



**UNIVERSIDAD  
NACIONAL  
AUTÓNOMA DE MÉXICO**  
FACULTAD DE INGENIERÍA



COMPUTACIÓN GRÁFICA E INTERACCIÓN  
HUMANO-COMPUTADORA

**FINAL PROJECT - TECHNICAL MANUAL**

**Team 10**

**NAME:**

Cruz Vázquez Zayra Sheccid

No. Cuenta: 316159126

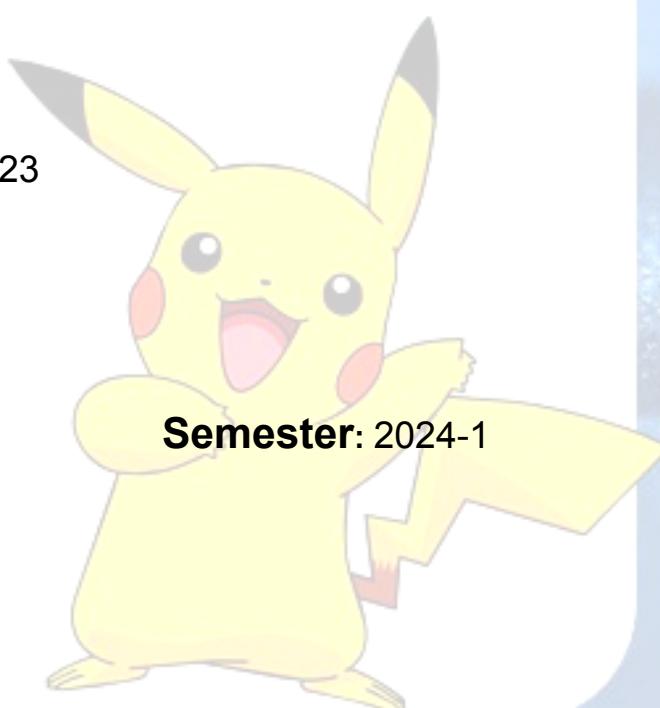
**PROFESSOR:**

Arturo Pérez De la Cruz

**Delivery:** Wednesday, 22 / November / 2023

**Group:** 1

**Semester:** 2024-1



## Description

This project has the main objective of providing Pokémon fans and anyone interested with a unique experience: a digital interactive visit to a museum dedicated to the iconic Pokémon franchise. In this project, our focus will be on the creation and 3D modeling of Pokémon figures to bring to life the rich history and world of these adorable characters that held significant meaning in the childhoods of countless individuals.

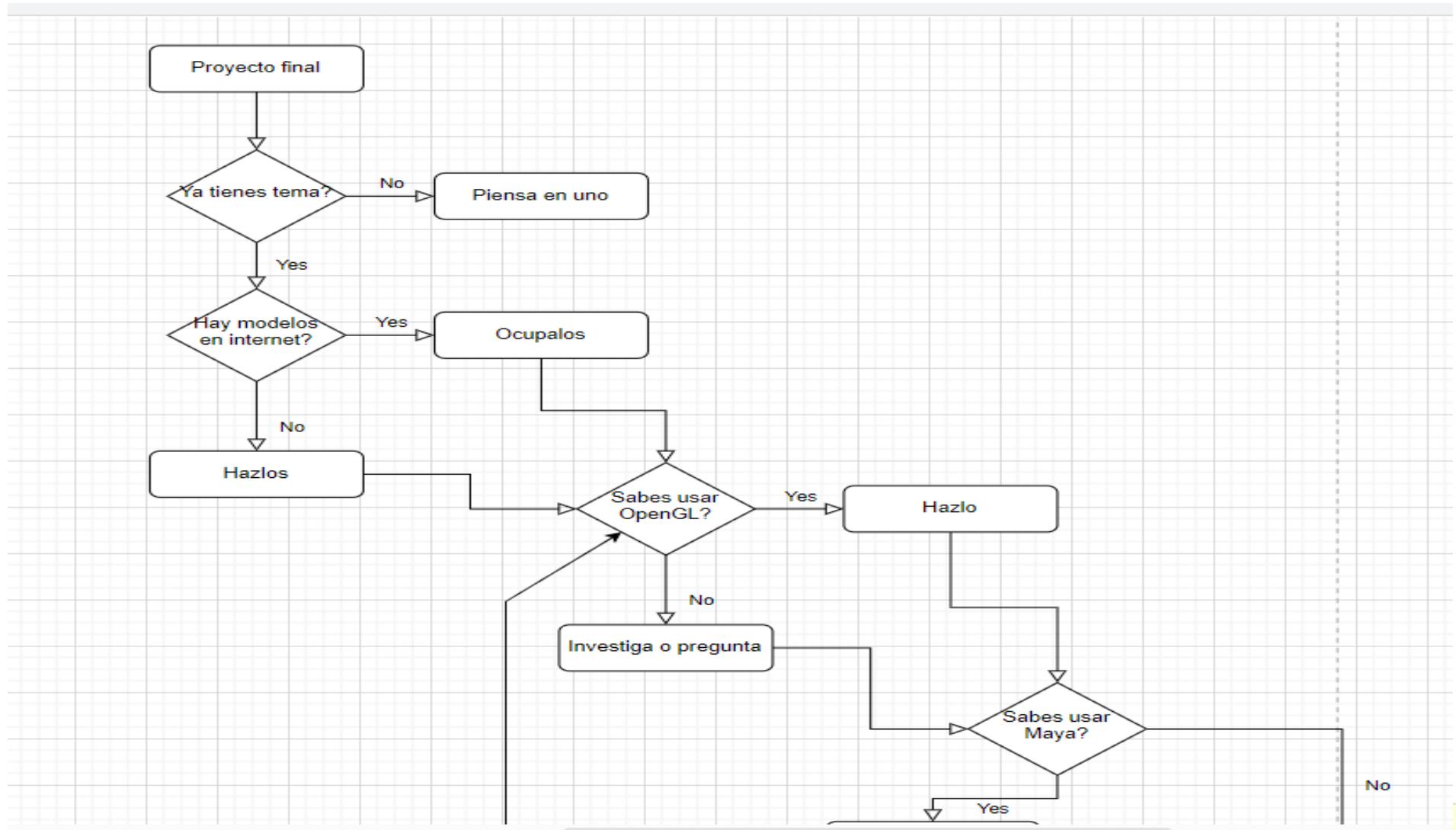
Additionally, it will serve as a tool to assist team members in developing, reinforcing, and applying the knowledge acquired in both theoretical and laboratory aspects of the Computer Graphics course. This will be done using applications such as Visual Studio and modeling software like Maya or similar tools.

