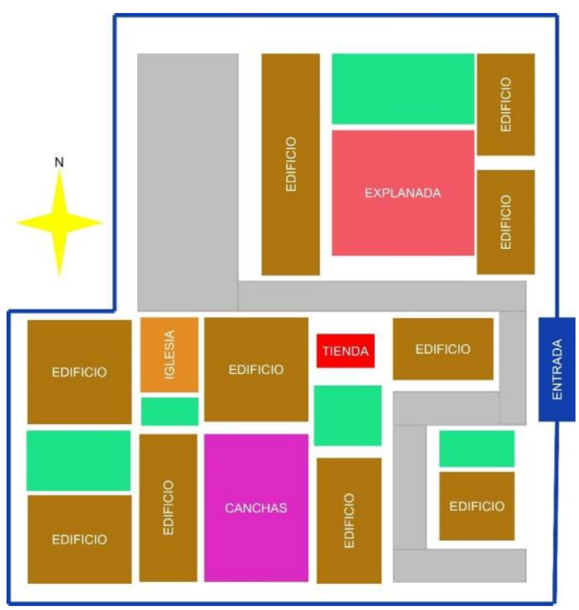


Figure 1. Map of the housing unit.

Figure 2. Room 1.

Figure 3. Room 2.

Development.

Model information.

Next, we will show the download data of the models obtained from the internet and some details of the models made by us.

Models Downloaded from Internet:

Model: Electric Floor Fan 3D Model.

Animation: Spinning blades animation.

Site: open3dmodel.com.

Figure 4. Fan.

Model: Hombre en chaqueta de pie y Rigged 3D Modelo.

Animation: Person drive tricycle.

Site: open3dmodel.com

Figure 5. Person.

Model: Man 3D Model.

Animation: Person drive bicycle.

Site: archive3d.net

Figure 6. Man on bicycle.

Model: Mini fridge.

Animation: Animation of fridge.

Site: 3dsky.org

Figure 7. Fridge.

Model: Pete

Animation: Left Strafe Walking.

Sitio de internet: Mixamo.

Figura 8. Man walking.

Model: Samoyedo Perro Modelo 3D

Animation: Dog walking.

Sitie: open3dmodel.com

Figure 9. Dog.

Model: Shannon.

Animation: Man running.

Site: Mixamo.

Figure 10. Athlete.

Model: Sporty Granny

Animation: Female Walk

Site: Mixamo.

Figure 11. Woman walk.

Model: Triciclo de reparto 2.

Animation: Tricycle around the housing unit.

Site: 3dwarehouse.sketchup.com.

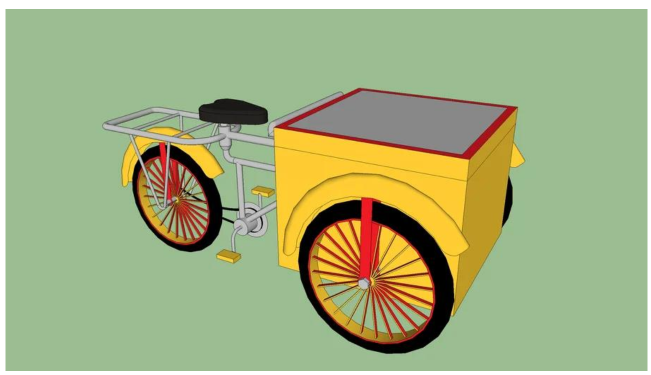


Figure 12. Tricycle*.*

Figure 13. Image of un mundo de tamal.

Model: Vintage Vw Volkswagen Beetle Car Modelo 3D.

Animation. Car around the housing unit.

Site: Open3DModel.

Figure 14. Beetle Car.

Made by the team:

Model: Desk.

Design application: Autodesk 3ds Max.

Figure 15. Desk.

Model: Bed 1.

Design application: Autodesk 3ds Max.

Figure 16. Bed 1.

Model: Bed 2.

Design application: Autodesk 3ds Max.

Figure 17. Bed 2.

Model: Court.

Design application: Blender.

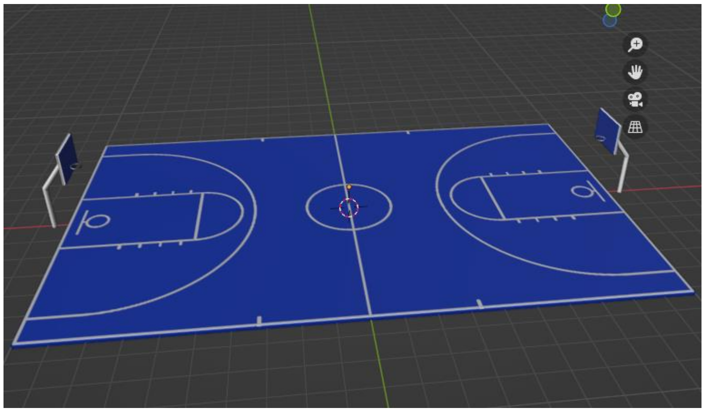


Figure 18. Court.

