



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



COMPUTACIÓN GRÁFICA E INTERACCIÓN
HUMANO-COMPUTADORA

PROYECTO FINAL - MANUAL TÉCNICO

Equipo 10

NOMBRE:

Cruz Vázquez Zayra Sheccid

No. Cuenta: 316159126

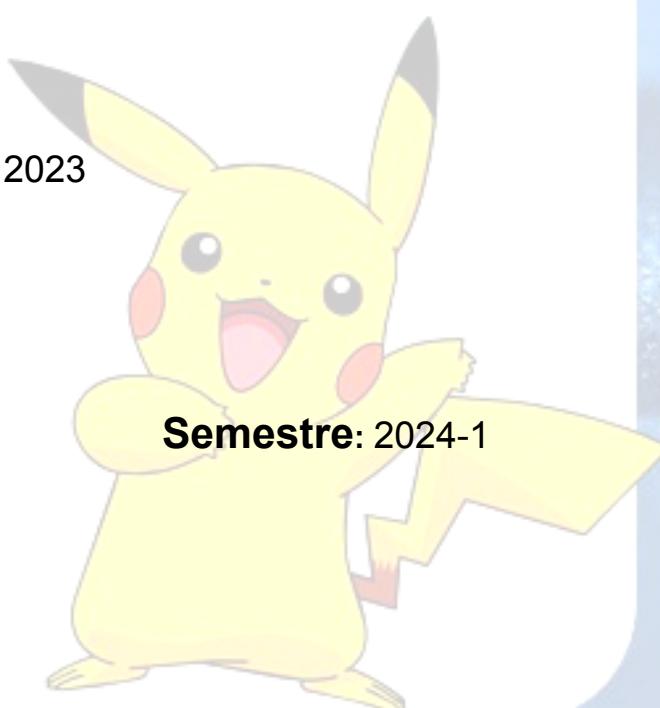
PROFESOR:

Arturo Pérez De la Cruz

Entrega: Miércoles, 22 de Noviembre del 2023

Grupo: 1

Semestre: 2024-1



Descripción

Este proyecto tiene como objetivo principal llevar a los fanáticos de Pokémon y a cualquier persona a una experiencia única: una visita digital interactiva a un museo dedicado a la icónica franquicia Pokémon. En este proyecto, nos centraremos en la creación y el modelado en 3D de figuras Pokémon para dar vida a la rica historia y el mundo de estos adorables personajes que tuvo mucho significado en la infancia de incontables personas.

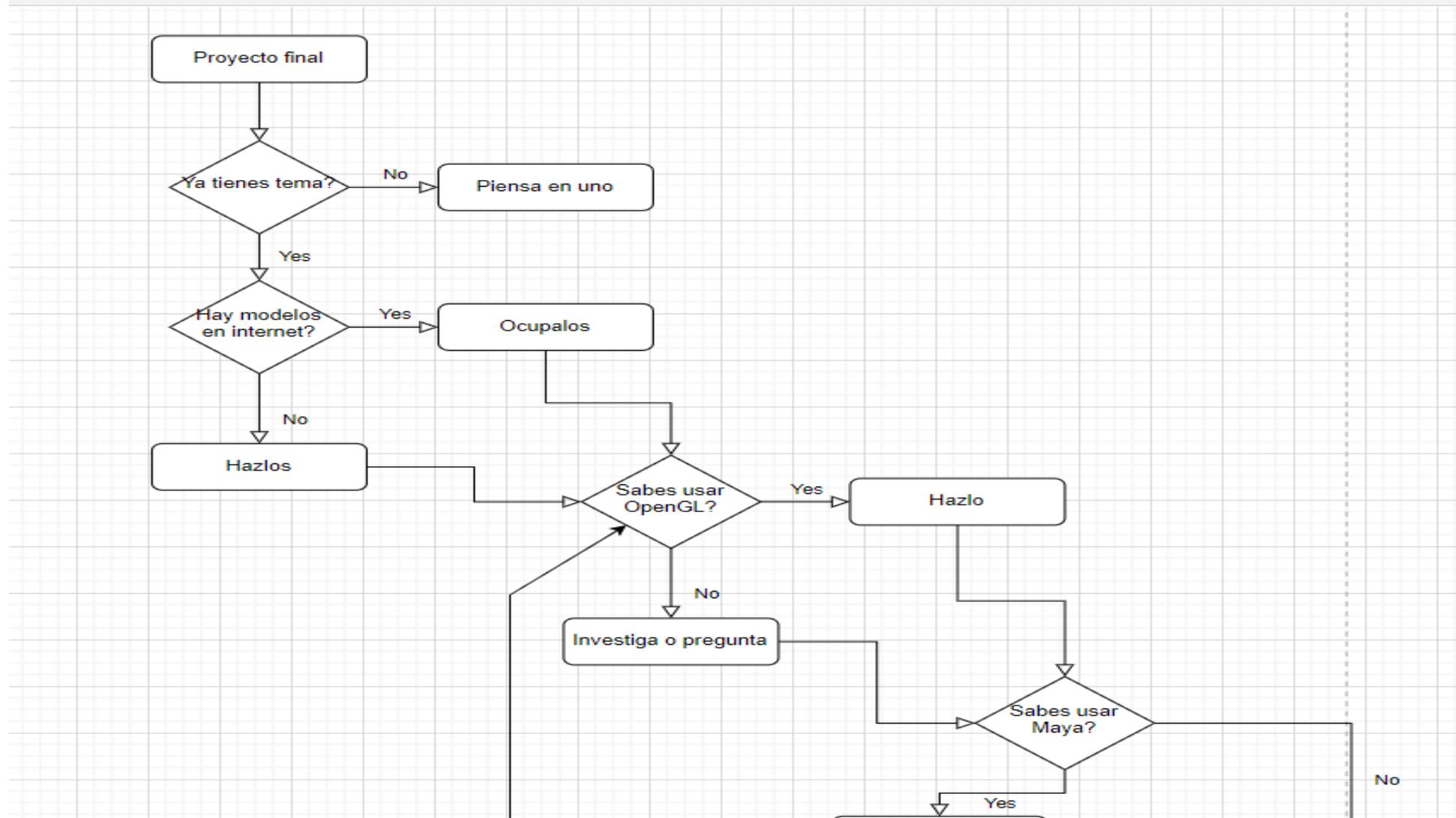
Además que servirá como ayuda a los integrantes del equipo para desarrollar, reforzar y aplicar los conocimientos adquiridos tanto en teoría como en laboratorio de la materia de Computación Gráfica, utilizando aplicaciones como Visual Studio y la aplicación de modelado Maya o similares.

