



UNIVERSIDAD NACIONAL AUTÓNOMA MÉXICO

FACULTAD DE INGENIERÍA

COMPUTACION GRAFICA E INTERACCION
HUMANO– COMPUTADORA

PROFESOR: ARTURO PÉREZ DE LA CRUZ

PROYECTO FINAL

MANUAL TECNICO

INTEGRANTES:

NAVARRO OSORIO ARMANDO

ROMERO LARES LUIS DANIEL

PERIODO: 2022-1

CIUDAD DE MEXICO, DICIEMBRE DE 2021

Objetivo:

La finalidad de la creación de este manual es poder brindar al lector cómo es el funcionamiento del código y así mismo las diferentes metodologías que se implementaron para desarrollar del proyecto.

Descripción:

Se deberá realizar un recorrido virtual del interior de un espacio habitacional donde se incluirán elementos generados por cada alumno en específico.

Además de una ambientación tridimensional para simular lo mejor posible el ambiente virtual.

El ambiente virtual será con temática navideña.

Se deberá realizar el ambiente virtual lo más completo posible contemplando principalmente el interior, aunque deberá contemplar el exterior por lo menos de una fachada principal, colocando diversos elementos.

Desarrollo:

Para el desarrollo del proyecto se implantaron un total de siete semanas en las cuales se dividieron de la mejor manera para poder tener los mejores resultados de este proyecto, para tener un mejor control se implementó el uso de diferentes herramientas con el caso de GitHub, Monday, Maya, Blender, Visual Studio y Zoom para realizar unas reuniones y para tener una mejor comunicación entre los desarrolladores.

Diagrama de Gantt

		Días	Octubre/24 - 30 /2021	Octubre 31 - Noviembre6 /2021	Noviembre/7 - 13 /2021	Noviembre/14 - 20 /2022	Noviembre/21 - 27 /2023	Noviembre 28 - Diciembre4 /2021	Diciembre /5 - 11 /2022
# Actividad	Actividad	Descripción(Semana)	1	2	3	4	5	6	7
1	Planeación	Se elige la temática del proyecto y las herramientas a utilizar	Lun						
2	Propuesta	Se entrega una propuesta de las actividades para cumplir con los requerimientos del proyecto	Sab						
3	Aprender Git y Github	El equipo aprende a utilizar un manejador de versiones para llevar un orden en el proyecto		Dom Sab					
4	Creación de Repositorio	Se crea el repositorio que contendrá el proyecto y se agregan a los colaboradores		Lun					
5	Modelado	Se modelan los elementos que estarán presentes en el proyecto		Lun Sab	Dom Sab	Dom Sab	Dom Mier		
6	Texturizado	Se texturizan los modelos y se les agrega iluminación			Dom Sab	Dom Sab	Dom Sab		
7	Documentación	A la par del modelado y texturizado se inicia la documentación de las actividades realizadas			Dom Sab	Dom Sab	Dom Sab	Dom Sab	Dom Mier
8	Carga de escenario	Se cargan los elementos estáticos del entorno				Lun			
9	Carga de modelos	Se cargan los elementos que posteriormente se les dará animación				Mier			
10	Carga de audio	Se agregan las pistas de audio que se sincronizarán con las animaciones					Lun		
11	Animación	Se implementan las animaciones requeridas para el proyecto					Dom Sab	Dom Sab	Dom Sab
12	Pruebas	Probamos que funcione nuestro proyecto					Mier	Dom Sab	
13	Resolución de problemas	En caso de que tengamos problemas al juntar, trabajaremos en solucionarlos						Dom Mier	Dom Mier
14	Entrega de proyecto	Fin del proyecto							Mier

Como se observa en el Diagrama de Gantt se consideró un total de siete semanas en las cuales se trabajo en el proyecto 5 días para cada desarrollador dedicando un promedio de 3 a 4 horas por días.

Cabe mencionar que algunas actividades tomaron mas tiempo de lo establecido como fue el caso del curso y de la implantación de GitHub que establecimos un total de días, pero al final del proyecto visualizamos un total de cinco días, ya que no se encontraba un curso rápido y fácil para poder comprender Git y Github.