Model: Painting 1.

Design application: Autodesk 3ds Max.

Figura 19. Painting 1.

Model: Painting 2.

Design application: Autodesk 3ds Max.

Figura 20. Painting 2.

Model: Bulding.

Design application: Blender.

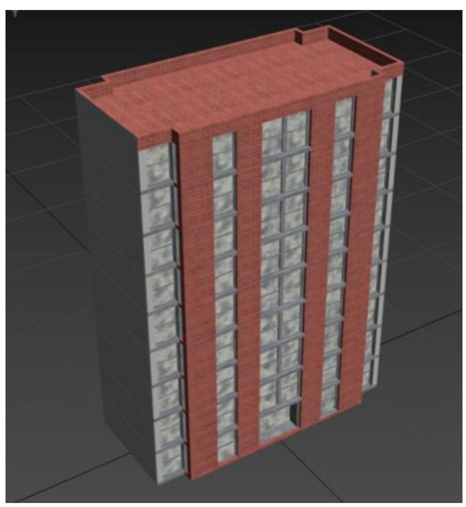


Figura 21. Bulding.

Model: Access.

Desing application: Autodesk 3ds Max.

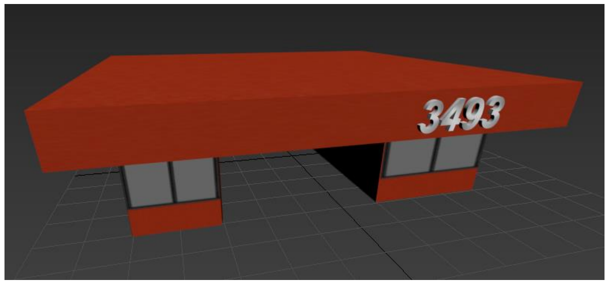


Figure 22. Access.

Figure 23. **Royal facade of the access.**

Model: Bookcase.

Design application: Autodesk 3ds Max.

Figure 24. Bookcase.

Model: Table.

Design application: Autodesk 3ds Max.

Figure 25. Table.

Model: Church.

Design application: Blender.

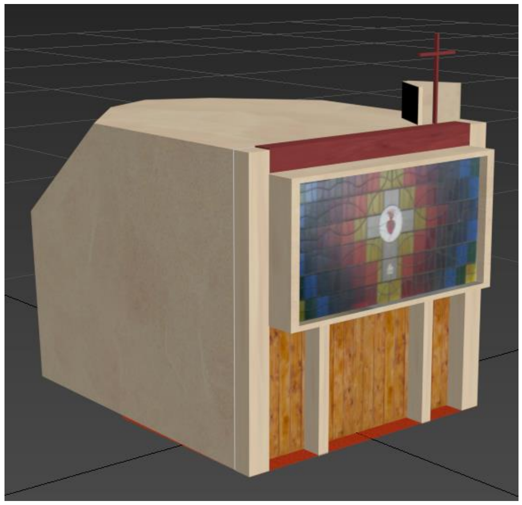


Figure 26. Church.

Model: OXXO.

Design application: Autodesk 3ds Max.

Figure 27. OXXO.

Figure 28. **Facade Oxxo.**

Model: Chair.

Design application: Autodesk 3ds Max.

Figura 29. Chair.

Model: TV.

Design application: Autodesk 3ds Max.

Figure 30. TV.