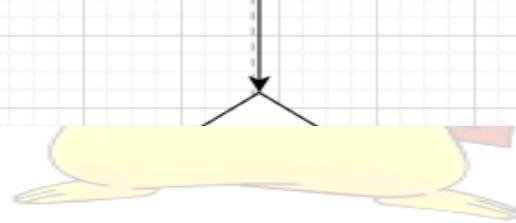
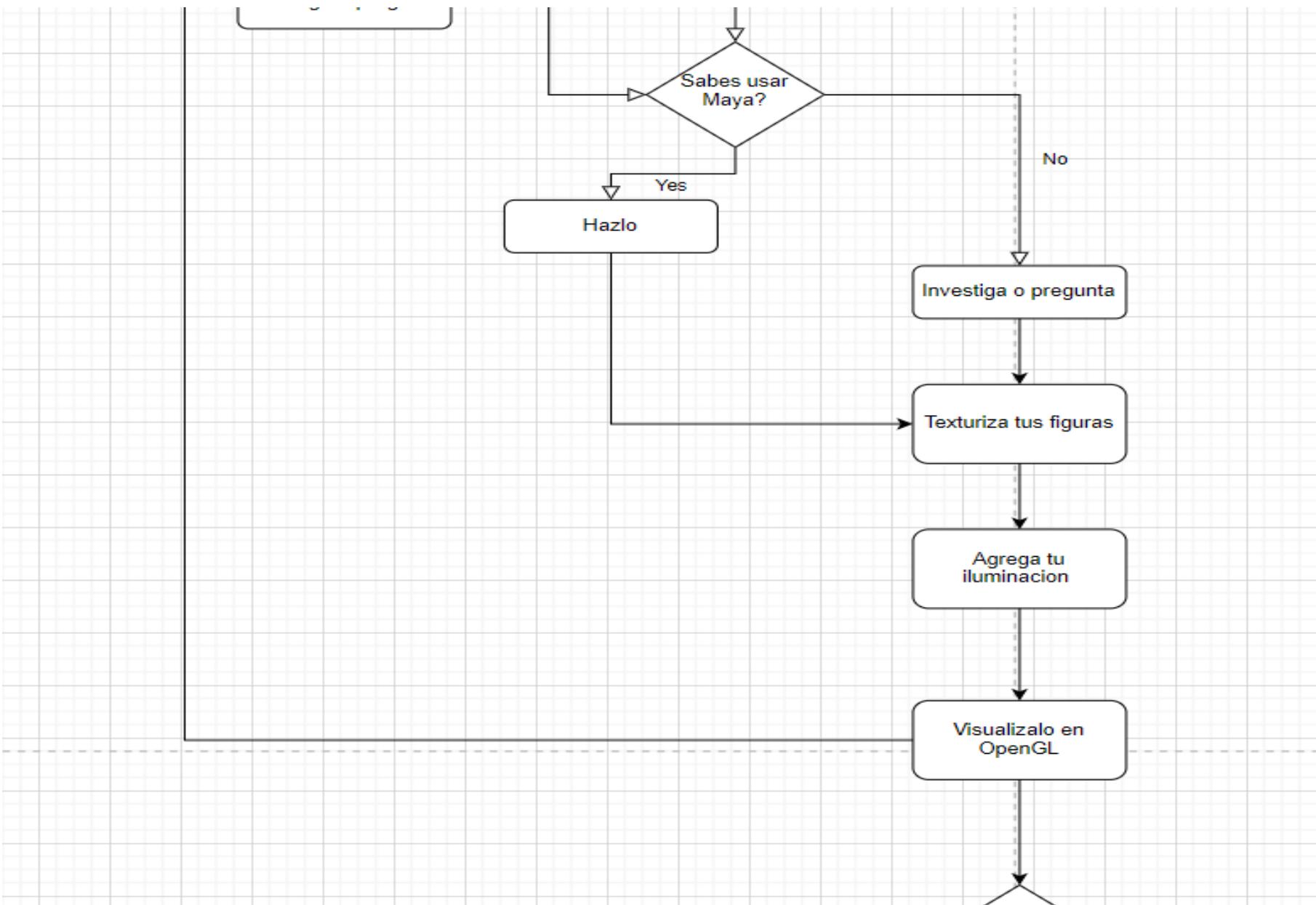
Objetivos:

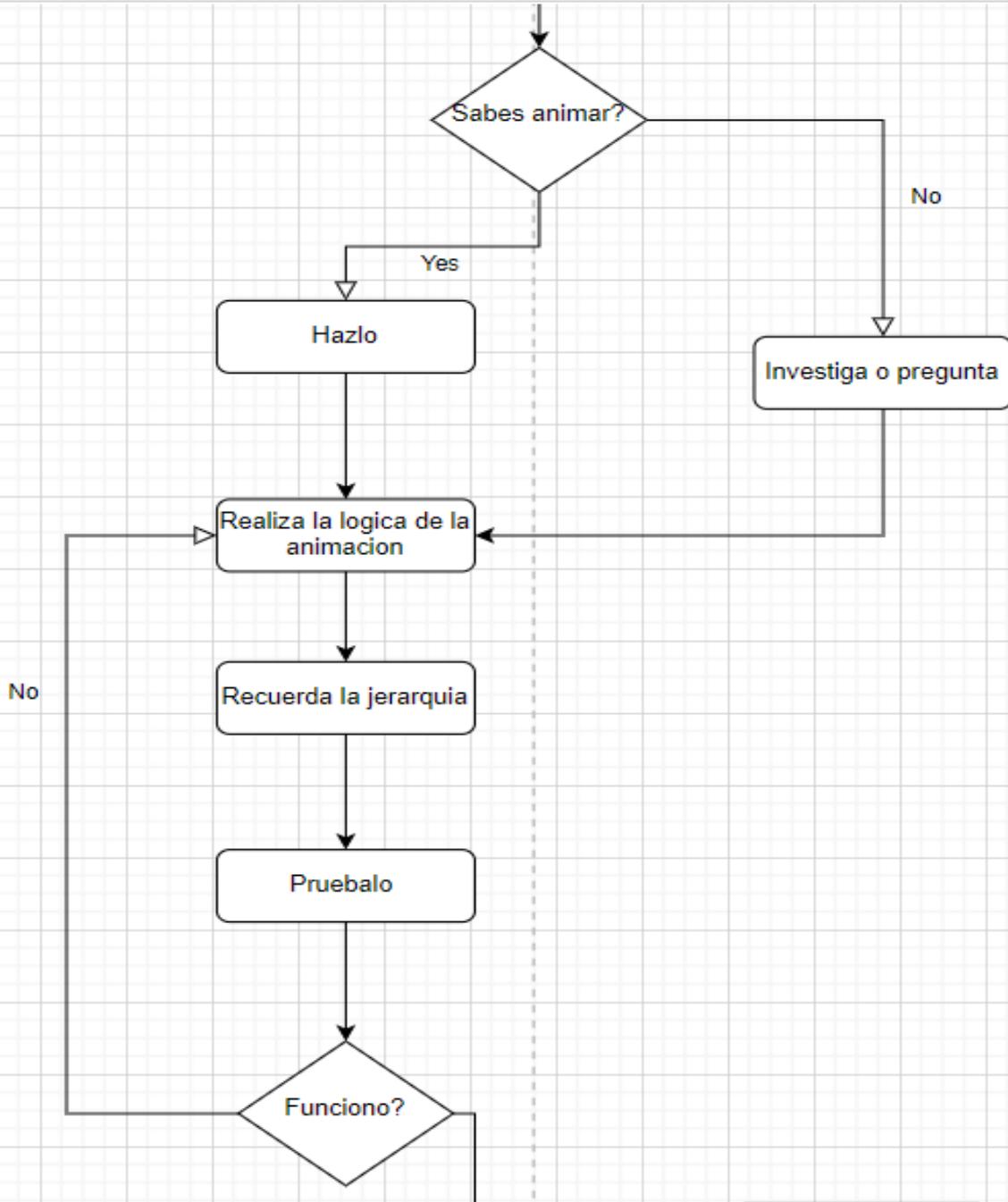
1. Realizar una Visita Digital al Museo Pokémon
2. Realización de modelado 3D en OpenGL
3. Diseño de Contenido en Autodesk Maya
4. Integración de Contenido de animación
5. Documentación detallada sobre el proyecto además de un Manual de Usuario