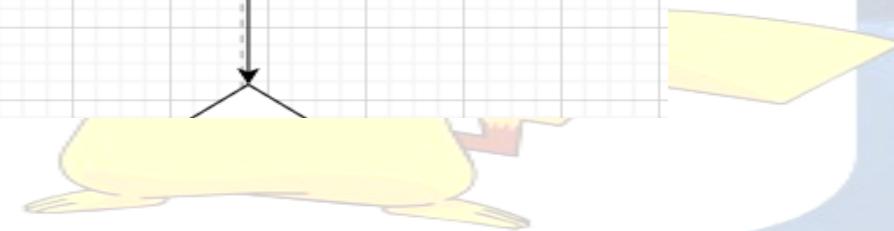
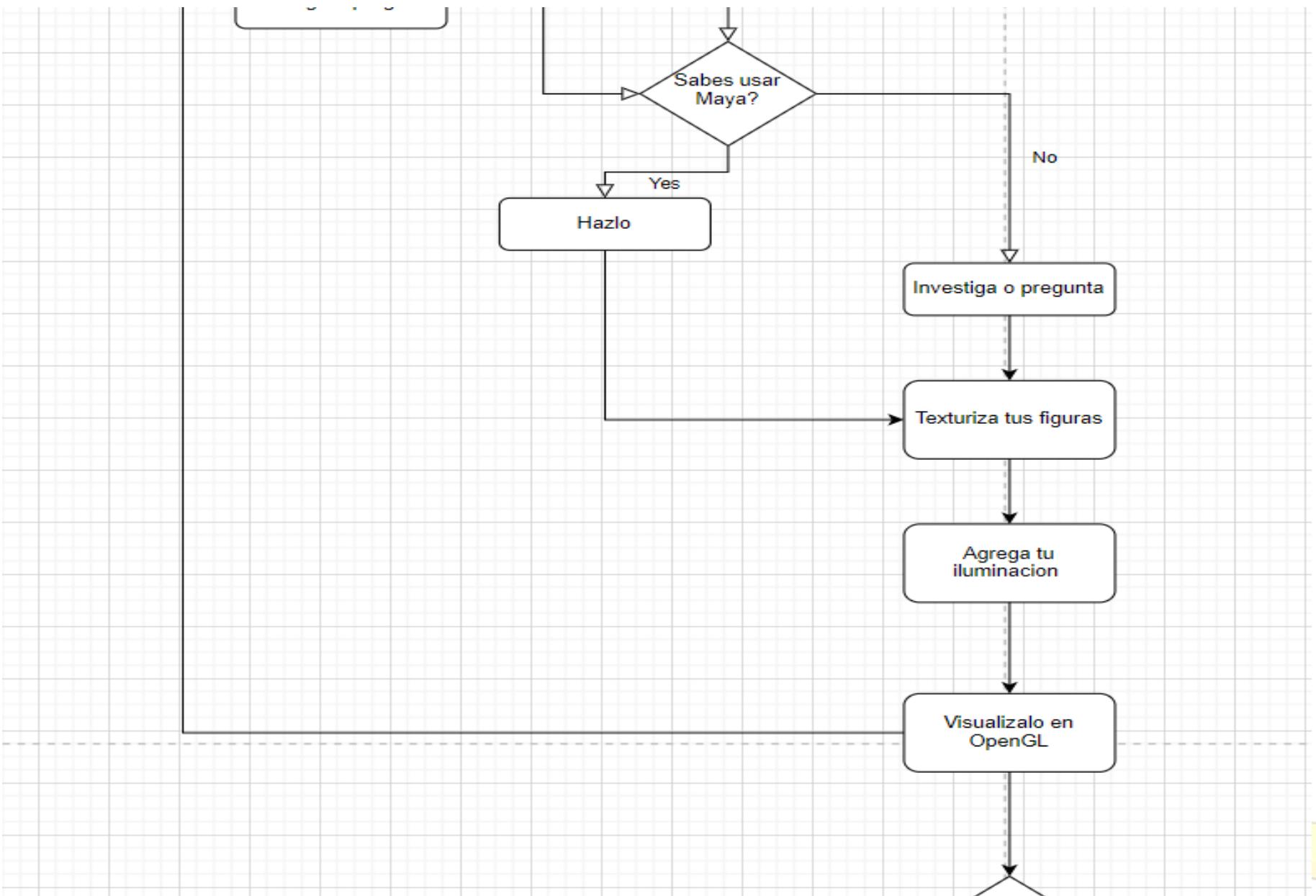
## Objectives

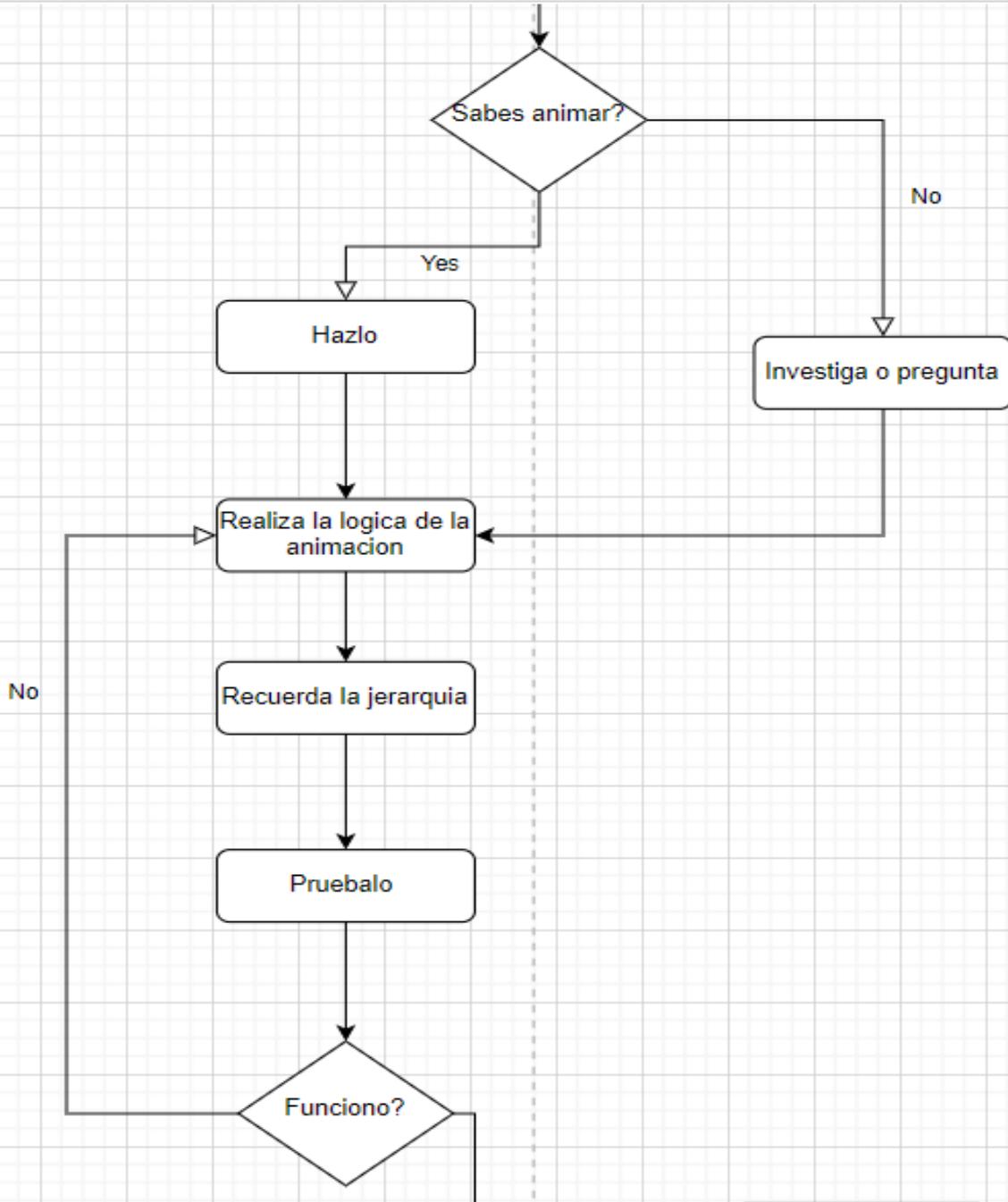
- Conduct a Digital Visit to the Pokémon Museum
- 3D Modeling in OpenGL
- Content Design in Autodesk Maya
- Integration of Animation Content
- Detailed documentation on the project along with a User Manual