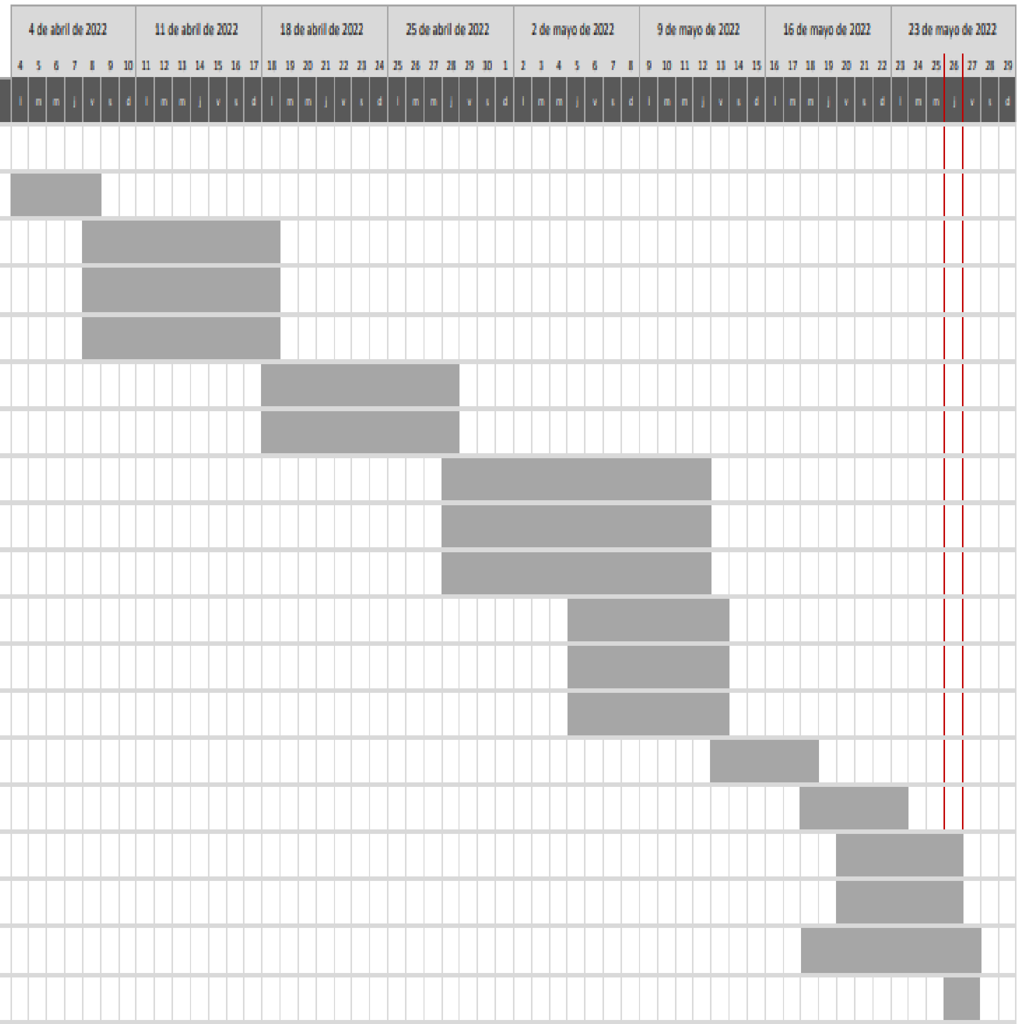
Schedule of activities.

Proyecto Final Teoria

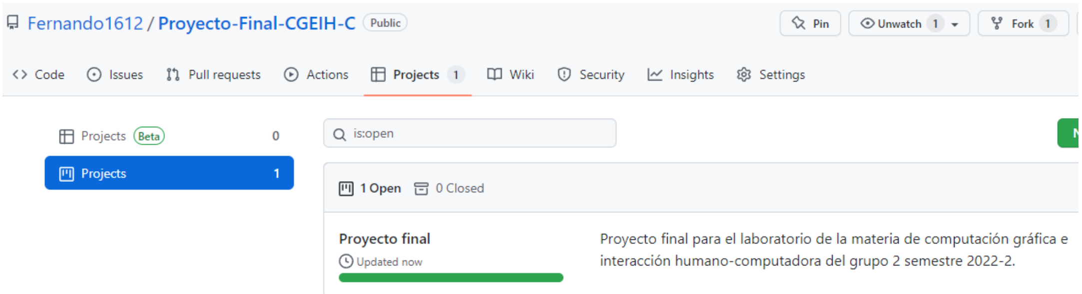
Fernando Maceda Patricio

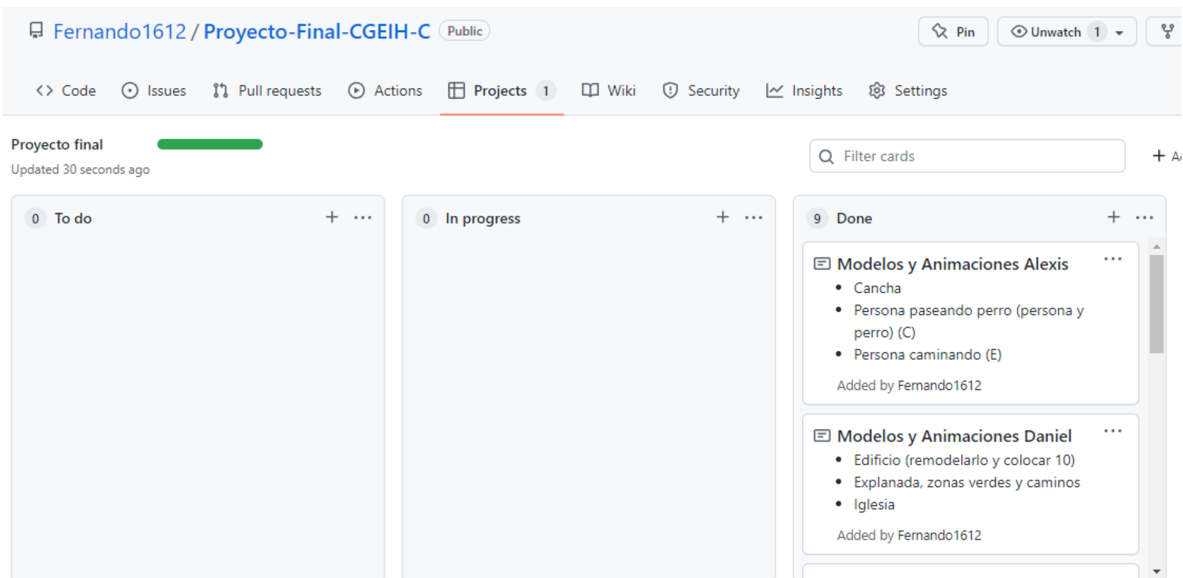
Inicio del proyecto:	vie, 4/1/2022
Semana para mostrar:	2