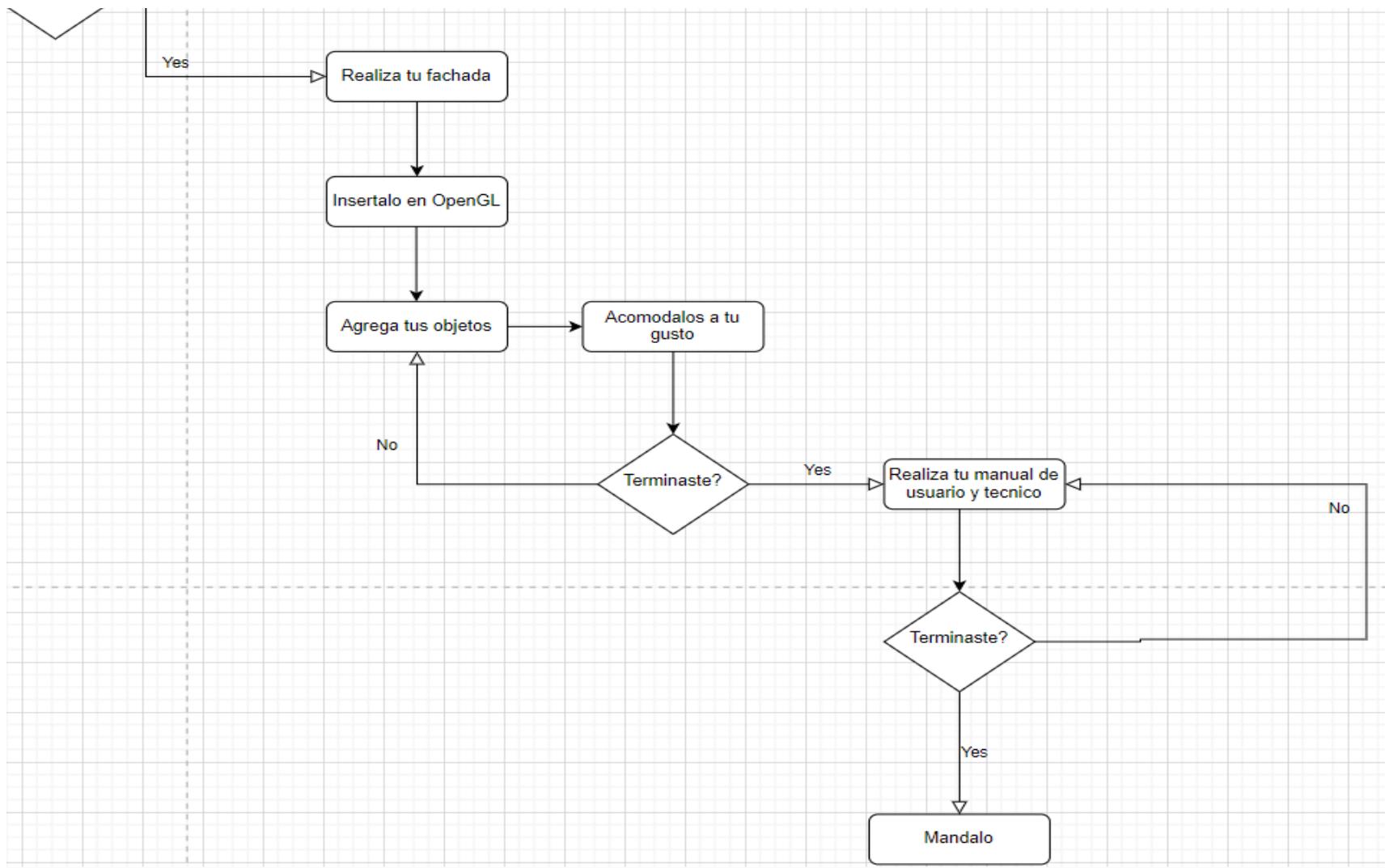


Diagrama de flujo del software









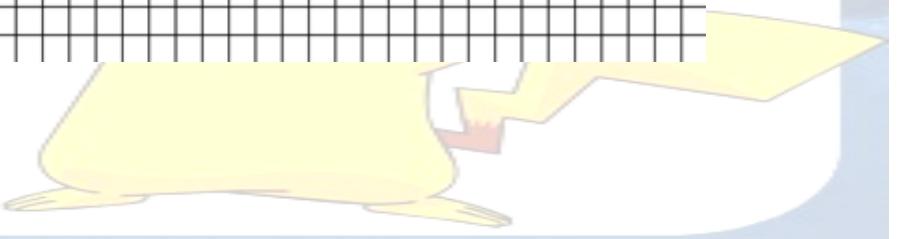
Para observarlo mejor visite:

https://drive.google.com/file/d/1x2DHGOBh9tpGSBkifsT_IrC-t2XOyoID/view?usp=sharing