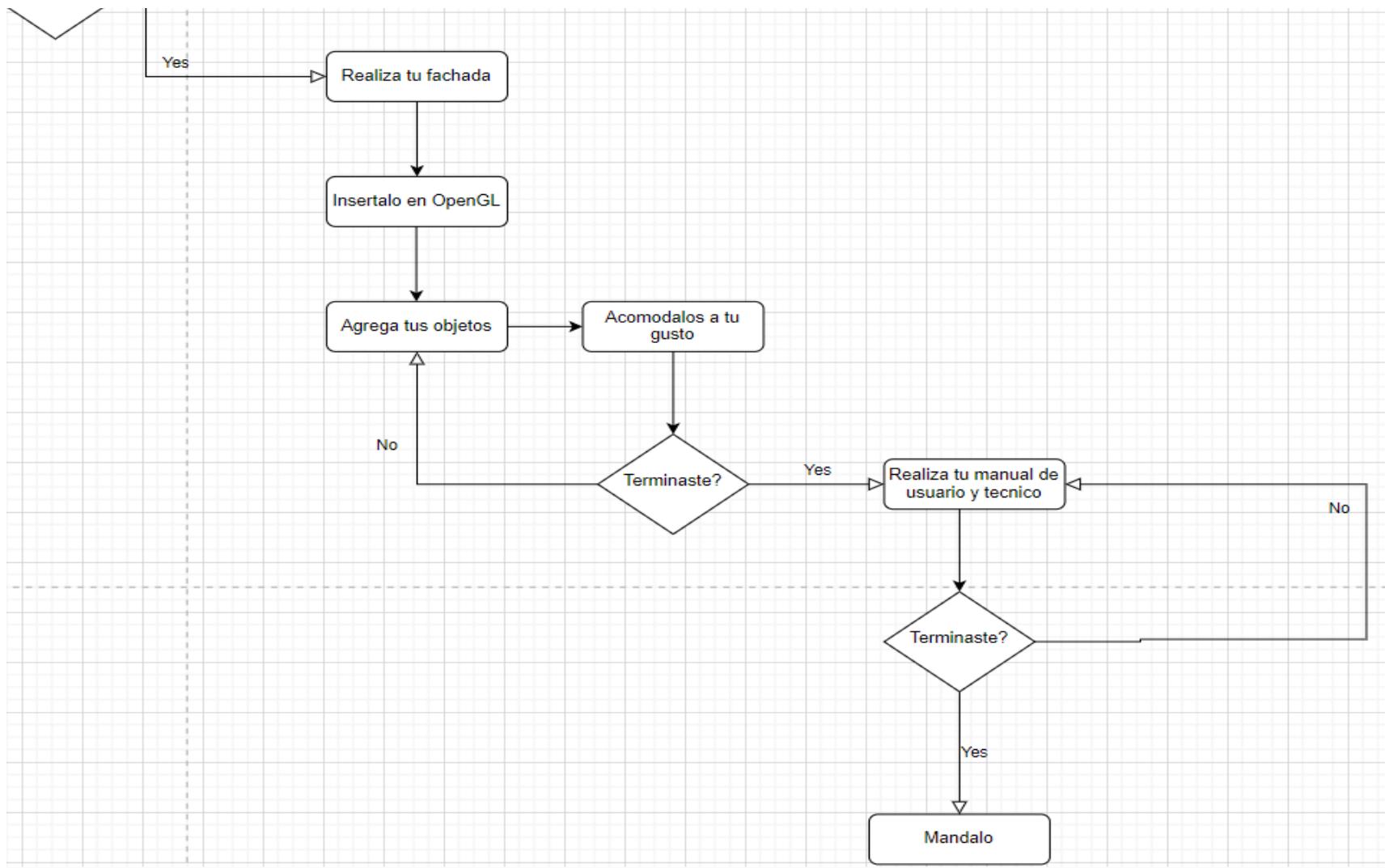


## Software Flow Diagram









For a better view, please visit

[https://drive.google.com/file/d/1x2DHGOBh9tpGSBkifst\\_IrC-t2XOyoID/view?usp=sharing](https://drive.google.com/file/d/1x2DHGOBh9tpGSBkifst_IrC-t2XOyoID/view?usp=sharing)



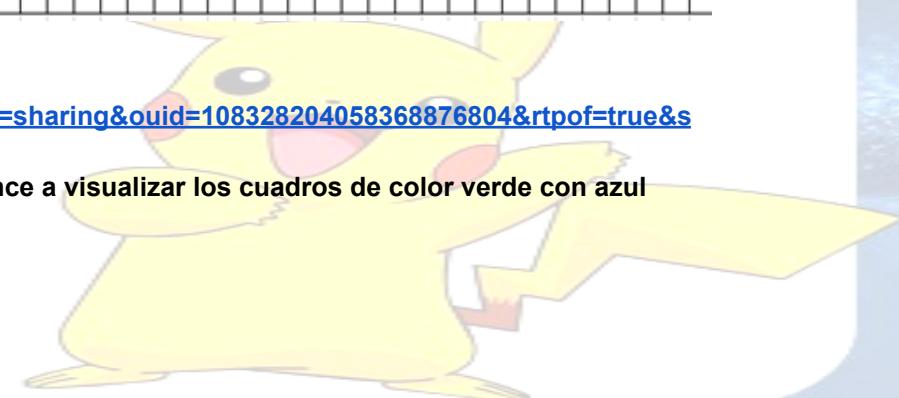
## Gantt Chart



**Para observarlo mejor visite:**

<https://docs.google.com/spreadsheets/d/1Mur-xOHc1p8pzACTKCz-J5z-BH864xnd/edit?usp=sharing&ouid=108328204058368876804&rtpof=true&sdr=true>

**NOTA:** El archivo se hizo en una versión de excel actual, así que es posible que no se alcance a visualizar los cuadros de color verde con azul



## Project Scope

The objective of this project is to recreate a space from the well-known Pokémon series, considering some Pokémon to give the impression of a museum, as well as a recreational area for them.