TAREA	ASIGNADO A	PROGRESO	INICIO	FIN
Proyecto Final				
Propuesta del proyecto	Fernando Maceda, David Reyes, Daniel Salinas. Alexis Vaquero	100%	1-4-22	8-4-22
Tienda (OXXO).	Fernando Maceda	100%	8-4-22	18-4-22
Triciclo - Persona	Fernando Maceda	100%	8-4-22	18-4-22
Persona en bicicleta.	Fernando Maceda	100%	8-4-22	18-4-22
Automóviles	David Reyes	100%	18-4-22	28-4-22
Persona-Cancha.	David Reyes	100%	18-4-22	28-4-22
Edificio (10).	Daniel Salinas	100%	28-4-22	12-5-22
Explanada, caminos y zona verde.	Daniel Salinas	100%	28-4-22	12-5-22
Iglesia.	Daniel Salinas	100%	28-4-22	12-5-22
Cancha.	Alexis Vaquero	100%	5-5-22	13-5-22
Persona-Perro.	Alexis Vaquero	100%	5-5-22	13-5-22
Persona caminando.	Alexis Vaquero	100%	5-5-22	13-5-22
Cuarto 1	Fernando Maceda	100%	13-5-22	18-5-22
Cuarto 2	Fernando Maceda	100%	18-5-22	23-5-22
Manual de ejecución	Fernando Maceda	100%	20-5-22	26-5-22
Manual de usuario	Fernando Maceda	100%	20-5-22	26-5-22
Documentación completa	Fernando Maceda	100%	18-5-22	27-5-22
Generación de archivo ejecutable	Fernando Maceda	100%	26-5-22	27-5-22



Kanban board.





Information of libraries, shaders, and functions.

Library	Description
GLEW, GLFW	For communication with the OS for handling windows in Visual Studio and OpenGL (creation, deletion, etc.).
GLM	C++ math library for projects on the OpenGL API.
Assimp	Library to import 3d models from programs like 3ds MAX, Maya, Blender, etc. and that OpenGL can understand its format, in our case it allows

Shader_m.h	It creates the vertex shader and fragment shader, all it receives is two addresses, the first for the vertex shader file and the second for the fragment shader file. It opens these files, recovering the code they contain and then creates both shaders by assigning them memory locations, they are compiled, it verifies that they have been created correctly and the program is created linking the created shaders (VAO, VBO and EBO), it compiles, it verifies the correct creation and at the end it deletes the information.
Camera.h	Create the synthetic or interactive camera, this .h defines the movements that the camera will have (forward, backward, left and right), the positions or initial values of the camera for rotation, tilt, speed of movement, sensitivity and zoom or angle aperture for perspective projection. For movements, define listeners or callbacks which will update the values when press the indicated keys or have mouse interaction for control of movement
Model.h	It receives the address of the .obj file and is responsible for importing said model using assimp, loading coordinates of its vertices (positions, normals and textures)
modelAnim.h	It receives the address of the .dae file and is responsible for importing said model using assimp, loading coordinates of its vertices (positions, normals and textures). It is used to import animations made in other software.

Shaders	**Description**
Skybox.frag y Skybox.vs	The vertex shader (.vs) captures the position of the image and sends it to the fragment (.frag) to link it to the cube.

lighting.vs y lighting.frag	The vertex shader accesses all the VAO coordinates (position, color and texture), receives the model, view and projection matrices to modify them according to the geometry received from the VAO. The fragPos and normals are calculated to be passed to the fragment along with the texture coordinates. The fragment shader defines the types of light that will be available (directional, point and spot) in the form of a structure with its components. It defines its functions for each type of light where each one calculates the illumination based on its diffuse, specular, etc. components. with the information of the vertex shader and the matrices of the program.
anim.vs y anim.frag	

Función	Descripción
my_input	Processes keyboard input, the keys pressed to perform an action. I know consider 7 keys specifically for the interaction with the virtual environment (“Esc”, “w”, “a”, “s”,”d”, "z" and "space").
mouse_callback	It is the mouse input listener. The action you define for that input is the update on the camera rotation
scroll_callback	It is the mouse scroll input listener. The action you define for that input is to zoom the camera
animate	This function defines the actions to perform if there are activated animations started on KeyCallback, when they change state and also the camera update.
music	Function used to activate the sound with the "z" key

main	Main function with the program to set the virtual environment. Create the window, check for errors, call listener functions or callbacks for keyboard and mouse events. Defines the viewport, enables OpenGL options for handling translucent and/or transparent materials. Create and configure the shaders. Define the geometry for the point light, skybox, and animation plane. Defines projection, model, view matrices. Captures the events of the mouse and keyboard inputs and motion listeners. Define the lighting. Model loading happens and basic transformations for accommodate them in space or animate them. Draw the point lights. Send to update the camera and by capturing events is listening for when command to terminate the program.