Diagrama de Gantt

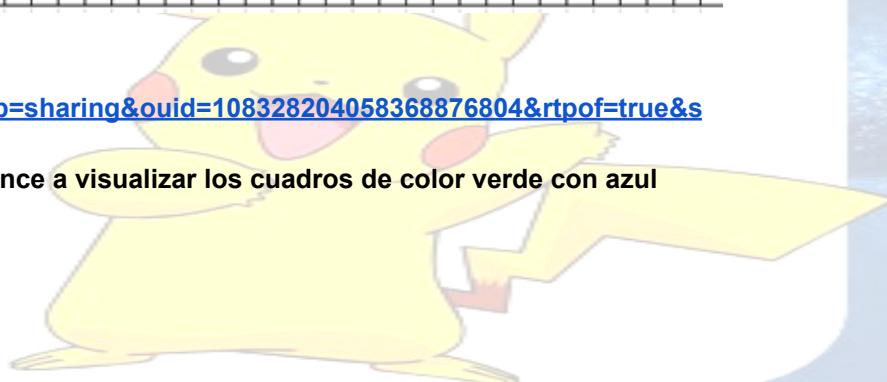
Proyecto final de Computación Gráfica e Interacción Humano-Computadora					
Integrantes:					
Abad Vásquez Aldo Caudillo Avilés Esteban Tonatiuh Cruz Vazquez Zayra Sheccid					
Nombre de la tarea	Fecha de inicio	Fecha de entrega	Asignado	Estado	
Inicio del proyecto y fin de proyecto	06/09/2023	22/11/2023	Todos	En Progreso	06/09/2023 07/09/2023 08/09/2023 09/09/2023 10/09/2023 11/09/2023 12/09/2023 13/09/2023 14/09/2023 15/09/2023 16/09/2023 17/09/2023 18/09/2023 19/09/2023 20/09/2023 21/09/2023 22/09/2023 23/09/2023 24/09/2023 25/09/2023 26/09/2023 27/09/2023 28/09/2023 29/09/2023 30/09/2023 01/10/2023 02/10/2023 03/10/2023 04/10/2023 05/10/2023 06/10/2023 07/10/2023 08/10/2023 09/10/2023 10/10/2023
1.- Entregar una lista con los nombres de los integrantes	06/09/2023	10/09/2023	Todos	Terminado	
2.- Elección de tema	08/09/2023	24/09/2023	Aldo	Terminado	
3.- Realización del documento de la propuesta del proyecto (borrador)	08/09/2023	08/10/2023	Todos	Terminado	
3.1.- Realizar el croquis del Museo	08/09/2023	06/10/2023	Zayra	Terminado	
3.2.- Análisis financiero	08/09/2023	10/09/2023	Esteban	Terminado	
3.3.- Diagrama de Gantt	01/10/2023	07/10/2023	Esteban	Terminado	
3.4.- Animaciones	06/10/2023	08/10/2023	Aldo	Terminado	
3.5.- Descripción	01/10/2023	02/10/2023	Zayra	Terminado	
3.6.- Objetivos	01/10/2023	02/10/2023	Zayra	Terminado	
3.7.- Entregables	01/10/2023	02/10/2023	Zayra	Terminado	
3.8.- Requisitos generales del proyecto	06/10/2023	08/10/2023	Esteban	Terminado	
3.9.- Requisitos del cliente	06/10/2023	08/10/2023	Esteban	Terminado	
4.- Elección de modelos	06/10/2023	08/10/2023	Aldo	Terminado	
5.- Planeación de animaciones	06/10/2023	08/10/2023	Aldo	Terminado	



Para observarlo mejor visite:

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



Alcance del proyecto

El objetivo de este proyecto es recrear un espacio de la conocida serie Pokemón, teniendo en cuenta algunos pokemones dando la impresión de un museo, además de un lugar recreativo para ellos.

- Diseño

El diseño de la fachada se basó en el videojuego PokemonGO con un estilo conservador a lo que se conoce

La recepción está diseñada para visitantes como pokemones, en donde se ve algunas cosas de historia como jeroglíficos o la anatomía de un pokémon, con una parte trasera en donde los pokemones podrán jugar y convivir.