- Design

The design of the facade was based on the Pokémon GO video game with a conservative style to what is known.

The reception is designed for visitors like Pokémon, where you can see some historical elements such as hieroglyphs or the anatomy of a Pokémon, with a back area where Pokémon can play and interact.

- References

In the museum, you can find representative figures from the series, including: Pokeball, Mimikyu, and Mew.

- Ambiance

Some additional items were included alongside the chosen objects, such as the addition of Mew, lamps, pots, trees, and an outdoor table.



## Limitations

My limitations while working on this project throughout the semester were as follows:

- Lack of time

As this subject consumes a lot of time due to modeling, managing seven subjects made it difficult to dedicate time to this project. If I did, I had to dedicate a minimum of 2 hours a day from Friday, when the assignment was given, to Thursday, when reports and practices were due.

- Slow laptop

Similar to the above, having a computer without the necessary capacity to operate Maya or Visual Studio was challenging, as it took too long to open applications or compile code.

- Health problems

In the last two weeks before the project deadline, I underwent surgery, resulting in some time spent in the hospital. This situation, along with therapy sessions throughout the day, made it challenging.

- Lack of creativity

There was a moment where I got stuck at the end because I couldn't think of how to animate or what animation to apply. However, this turned out to be the least limiting factor as I received help from highly ingenious and creative minds suggesting animations for my objects.



## Software Methodology

- Planning

At the beginning of the course, we were informed that a project would be required for this subject, but the details were not specified at that time. After a month, both theory and lab professors provided project requirements. The theory project allowed freedom to choose a theme for a museum, real or fictional, while the lab project gave us the option to work on the same theory project, but not submit the same work. After forming the teams, we decided:

- Our theme would be a Pokémon museum.
- Each member would create a different museum room, later combining them using the GitHub platform.
- Each person would create their animations.
- Progress updates would be checked weekly.

- Analysis

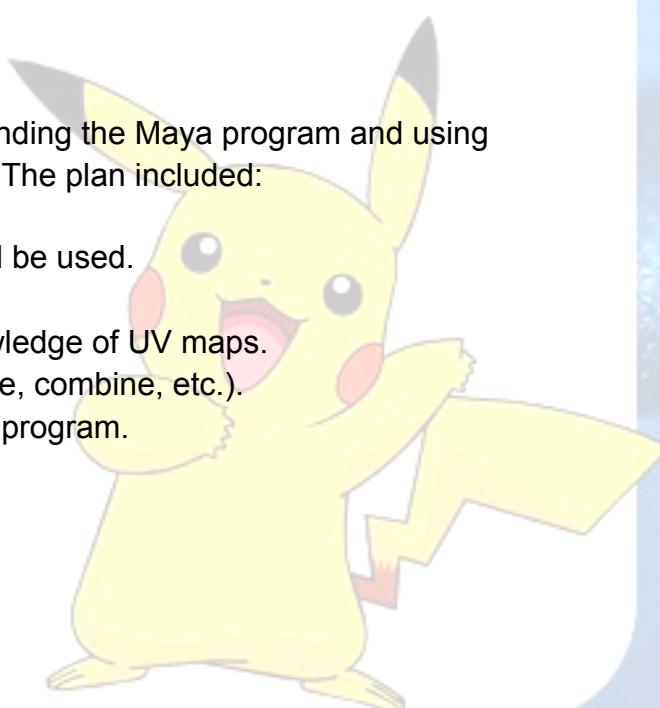
My lack of experience in the subject led me to find ways to be useful to my teammates, so the plan was:

- Be self-taught.
- Learn with the help of videos or like-minded individuals.
- Pay attention in lab classes.
- Ask questions about my doubts.

- Development

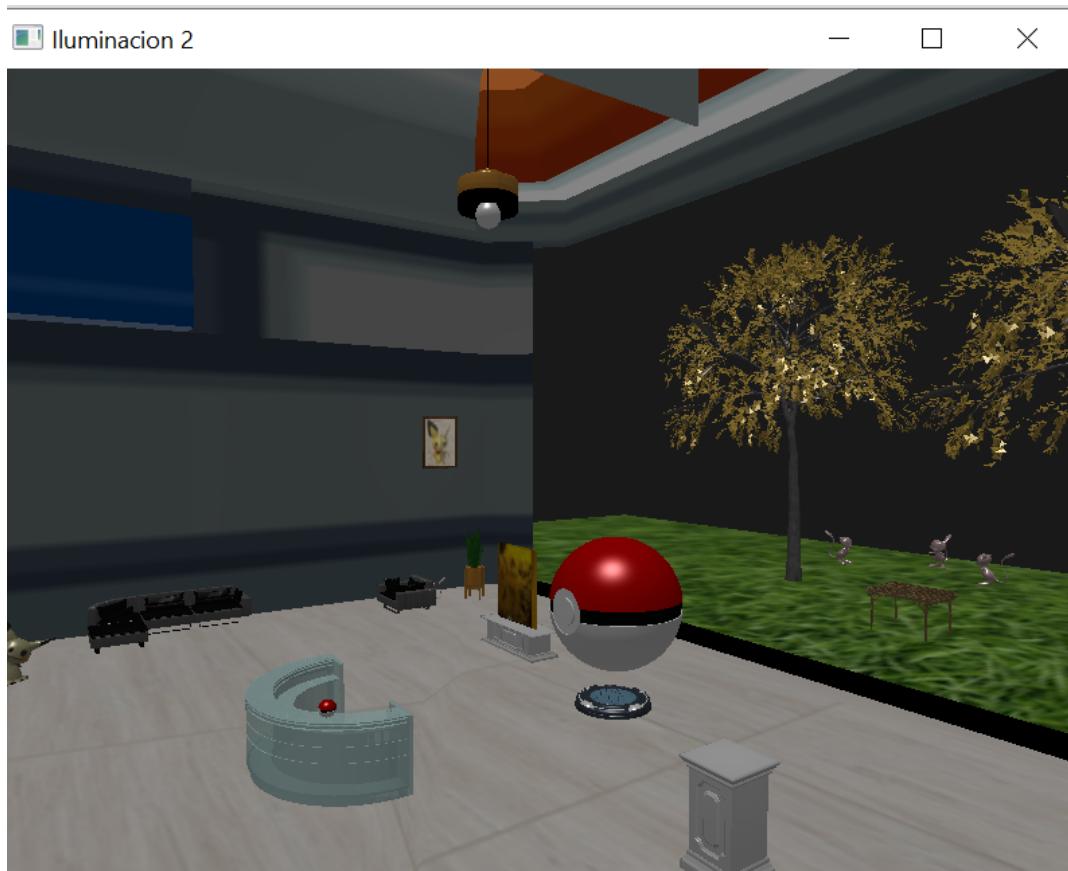
Unlike my team members, I was new to understanding the Maya program and using OpenGL, so at first, it was challenging to keep up. The plan included:

- Search for models on the internet that could be used.
- Learn to use the Maya program.
- Know how to apply textures and apply knowledge of UV maps.
- Use Maya's provided tools (separate, reduce, combine, etc.).
- Learn to reduce texture quality in the GIMP program.
- Use the pivot as a way to animate.
- Understand how to use shaders.
- Place lighting.