Information of animations.

Animation of Man on Bike:

The animation is divided into several objects:

- Man pedaling
- Bicycle frame
- Pedals
- Wheels

The animation of the man pedaling was created in 3ds Max using the Auto key tool. when drawing the man in the program we assign the variables of movement in x and z of the bicycle and rotation so that when the bicycle moves, the man does too.

We draw the bike with its initial values and create a temporary model. For the animation of the pedals and the wheels we add a rotation of the x axis.

For the animation of the bike, a route was assigned within the map where 6 states from 0 to 5 will be used using the Switch/Case control structure, the bicycle will change its position in the x axis and the z axis. This animation repeats itself infinitely since when reaching state

5 the bicycle is at the origin coordinates and returns to state 0, it is not necessary to press any key to start it either, the animation starts along with the program.

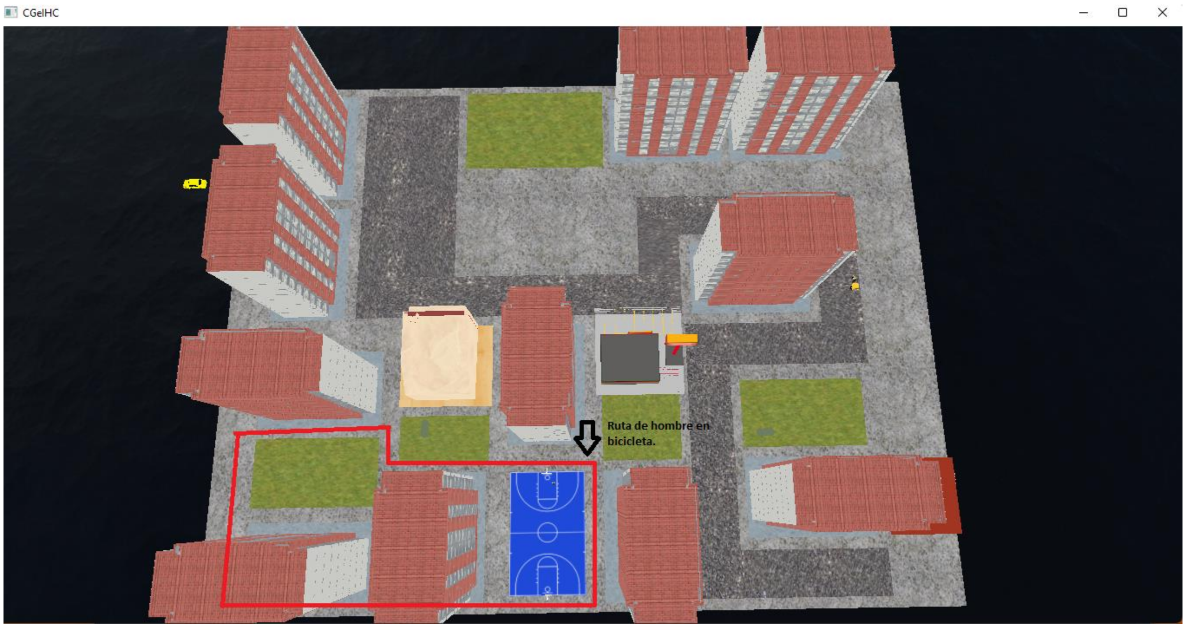


Figure 31. Bicycle animation route.

Animation of man selling tamales:

The animation is divided into several objects:

- Man pedaling
- Tricycle frame
- Pedals
- Wheels

The animation of the man pedaling was created in 3ds Max using the Auto key tool. At the time of drawing the man in the program we assign the variables of movement in x and z of the tricycle and of rotation so that when the bicycle moves, the man does too. At the time of loading the model it had an incorrect rotation, so two rotations were made, one at the beginning to accommodate it in the tricycle and the second that will change as the tricycle advances.

We draw the tricycle with its initial values and create a temporary model. For the animation of the pedals and the wheels we add a rotation of the x axis.

For the animation of the tricycle, a route was assigned within the map where the entire housing complex is traversed using 16 states from 0 to 15, using the Switch/Case control structure for changes in the x and z axis. This animation is repeated infinitely since when reaching state 15 the tricycle is in the origin coordinates and returns to state 0, it is not necessary to press any key to start it, the animation starts along with the program.

Figure 32. Route of the man selling tamales.