- Referencias

En el museo se pueden encontrar figuras representativas a la serie entre ellos: Pokebola, Mimikyu y Mew

- Ambientación

Se tienen algunas cosas adicionales a los objetos elegidos, esto por ejemplo la agregación de Mew, lámpara, macetas, arboles, mesa exterior.



Limitantes

Mis limitantes al realizar este proyecto a lo largo del semestre fueron los siguientes:

- Falta de tiempo

Como se sabe esta materia consume mucho tiempo ya que se está modelando, por lo que al meter 7 materias se me dificultó en algún momento dedicarle tiempo a esto, y si lo hacía le tenía que dedicar mínimo 2 horas al día desde el viernes que se dejaba la tarea al jueves que era cuando se entregaban los reportes y prácticas

- Laptop lenta

Al igual que lo anterior tener una computadora sin una capacidad necesaria para operar Maya o VisualStudio fue difícil ya que se tardaba demasiado en abrir las aplicaciones o en la compilación del código.

- Problemas de salud

En las últimas dos semanas antes de la entrega del proyecto fui operada por lo que estuve algún tiempo en el hospital, cosa que me hizo perderme de la última clase en donde se iban a resolver dudas, cosa que yo tenía y muchas. Esta situación además de las terapias que consumen tiempo a lo largo del día fue difícil.

- Falta de creatividad

Al final hubo un momento donde me atoré ya que no se me ocurrió que podía animar o como animarlo, esto al final fue lo que menos me limitó ya que tuve ayuda de mentes demasiado ingeniosas y creativas que me decían que animación podrían realizar mis objetos.



Metodología de software

- Planeación

Al inicio del curso se nos comentó que se debería de hacer un proyecto para esta materia, los detalles en su momento no fueron especificados.

Después de un mes ambos profesores tanto en teoría como en laboratorio mandaron los requerimientos para el proyecto, el de teoría quería un museo de algo real o ficticio con libertad de decidir tema, el de laboratorio nos dio la opción de trabajar en el mismo proyecto de teoría pero, obviamente sin entregar lo mismo. Por lo que después de formar los equipos se determinó:

- Nuestro tema sería de un museo de pokémon
- Cada integrante realizará una sala del museo diferente, para después unirlo con ayuda de la plataforma GitHub
- Cada uno realizaría sus animaciones
- Cada semana se tendrían que estar viendo los avances de esto

- Análisis

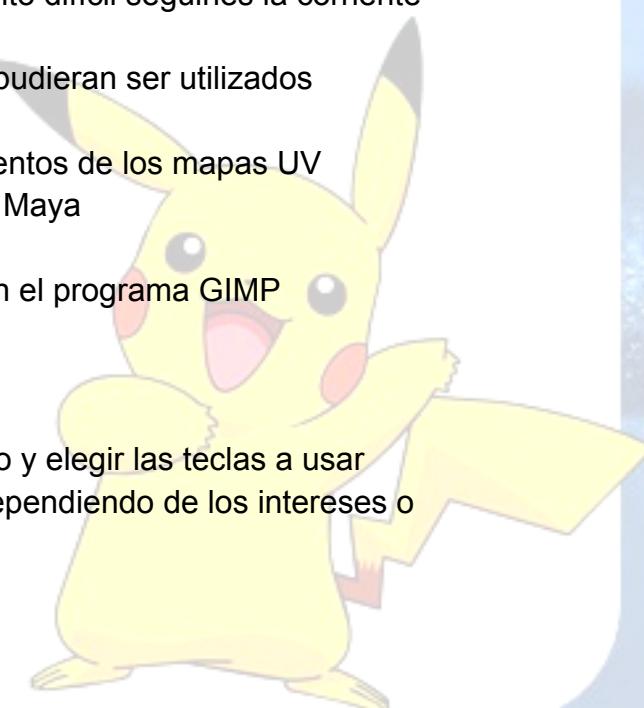
Mi falta de experiencia en la materia me hizo averiguar formas de ser útil para mis compañeros por lo que se pensó:

- Ser autodidacta
- Aprender con ayuda de videos o personas afines
- Poner atención en las clases de laboratorio
- Preguntar mis dudas