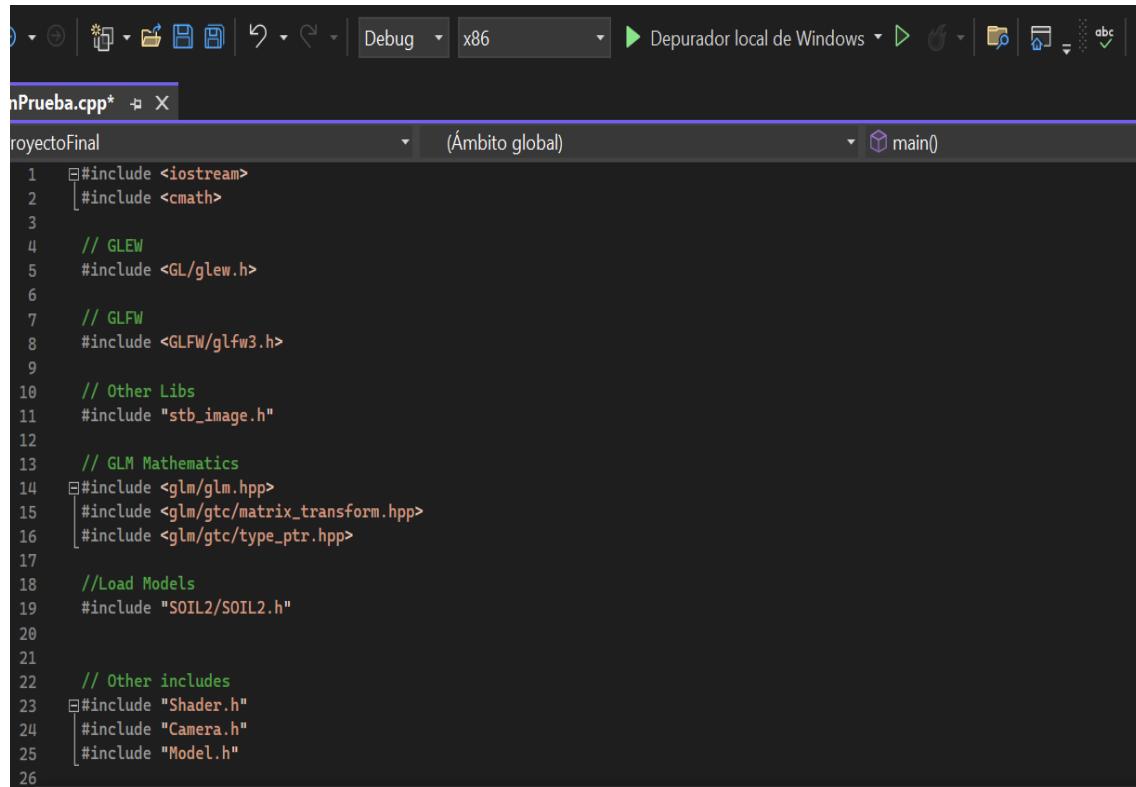
- Analyze what would be animated and choose the keys to use.
- Conduct tests and add or remove things depending on interests or perspective.

## Results



## Code Documentation

Libraries are added, and x64 is changed to x86.



```
Prueba.cpp* X
ProyectoFinal (Ámbito global) main()
1 #include <iostream>
2 [ #include <cmath>
3
4 // GLEW
5 #include <GL/glew.h>
6
7 // GLFW
8 #include <GLFW/glfw3.h>
9
10 // Other Libs
11 #include "stb_image.h"
12
13 // GLM Mathematics
14 #include <glm/glm.hpp>
15 [ #include <glm/gtc/matrix_transform.hpp>
16 [ #include <glm/gtc/type_ptr.hpp>
17
18 // Load Models
19 #include "SOIL2/SOIL2.h"
20
21
22 // Other includes
23 #include "Shader.h"
24 [ #include "Camera.h"
25 [ #include "Model.h"
26
```

Function prototypes are added, and the camera is configured.

```
// Function prototypes
void KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mode);
void MouseCallback(GLFWwindow* window, double xPos, double yPos);
void DoMovement();
void animacion();
void animacion1();

// Window dimensions
const GLuint WIDTH = 800, HEIGHT = 600;
int SCREEN_WIDTH, SCREEN_HEIGHT;

// Camera
Camera camera(glm::vec3(0.0f, 0.0f, 3.0f));
GLfloat lastX = WIDTH / 2.0;
GLfloat lastY = HEIGHT / 2.0;
bool keys[1024];
bool firstMouse = true;
// Light attributes
glm::vec3 lightPos(0.0f, 0.0f, 0.0f); // posicion
```

Variables to be used throughout the project are created.



```

50
51     bool active = false;
52     bool activAnim = true;
53     bool rotatePoke2 = false;
54     bool rotatePinto = false;
55     bool movePika = false;
56     bool animacionActivada = false;
57     bool activePot = true;
58     bool moveSofa1Positive = false;
59     bool moveSofa1Negative = false;
60     bool circuito = false;
61     bool recorrido1 = true;
62     bool recorrido2 = false;
63     bool recorrido3 = false;
64     bool recorrido4 = false;
65     bool recorrido5 = false;
66     bool recorrido6 = false;
67     bool circuito1 = false;
68     bool reco1 = true;
69     bool reco2 = false;
70     bool reco3 = false;
71     bool reco4 = false;
72     float pikaSpeed = 0.01f;
73     float poke2Rotation = 0.0f;
74     float pokeRotation = 0.0f;
75     float pikaPosX = 0.0f;
76     float pikaY = 0.0f;
77     float pikaPosY = 0.0f;
78     float sofa1PosZ = -4.0f;
79     const float sofaSpeed = 0.01f;
80     const float bounceSpeed = 1.5f;
81     const float rotationSpeed = 50.0f;
82     float movKitY = 0.0;
83     float movKitX = 0.0;
84     float movKitZ = 0.0;
85     float rotKit = 0.0f;

```

Point lights are positioned where desired.

```

99
100
101     // Positions of the point lights
102     glm::vec3 pointLightPositions[] = {
103         glm::vec3(7.0f, 1.0f, 0.0f),
104         glm::vec3(1.0f, 1.0f, -3.0f),
105         glm::vec3(-4.5f, 1.5f, -4.0f),
106         glm::vec3(0.0f, 12.8f, -0.12f)
107     };
108