Con respecto al modelado del ambiente fue un poco más difícil de implementar, ya que no se encontraban modelos con las características que habíamos planeado en la propuesta del proyecto. Por lo que se tomaron cursos de manejo de Git y Github, Trello y un curso de modelado para crear los objetos del entorno. Estos cursos fueron de gran ayuda en el desarrollo del proyecto tanto en la organización como en el desarrollo.



RECURSOS Y SOFTWARE IMPLEMENTADOS

Los recursos que se implementaron fueron los siguientes:

- Blender



- Maya



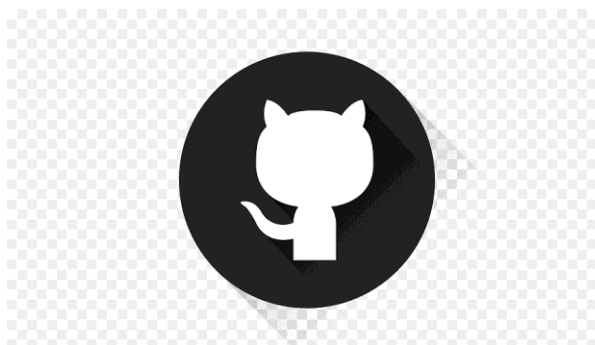
- GIMP



- Visual Studio 2019

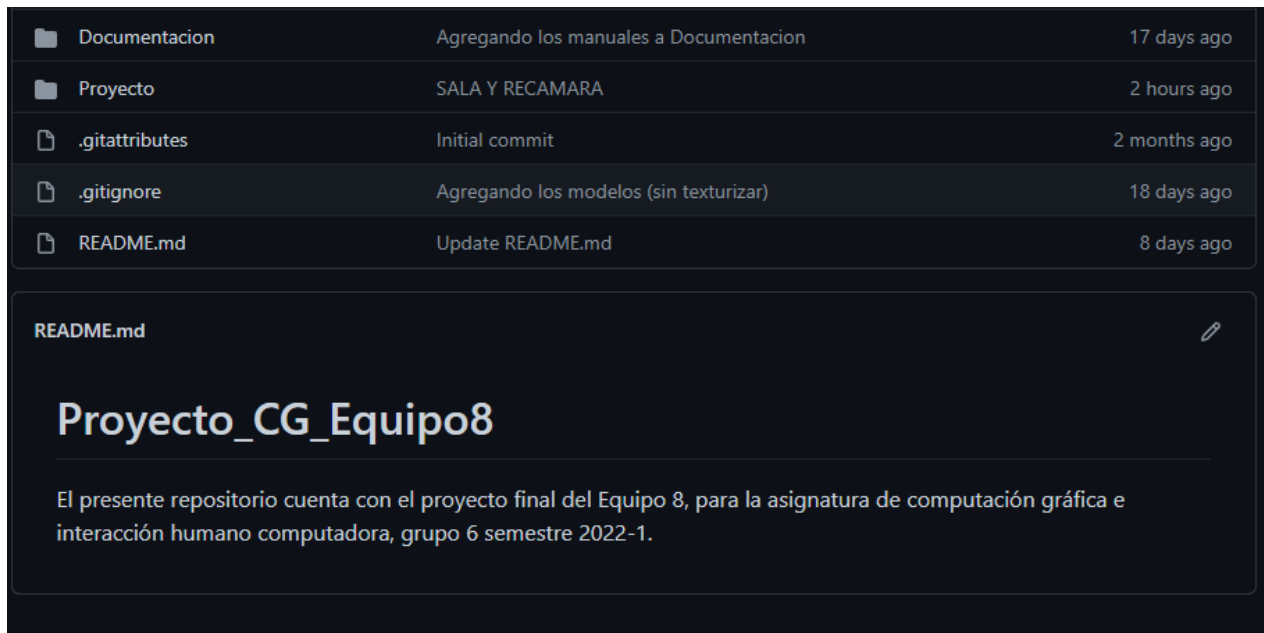


- GitHub



Creación del repositorio

La implementación de un repositorio ayudo a que el trabajo fuera más fácil para el equipo de trabajo ya que con esto nos brinda la posibilidad de tener un mejor manejo de versiones y en caso de tener algún error o alguna complicación se poda regresar a versiones anteriores sin problema alguno.