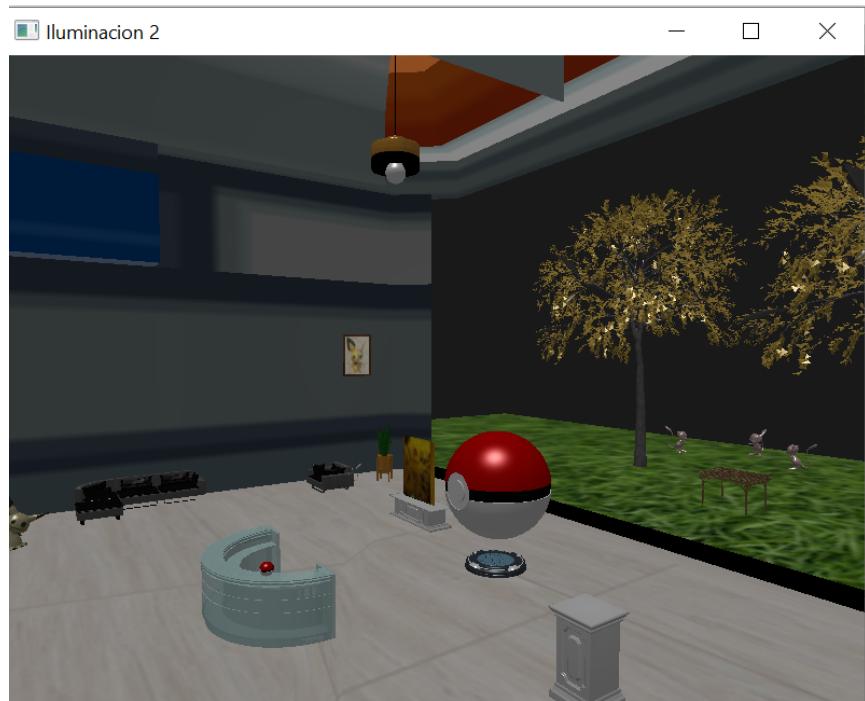
- Desarrollo

Al contrario de mis integrantes de equipo yo era nueva en entender el programa Maya y utilizar OpenGL por lo cual al inicio me resultó difícil seguirles la corriente por lo que lo que se planeó

1. Buscar modelos en páginas de internet que pudieran ser utilizados
2. Aprender a utilizar el programa Maya
3. Saber aplicar texturas y aplicar los conocimientos de los mapas UV
4. Utilizar las herramientas proporcionadas por Maya
(separate,reduce,combine,etc)
5. Aprender a bajar la calidad de las texturas en el programa GIMP
6. Utilizar el pivote como forma de animar
7. Saber utilizar los shaders
8. Colocar iluminación
9. Realizar el análisis de lo que estaría animado y elegir las teclas a usar
10. Hacer pruebas y eliminar o agregar cosas dependiendo de los intereses o perspectiva



- Resultados



Documentación del código

Se agregan las bibliotecas y se cambiar el x64 a 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"
```

Se agregan los prototipos de función y se configura la cámara

```

// 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); //position

```

Se crean las variables que se ocuparan a lo largo del proyecto

```

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

```

Se posicionan los point light en donde los queramos

```

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

```

Se crean los modelos que serán utilizados

```

204     Shader lightingShader("Shaders/lighting.vs", "Shaders/lighting.frag");
205     Shader lampShader("Shaders/lamp.vs", "Shaders/lamp.frag");
206
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/fachada1.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

```

Empieza el bucle principal, es importante llamar a todos los prototipos que serán utilizados, en mi caso fue lo que ya se tenia y se agrego animacion y animacion1

```

// 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);
}

```

Se especifican los point light y la spot light



```

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);
glUniform1f(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

```

Se mandan a llamar a los modelos para dibujarlos y presentarlos en OpenGL

```

//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 translucidos
 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);

//sofa1

```



```

//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);

```

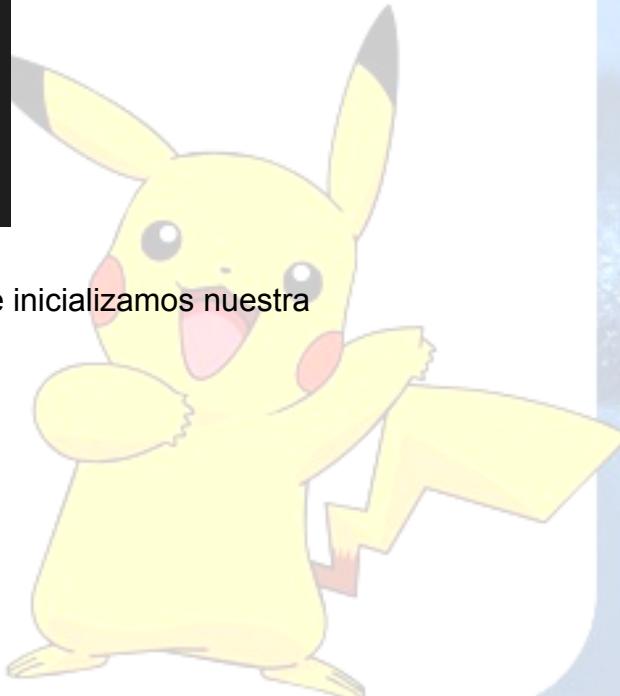
En DoMovement se pondrán los rangos o teclas que harán que nuestros objetos se muevan

```

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

```

En KeyCallback se ponen las teclas que preferimos e inicializamos nuestra animación