```

Models to be used are created.

```

204
205     Shader lightingShader("Shaders/lighting.vs", "Shaders/lighting.frag");
206     Shader lampShader("Shaders/lamp.vs", "Shaders/lamp.frag");
207
208     Model sofa((char*)"Models/Pokemon/sofaPok1.obj");
209     Model sofa1((char*)"Models/Pokemon/sofaPok2.obj");
210     Model mesa((char*)"Models/Pokemon/mesaRecep.obj");
211     Model poke((char*)"Models/Pokemon/pokebola1.obj");
212     Model pedes((char*)"Models/Pokemon/pedes.obj");
213     Model pika((char*)"Models/Pokemon/pikachufanta.obj");
214     Model piso((char*)"Models/Pokemon/pisoMus.obj");
215     Model pedes2((char*)"Models/Pokemon/pedes2.obj");
216     Model jero((char*)"Models/Pokemon/jeropok.obj");
217     Model fachada((char*)"Models/Pokemon/fachadal.obj");
218     Model poke2((char*)"Models/Pokemon/poke2.obj");
219     Model pintura((char*)"Models/Pokemon/pinturaPok.obj");
220     Model pedes3((char*)"Models/Pokemon/pedesPok.obj");
221     Model skybox((char*)"Models/Pokemon/skybox.obj");
222     Model lampara((char*)"Models/Pokemon/lampara.obj");
223     Model maceta((char*)"Models/Pokemon/maceta.obj");
224     Model mesaext((char*)"Models/Pokemon/mesaext.obj");
225     Model mew((char*)"Models/Pokemon/mew.obj");
226
227

```

The main loop begins; it is important to call all the prototypes that will be used, in my case, it was what we already had plus animation and animation1.

```

// Game loop
while (!glfwWindowShouldClose(window))
{
    // Calculate deltatime of current frame
    GLfloat currentTime = glfwGetTime();
    deltaTime = currentTime - lastFrame;
    lastFrame = currentTime;

    // Check if any events have been activated (key pressed, mouse moved etc.) and call corresponding response functions
    glfwPollEvents();
    DoMovement();
    //animacion(deltaTime);
    animacion();
    animacion1();

    // Clear the colorbuffer
    glClearColor(0.1f, 0.1f, 0.1f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // OpenGL options
    glEnable(GL_DEPTH_TEST);

```

Point lights and the spot light are specified.

```

glUniform1i(glGetUniformLocation(lightingShader.Program, "material.diffuse"), 0);
glUniform1i(glGetUniformLocation(lightingShader.Program, "material.specular"), 1); //carga la especular comentar para no hacer calculo
glUniform1f(glGetUniformLocation(lightingShader.Program, "material.shininess", 32.0f);
GLint viewPosLoc = glGetUniformLocation(LightingShader.Program, "viewPos");
glUniform3f(viewPosLoc, camera.GetPosition().x, camera.GetPosition().y, camera.GetPosition().z);

// Directional light
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.direction"), -0.2f, -1.0f, -0.3f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.ambient"), 0.3f, 0.3f, 0.3f); //un numero grande lo hace mas brillante
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.diffuse"), 0.3f, 0.3f, 0.3f); //se cambia ambiental y difusa
glUniform3f(glGetUniformLocation(lightingShader.Program, "dirLight.specular"), 1.0f, 1.0f, 1.0f); //aumenta la intensidad
//transparencia imagenes png

// Point light 1
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[0].position"), pointLightPositions[0].x, pointLightPositions[0].y, pointLightPositions[0].z);
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[0].ambient"), 0.05f, 0.05f, 0.05f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[0].diffuse"), LightP1.x, LightP1.y, LightP1.z);
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[0].specular"), LightP1.x, LightP1.y, LightP1.z);
glUniform1f(glGetUniformLocation(lightingShader.Program, "pointLights[0].constant"), 1.0f);
glUniform1f(glGetUniformLocation(lightingShader.Program, "pointLights[0].linear"), 0.7f);
glUniform1f(glGetUniformLocation(lightingShader.Program, "pointLights[0].quadratic"), 1.8f);

// Point light 2
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[1].position"), pointLightPositions[1].x, pointLightPositions[1].y, pointLightPositions[1].z);
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[1].ambient"), 0.05f, 0.05f, 0.05f);
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[1].diffuse"), LightP2.x, LightP2.y, LightP2.z);
glUniform3f(glGetUniformLocation(lightingShader.Program, "pointLights[1].specular"), LightP2.x, LightP2.y, LightP2.z);
glUniform1f(glGetUniformLocation(lightingShader.Program, "pointLights[1].constant"), 1.0f);
glUniform1f(glGetUniformLocation(lightingShader.Program, "pointLights[1].linear"), 0.7f);
glUniform1f(glGetUniformLocation(lightingShader.Program, "pointLights[1].quadratic"), 1.8f);

// Point light 3

```

Models are called to be drawn and presented in OpenG



```

//Carga de modelo
view = camera.GetViewMatrix();
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));

//model = glm::mat4(1);
//skybox.Draw(lightingShader);

//fachada museo
model = glm::mat4(1);
fachada.Draw(lightingShader);

//mesa recepcion
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(0.0f, 0.22f, 4.0f));
glEnable(GL_BLEND); //Activa la funcionalidad para trabajar el canal alfa se pone objetos translúcidos
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA); //Operaciones de transparencia
glUniform4f(glGetUniformLocation(lightingShader.Program, "colorAlfa"), 1.0f, 1.0f, 1.0f, 0.3f); //TRANSPARENCIA definir
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
mesa.Draw(lightingShader);
glDisable(GL_BLEND); //Desactiva el canal alfa

//sofa
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-13.0f, -0.5f, 3.5f));
model = glm::rotate(model, glm::radians(90.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
sofa.Draw(lightingShader);

//sofá

//maceta
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(11.0f, -0.47f, -6.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
maceta.Draw(lightingShader);

//maceta
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-11.0f, -0.47f, -6.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
maceta.Draw(lightingShader);

//mesa exterior
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(0.0f, -0.47f, -18.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
mesaext.Draw(lightingShader);

//mew
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(0.0f, 1.65f, -18.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
mew.Draw(lightingShader);

//mew
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(2.0f, -0.47f, -18.0f));
model = glm::rotate(model, glm::radians(270.0f), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
mew.Draw(lightingShader);

glBindVertexArray(0);

```

DoMovement contains the ranges or keys that will make our objects move.



```

540 // Moves/alters the camera positions based on user input
541 void DoMovement()
542 {
543
544     // Camera controls
545     if (keys[GLFW_KEY_W] || keys[GLFW_KEY_UP])
546     {
547         camera.ProcessKeyboard(FORWARD, deltaTime);
548     }
549
550     if (keys[GLFW_KEY_S] || keys[GLFW_KEY_DOWN])
551     {
552         camera.ProcessKeyboard(BACKWARD, deltaTime);
553     }
554
555     if (keys[GLFW_KEY_A] || keys[GLFW_KEY_LEFT])
556     {
557         camera.ProcessKeyboard(LEFT, deltaTime);
558     }
559
560     if (keys[GLFW_KEY_D] || keys[GLFW_KEY_RIGHT])
561     {
562         camera.ProcessKeyboard(RIGHT, deltaTime);
563     }
564
565 }
566
567
568