Animation of man doing sports on the court:

The animation is made up of a single object, the character Shannon with the Running animation, downloaded from the Miaxamo page. At the time of downloading, it was indicated that the animation should be in one place, so that the displacement would be generated based on the code. To draw the object, the variable that will allow it to move in the Z axis had to be placed, this taking advantage of the fact that it will not move in the X and Y axes; in conjunction with a variable that allows its rotation on the Y axis to simulate its change of direction.

Occupying a SWITCH, one chooses between the different states, which represent how far the object must go from the field. Within each CASE there is an IF statement that verifies if the object is advancing (towards higher values) or returning (towards smaller values). minors). If the condition that it reaches a certain distance is met, the object will make a 180° turn and will make the following movement by changing state.

For the animation to result in the character running to each mark and returning to the origin repeatedly, since the animation was not conditioned on user input when pressing a key.

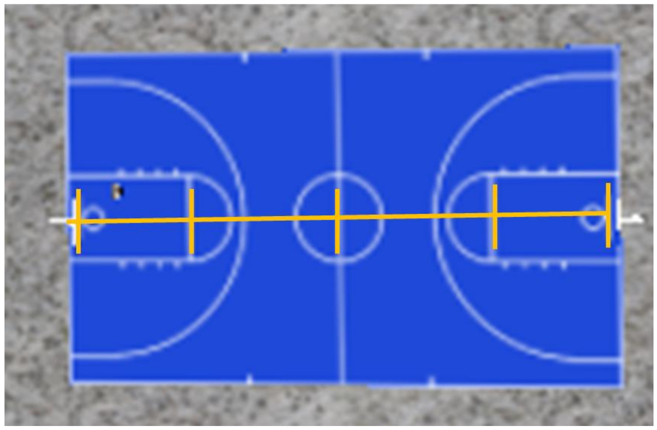


Figure 33. Court showing distances that the character travels.

Animation of Car:

The animation is made up of five objects, the body and four wheels, I think that the most complicated part of placing the objects was the proportions and placing the wheels in the correct position so that they do not look bigger than they should. The variables that allow its movement in X and Z and one to make the turn on Y were placed in the bodywork; for the wheels only one variable is used for its rotation about X.

For the car to make its movements, the user needs to press the SPACE key to move forward or stop. For the route, it was decided to start next to the entrance object and end stationary. The first thing to do is verify the state of the car so that it moves forward, and the wheels turn in one direction, then it asks if it has already reached a position that we select so that it makes a turn, changes state and continues moving forward; the positions depend on the distribution of the asphalt. I think that was the most difficult part of the animation, it was to see how far it should turn and not go through objects that we did not want.

Figure 34. Route of the car.

Animation of man walking:

For this animation, only one object downloaded from Mixamo was used. The character will be placed under the entrance and with two variables for its movement in X and Z it will move until it reaches the OXXO and with a variable in its rotation it will rotate on the Y axis.

Inside the animate function, an IF is placed so that the user presses the 1 key and thus the character begins its journey. Subsequently, the state in which it is found is verified so that it advances in the correct direction and when it reaches a certain location, it changes orientation and state. It should be clarified that in state five, the character will not advance if the car is going to cross the same spot, verifying that the state of the car is different from two; turn the character only through the OXXO object, the doors do not open for him to enter.

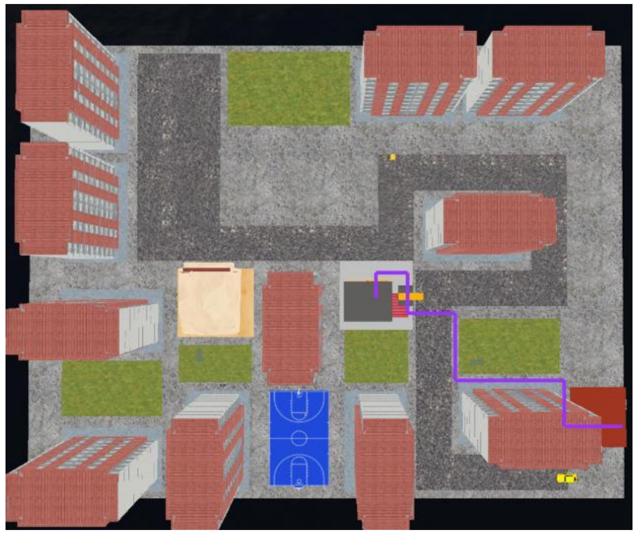


Figure 35. Route of Man walking.

Animation Person walking dog.

To create the corresponding animation, a constant was first defined, which will indicate the speed at which the characters as well as 5 floating variables, where these will help us move the objects on the X, Z axis and finally rotate these same.

Another variable that is also declared is the state of the animation, which will be the one that will define the change of state by means of a switch-case.

For the objects to perform the animation, it is important that where we draw them, we add the corresponding variables in the translation (X, Z), where these will be added to their defined position, so that the initial position will not be modified. In the same way, so that the objects can perform a rotation, the corresponding variable is added in this.