```

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

```

Procesamos el movimiento del ratón de la maquina utilizada

```

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);
}

```

Creamos nuestros voids para nuestras animaciones compleja



```

781 void animacion() //circuito de la pokebola
782 {
783     if (circuito)
784     {
785         if (recorrido1)
786         {
787             if (recorrido1)
788             {
789                 movKitY += 0.1f;
790                 movKitX += 0.1f;
791                 if (movKitY > 2)
792                 {
793                     recorrido1 = false;
794                     recorrido2 = true;
795                 }
796             }
797         }
798     }
799     else if (recorrido2)
800     {
801         movKitY -= 0.1f;
802         if (movKitY < -1.5)
803         {
804             recorrido2 = false;
805             recorrido3 = true;
806         }
807     }
808     else if (recorrido3)
809     {
810         movKitY += 0.1f;
811         movKitX += 0.1f;
812         if (movKitY > 0.5)
813         {
814             recorrido3 = false;
815             recorrido4 = true;
816         }
817     }
818 }
819

```



```

829 void animacion() //circuito de la pokebola
830 {
831     if (circuito1)
832     {
833         if (reco1)
834         {
835             rotPik = 360;
836             movPikX += 0.1f;
837             if (movPikX > 16.5) {
838                 reco1 = false;
839                 reco2 = true;
840             }
841         }
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
855     else if (reco3)
856     {
857         rotPik = 270;
858         movPikY += 0.1f;
859         if (movPikY > 1.75)
860         {
861             reco3 = false;
862         }
863     }
864 }
865

```

Fin del código

Para mayor conocimiento del código y visualización visite:
<https://github.com/ZayraSheccid/ProyectoCompuGrafica>



Conclusiones

- Cruz Vazquez Zayra Sheccid

Este proyecto tuvo como propósito demostrar todo lo aprendido a lo largo del semestre, aunque se supone que debería de haber tomado menos tiempo ya que se empezó a trabajar desde la mitad del semestre con los modelados, me di cuenta de que esto no fue así.

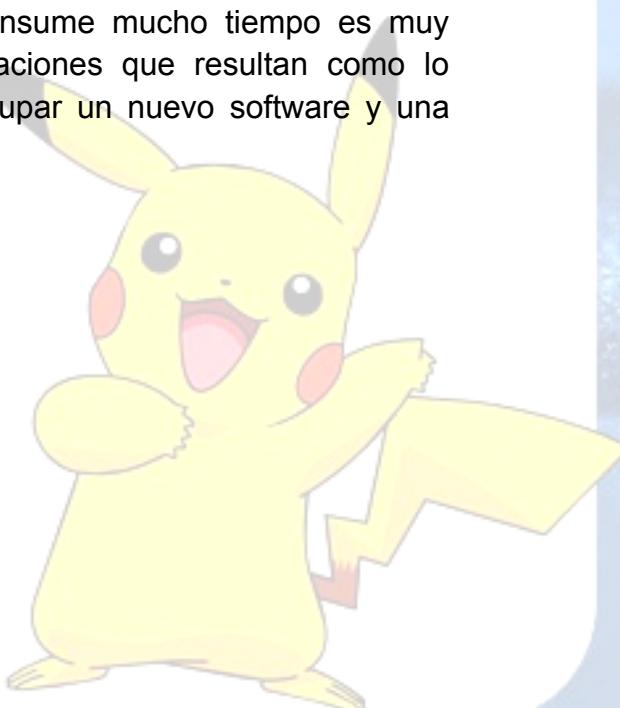
Ya que aunque los modelados que se escogieron,,la creación de la fachada fue complicada, cosa que al final no quedó como hubiera querido.

Lo que más se me dificulto al momento de terminar el proyecto, fue la animación ya que mis figuras eran más de amueblado que de personajes o cosas que no se verían raro al momento de animar, las animaciones sencillas fue lo más fácil, ya que como lo aprendido en clase solo era modificar el rotate o translate en el bucle principal, modificar el DoMovement y KeyCall.

Con respecto a las animaciones complejas lo difícil fue el análisis para que esta saliera, cosa que me llevó más de un día en descubrir y emplear que estuviera bien, al final resultó como quería así que en ese aspecto estoy satisfecha.

Lo más difícil en este proyecto fue el abandono de mis compañeros, uno unos días antes de la entrega y que nos quitara los elementos hechos por él, y por el otro una hora antes de entregar el proyecto, por lo que al final pude realizar lo mínimo.

En conclusión pude entender que al final del día la computación gráfica es una materia muy bonita e interesante que aunque consume mucho tiempo es muy gratificante ver modelados hechos por ti o animaciones que resultan como lo imaginaste, me llevaré esa sensación de saber ocupar un nuevo software y una forma diferente de ocupar Visual Studio.



Bibliografía

- Modelos empleados:
 - 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 redonda1 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>