```

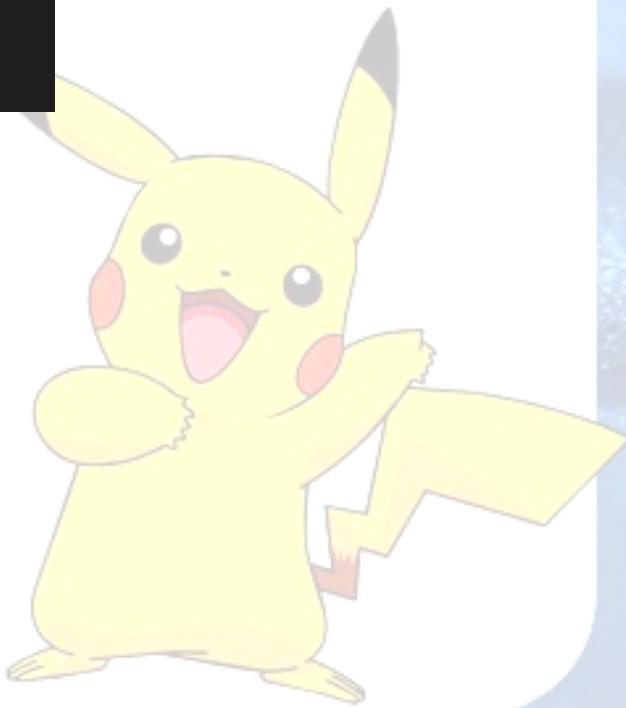
KeyCallback includes the preferred keys, and animation is initialized.

```

660 void KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mode)
661 {
662     if (key == GLFW_KEY_Z && action == GLFW_PRESS)
663     {
664         rotatePoke2 = !rotatePoke2; // Cambia el estado de la rotación
665     }
666
667     if (key == GLFW_KEY_SPACE && action == GLFW_PRESS)
668     {
669         activePot = false; // Cambia el estado de la rotación
670     }
671
672     if (key == GLFW_KEY_B && !rotatePintu) {
673         rotatePintu = true; // Activa la rotación de la pintura
674     }
675
676     if (keys[GLFW_KEY_R] && !movePika) {
677         movePika = true;
678     }
679
680     if (keys[GLFW_KEY_1] && !moveSofalNegative) {
681         moveSofalNegative = true;
682     }
683
684     if (keys[GLFW_KEY_2] && !moveSofalPositive) {
685         moveSofalPositive = true;
686     }
687
688     if (GLFW_KEY_ESCAPE == key && GLFW_PRESS == action)
689     {
690         glfwSetWindowShouldClose(window, GL_TRUE);
691     }
692
693

```

Mouse movement on the machine is processed.



```

void MouseCallback(GLFWwindow* window, double xPos, double yPos)
{
    if (firstMouse)
    {
        lastX = xPos;
        lastY = yPos;
        firstMouse = false;
    }

    GLfloat xOffset = xPos - lastX;
    GLfloat yOffset = lastY - yPos; // Reversed since y-coordinates go from bottom to left

    lastX = xPos;
    lastY = yPos;

    camera.ProcessMouseMovement(xOffset, yOffset);
}

```

Void functions are created for complex animations.

```

781 void animacion() //circuito de la pokebola
782 {
783     ...
784     if (circuito)
785     {
786         if (recorrido1)
787         {
788             movKitY += 0.1f;
789             movKitX += 0.1f;
790             if (movKitY > 2)
791             {
792                 recorrido1 = false;
793                 recorrido2 = true;
794             }
795         }
796         else if (recorrido2)
797         {
798             movKitY -= 0.1f;
799             if (movKitY < -1.5)
800             {
801                 recorrido2 = false;
802                 recorrido3 = true;
803             }
804         }
805         else if (recorrido3)
806         {
807             movKitY += 0.1f;
808             movKitX += 0.1f;
809             if (movKitY > 0.5)
810             {
811                 recorrido3 = false;
812                 recorrido4 = true;
813             }
814         }
815     }
816     if (recorrido1)
817 }

```

```

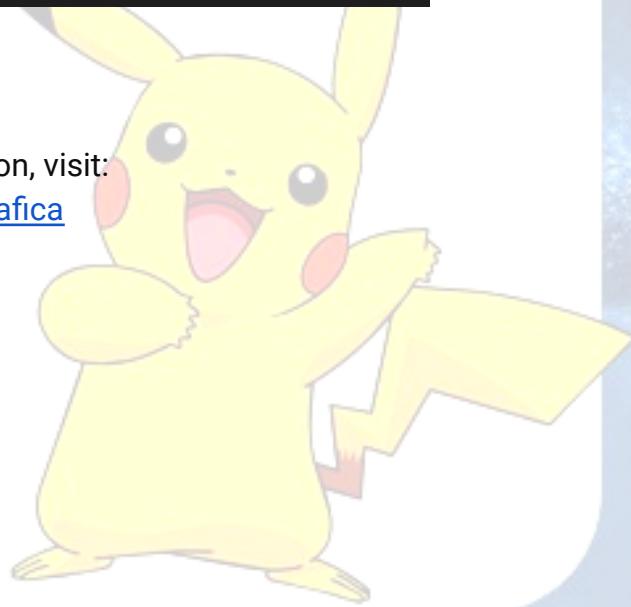
829 void animacion1() //circuito de la pokebola
830 {
831     ...
832     if (circuito1)
833     {
834         if (reco1)
835         {
836             rotPik = 360;
837             movPikX += 0.1f;
838             if (movPikX > 16.5)
839             {
840                 reco1 = false;
841                 reco2 = true;
842             }
843         }
844         else if (reco2)
845         {
846             rotPik = 90;
847             movPikZ -= 0.1f;
848             if (movPikZ < -9.0)
849             {
850                 reco2 = false;
851                 reco3 = true;
852             }
853         }
854         else if (reco3)
855         {
856             rotPik = 270;
857             movPikY += 0.1f;
858             if (movPikY > 1.75)
859             {
860                 reco3 = false;
861             }
862         }
863     }
864 }

```

End of the code.

For further understanding of the code and visualization, visit:

<https://github.com/ZayraSheccid/ProyectoCompuGrafica>



## Conclusion

- Cruz Vazquez Zayra Sheccid

The purpose of this project was to demonstrate everything learned throughout the semester, although it was supposed to take less time since we started working on the models halfway through the semester. I realized that this was not the case.