To create the states of the animation, the Switch-Case control structure is used, so that when the expected result is met, it moves to another case and continues the animation.

For each case there is an if-else which will verify that the movement in a certain axis does not exceed that determined by the developer, which consequently the objects will increase or decrease their position up to this limit in addition to rotating the established degrees. Finally, once the condition is not met, the dog will increase or decrease its position so that it gets the dog to the left side of the person and moves on to the next case (state of the animation with a maximum of 5 states).

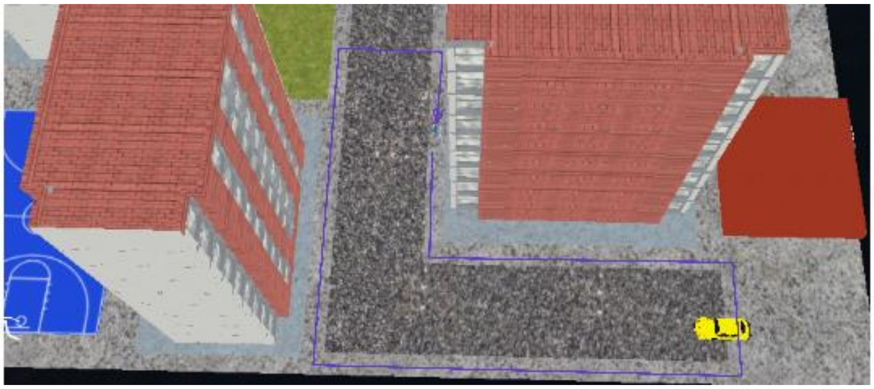


Figura 36. Route Person walking dog.

Animation Chair moving.

To make this animation, the only thing that is done is to move the chair in the z axis, for this we create a global variable with which it will change its value and with the help of a Switch structure we create 2 states which are the limits of the translation.

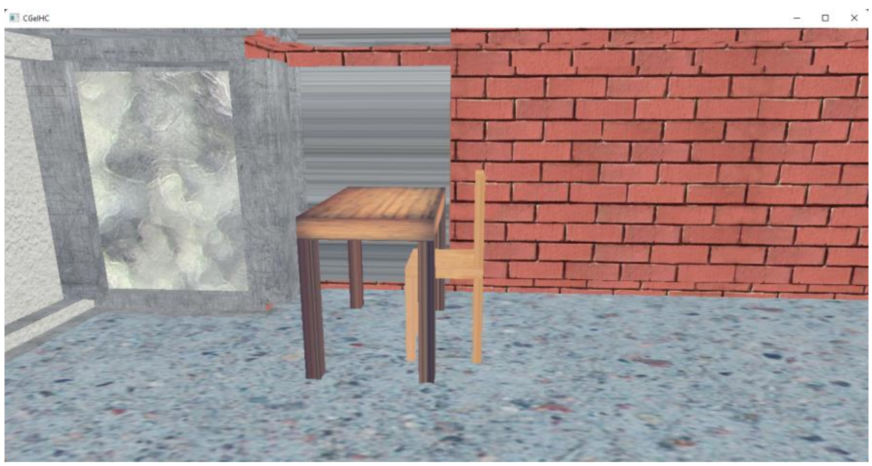


Figure 37. Chair and table.

Animation Refrigerator opening the door..

For this animation, the only thing that is done is a rotation in the Y axis of the door, just as in the chair, we create a variable that is going to change its value and two states for the limits of the rotation.

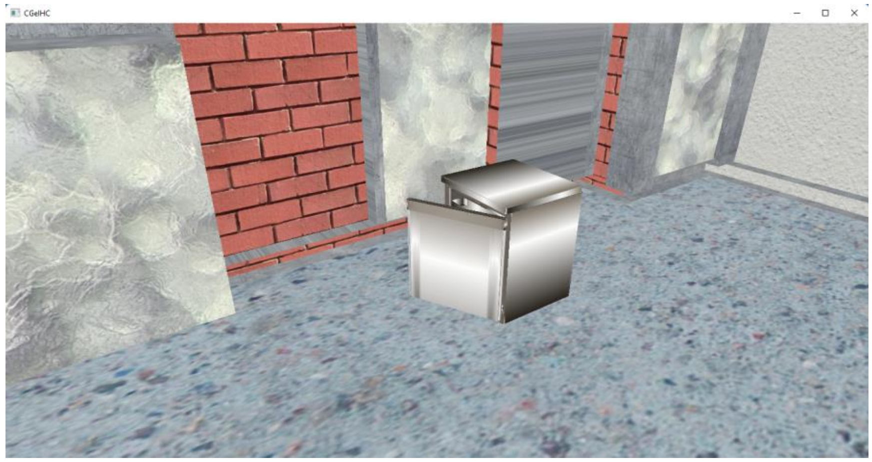


Figure 38. Refrigerator.

Animation Fan moving.

This animation was a little more complex since the fan cover rotates in the Y axis while the blades rotate in the X axis, for this we use two variables, one for each rotation and only two states for the rotation limits in Y.