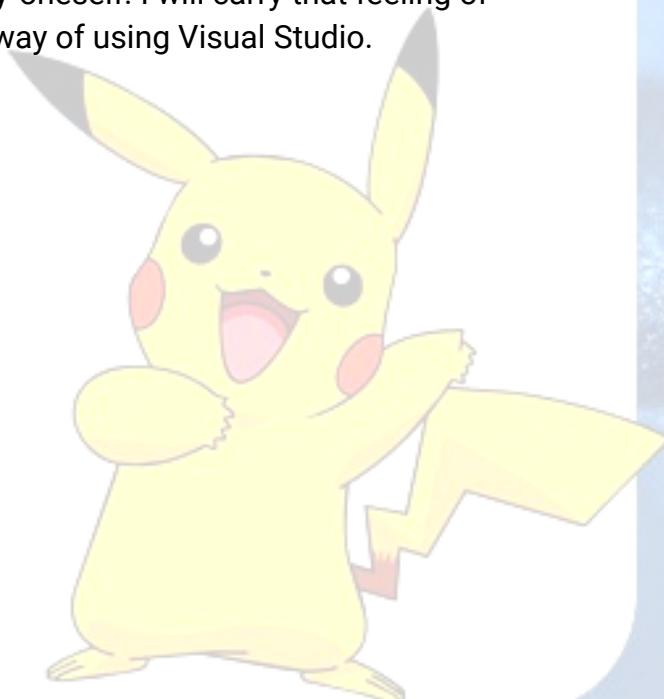
Despite choosing the models, creating the facade was challenging, and in the end, it didn't turn out as I would have liked.

The most challenging part of completing the project was the animation since my figures were more like furnishings than characters or things that wouldn't look strange when animated. Simple animations were the easiest, as it involved modifying the rotate or translate in the main loop, adjusting the DoMovement, and KeyCallback.

As for complex animations, the difficulty lay in the analysis to make them work, which took me more than a day to figure out and implement correctly. In the end, it turned out as I wanted, so I am satisfied in that regard.

The most challenging aspect of this project was the abandonment by my teammates, one a few days before the deadline, removing the elements he had contributed, and the other just an hour before delivering the project. As a result, I was only able to complete the minimum requirements in the end.

In conclusion, I understood that, at the end of the day, computer graphics is a very beautiful and interesting subject. Although it consumes a lot of time, it is very rewarding to see models or animations created by oneself. I will carry that feeling of knowing how to use new software and a different way of using Visual Studio.



## References

- Models used
  - Nguyen, B. (30 de enero de 2019). *Giradiscos de pedestal de ciencia ficción* modelo 3d. turbosquid.  
<https://www.turbosquid.com/es/3d-models/3d-scifi-pedestal-model-1373706>
  - artis 3D. (26 de junio de 2021). *Colección de cuadros de Zabel's y Bond Girl Giclee 2* modelo 3d. turbosquid.  
<https://www.turbosquid.com/es/3d-models/3d-painting-1683109>
  - Roombaroom. (16 de septiembre de 2013). *Pedestal* modelo 3d. turbosquid.  
<https://www.turbosquid.com/es/3d-models/free-max-model-architectural-modules/767833>
  - Elikchan. (14 de junio de 2022). *Mimikyu* modelo 3d. turbosquid.  
<https://www.turbosquid.com/es/3d-models/mimikyu-model-1840036>
  - VinnyHaw. (14 de marzo de 2010). *Pokeball Original* modelo 3d. turbosquid.  
<https://www.turbosquid.com/es/3d-models/free-max-model-pokeball-pokemon/521682>
  - 5th Dimension. (16 de octubre de 2022). *Conjunto de sofás 01* modelo 3d. turbosquid.  
<https://www.turbosquid.com/es/3d-models/3d-model-sofa-set-01-1973705>
  - *modelo 3d GenTree 103 Árbol genérico 103 gratis - TurboSquid 2062798.* (2023, April 26). Turbosquid.com.  
<https://www.turbosquid.com/es/3d-models/gentree-103-generic-tree-103-3d-model-2062798>
  - *modelo 3d Lámpara colgante Wooden LED cuadrada o redonda 1 gratis - TurboSquid 1990127.* (2022, November 20). Turbosquid.com.  
<https://www.turbosquid.com/es/3d-models/pendant-light-wooden-led-square-or-round-model-1990127>
  - *modelo 3d Pérgola de jardín de madera gratis - TurboSquid 1893894.* (2022, May 6). Turbosquid.com.  
<https://www.turbosquid.com/es/3d-models/wooden-garden-pergola-3d-model-1893894>
  - *modelo 3d Sansevieria gratis - TurboSquid 2071525.* (2023, May 18). Turbosquid.com.  
<https://www.turbosquid.com/es/3d-models/3d-sansevieria-2071525>

- *3DS - Pokémon X / Y - #042 Golbat - the models resource.* (n.d.).  
Models-resource.com. Retrieved November 23, 2023, from  
<https://www.models-resource.com/3ds/pokemonxy/model/8671/>
- *3DS - Pokémon X / Y - #074 Geodude - the models resource.* (n.d.).  
Models-resource.com. Retrieved November 23, 2023, from  
<https://www.models-resource.com/3ds/pokemonxy/model/11368/>
- *Nintendo 64 - Pokémon Stadium 2 - #013 Weedle - the models resource.* (n.d.). Models-resource.com. Retrieved November 23, 2023, from  
[https://www.models-resource.com/nintendo\\_64/pokemonstadium2/model/34027/](https://www.models-resource.com/nintendo_64/pokemonstadium2/model/34027/)
- *Nintendo 64 - Pokémon Stadium (JPN) - #039 - Jigglypuff - the models resource.* (n.d.). Models-resource.com. Retrieved November 23, 2023, from  
[https://www.models-resource.com/nintendo\\_64/pokemonstadiumjpn/model/24478/](https://www.models-resource.com/nintendo_64/pokemonstadiumjpn/model/24478/)
- *Nintendo Switch - Pokémon Scarlet / Violet - #0004 Charmander - the models resource.* (n.d.). Models-resource.com. Retrieved November 23, 2023, from  
[https://www.models-resource.com/nintendo\\_switch/pokemonscarletviolet/model/58692/](https://www.models-resource.com/nintendo_switch/pokemonscarletviolet/model/58692/)
- *Nintendo Switch - Pokémon Scarlet / Violet - #0642 Thundurus - the models resource.* (n.d.). Models-resource.com. Retrieved November 23, 2023, from

[https://www.models-resource.com/nintendo\\_switch/pokemonscarleyletvioret/model/58948/](https://www.models-resource.com/nintendo_switch/pokemonscarleyletvioret/model/58948/)

- *Wii - My Pokémon Ranch - #049 - Venomoth - the models resource.*  
(n.d.). Models-resource.com. Retrieved November 23, 2023, from  
<https://www.models-resource.com/wii/mypokemonranch/model/22560/>
- *Wii - Pokémon Battle Revolution - #104 Cubone - the models resource.*  
(n.d.). Models-resource.com. Retrieved November 23, 2023, from  
<https://www.models-resource.com/wii/pokemonbattlerevolution/model/56642/>