Figure 39. Fan.

Technical study to determine the cost of the project:

The cost of the project is divided into two main items, the direct cost that is based on the total duration of the project and the indirect cost that depends on all the expenses associated with the project. To make the calculation we consider 4 members in the team.

The calculations are shown in the following tables:

Direct cost	CGeIHC Final Project									
	Amount	Staff	Rol	No. of projects you work on	Fee/h	Disposition	Hours/Month	Hours/Project	Salary/Month	Salary/Project
	1	Project Leader	X	5	$700.00	0.2	32	64	$22,400.00	$44,800.00
	3	Developer	Jr.	5	$350.00	0.2	32	64	$11,200.00	$22,400.00
				6		0.2	32	55	$11,2000.0	$19,250.00
				7		0.2	32	45	$11,200.00	$15,750.00
							Total	256		$102,200.00

Indirect cost	Amount	Concept	Unit dollar cost	Unit pesos cost	Total cost per item
	1	Car model	$0.00	$0.00	$0.00
	5	Person model	$0.00	$0.00	$0.00
	1	Dog model	$0.00	$0.00	$0.00
	1	Bicycle model	$0.00	$0.00	$0.00
	0	Server	$0.00	$0.00	$0.00
	0	Licenses /Month	$244.00	$4,824.37	$964.87
	1	Commercial space rent /Month	X	$15,000.00	$4,500.00
	X	Services pay /Month	X	$10,000.00	$3,000.00
				Total	$8,464.874

Direct cost:

For the calculation of the direct cost, a working day of 8 hours five days a week was considered, which results in 40 working hours of work; considering that the project lasted 8 weeks, equivalent to 320 hours.

Duration	hours
1 day	8
1 week	40
1 month	160
Project (8W)	320

Since we are four members in the team, one member was designated as project leader and the rest with the role of Jr. developer; It should be noted that the rates reflected in the tables were retrieved from a case study in the field of finance. At the same time, as all the members of the team have more than one subject, their availability would not be one hundred percent, for which the item "number of projects in which they participate" is equivalent to "number of subjects that are studying". The following columns are multiplications until the calculation of the total man-hours worked approximately 256, which implies a direct cost of $102,200.00.

Indirect cost:

In the indirect cost, all the elements used throughout the project must be considered, such as the models downloaded from the internet, which fortunately did not cost anything, there we consider the item of its price in US dollars since in the download platforms its value is found in this type of currency.

All implicit expenses must also be considered, such as the rent of a premises, licenses of the different software such as GitHub Pro, Visual Studio business and 3Ds Max, and the payment of the services that are needed (internet, electricity, water, etc. .). These expenses must be distributed in the same way among all the projects that are carried out by the members of the team as shown below, it should be clarified that it was multiplied between the provision of a larger member.

$$\$\,4{,}824.37 \cdot 0.2 = \$\,964.87$$

$$\$15{,}000.00 \cdot 0.2 = \$4{,}500.00$$

$$\$10{,}000.00 \cdot 0.2 = \$3{,}000.00$$

Resulting in a total indirect cost of $8,464.87.

Total cost:

The total cost of the project corresponds to the sum of both total values, the direct cost and the indirect cost, therefore, it is concluded that the total cost of the project is equal to \$110,664.87.

User Manual and Navigation Map:

[Proyecto-Final-CGEIH-C/README.en.md at 316054416_ProyectoFinalTeo_GPO4 · Fernando1612/Proyecto-Final-CGEIH-C (github.com)]

Video:

https://youtu.be/Yrzgm1TxxlU

Limitations.

The virtual environment would need to limit the camera's path so as not to go through the objects in space, mainly so as not to position itself below the plane, since the skybox would lose meaning.

The animations lacked a bit of complexity to not be practically linear.

Finally, the executable file can only be executed on Windows systems and with somewhat high resources to be able to display the entire environment correctly.

Extra activity

For extra activities I added the winmm library to play music in our program creating the music function that is activated when the "**Z**" key is pressed and using the PlaySound function it plays "TamalesOaxaqueños.wav".

Conclusions.

The realization of this project was very interesting, but a lot of work from modeling the objects, adding textures, animating them and finally adding them to the project, personally it took me more time to learn the different software for modeling, editing images and animating in addition to obviously programming I consider that a project like this cannot be carried out only if you want to obtain a good result, but rather a team of specialists in modeling, UX/UI, programmers, etc. is needed. This project helped me realize all the work and investment behind projects like video games and what skills are needed to dedicate oneself to that area.

References.

- Edificios de México (s.f.). Villa Olímpica Miguel Hidalgo [mensaje en un blog]. Recuperado de https://www.edemx.com/site/villa-olimpica-miguel-hidalgo/

- MIXAMO. (2022). Recuperado de https://www.mixamo.com/#/

- Open3DModel. (2022). Recuperado de https://open3dmodel.com/es/