En el repositorio se pueden visualizar las carpetas de todo el proyecto, la documentación y los videos del proyecto.

Bibliotecas del proyecto:

Se utilizó Visual Studio 2019 el cual es un entorno de desarrollo integrado para Windows, Linux y macOS. Es compatible con múltiples lenguajes de programación, tales como C++ (que es el que se usará en este proyecto).

Windows.h: Contiene todas las funciones, estructuras, macros y constantes numéricas que forman la API.

Glad/glad.h: Es una biblioteca que carga punteros a las funciones de OpenGL en tiempo de ejecución, el núcleo y las extensiones.

GLFW: es una biblioteca de utilidad ligera para uso con OpenGL. Proporciona a los programadores la capacidad de crear y dirigir ventanas y aplicaciones OpenGL, así como recibir la entrada de joystick, teclado y ratón.

Stdlib.h: Contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras.

GLM: es una librería matemática escrita en C++ para el desarrollo de software Gráfico basado en OpenGL.

Time.h: Relacionado con formato de hora y fecha es un archivo de cabecera de Es la biblioteca estándar del lenguaje de programación C que contiene funciones para manipular y formatear la fecha y hora del sistema.

SDL: es una biblioteca multiplataforma (Linux, Windows, ...) para el control multimedia del ordenador
iostream: Es un componente de la biblioteca estándar del lenguaje de programación C++ que es utilizado para operaciones de entrada/salida.

camera.h: Se encargará del funcionamiento adecuado de la cámara.

Model.h: Se encargará de la funcionalidad que se requiere para cargar los modelos 3D.

Texture.h: Se encargará de la carga de texturas para su uso en el ambiente gráfico.

CODIGO VISUAL STUDIO

Esta parte de código establece cuáles son los archivos que se usan y los encabezados que se implementa para la carga de archivos y de librerías extras

```
Shader shader("Shaders/modelLoading.vs", "Shaders/modelLoading.frag");  
Shader lightingShader("Shaders/lighting.vs", "Shaders/lighting.frag");  
Shader lampShader("Shaders/lamp.vs", "Shaders/lamp.frag");  
Shader SkyBoxshader("Shaders/SkyBox.vs", "Shaders/SkyBox.frag");
```

Carga de modelos para el escenario

Casa y el patio del entorno

```
Model Casa((char*)"Models/Casa/casa3.obj");  
Model Piso((char*)"Models/Piso/piso.obj");
```

Tren

```
Model tren2((char*)"Models/TREN/TREN2.obj");  
Model tren((char*)"Models/TREN/TREN.obj");  
Model tren3((char*)"Models/TREN/TREN3.obj");  
Model llanta1((char*)"Models/TREN/LLANTA1.obj");  
Model llanta2((char*)"Models/TREN/LLANTA1.obj");  
Model llanta3((char*)"Models/TREN/LLANTA2.obj");  
Model llanta4((char*)"Models/TREN/LLANTA2.obj");  
Model llanta5((char*)"Models/TREN/LLANTA2.obj");  
Model llanta6((char*)"Models/TREN/LLANTA2.obj");
```

Primer muñeco de nieve

```
Model nieve((char*)"Models/TREN/NIEVE.obj");  
Model nieve2((char*)"Models/TREN/NIEVE2.obj");
```

Segundo muñeco de nieve

```
Model muneco((char*)"Models/BLANCO/MUNECO.obj");  
Model muneco1((char*)"Models/BLANCO/MUNECO1.obj");  
Model brazo1((char*)"Models/BLANCO/BRAZO.obj");  
Model gorro((char*)"Models/BLANCO/GORRO.obj");  
Model brazo2((char*)"Models/BLANCO/BRAZO2.obj");
```

Sala de la casa

```
Model sala((char*)"Models/SALA/SALA.obj");
```


Recamara de la casa

```
Model recamara_piso((char*)"Models/RECAMARA/PISORECAMARA.obj");
Model recamara1((char*)"Models/RECAMARA/RECAMARA1.obj");
Model recamara2((char*)"Models/RECAMARA/RECAMARA2.obj");
Model recamara3((char*)"Models/RECAMARA/RECAMARA3.obj");
Model recamara4((char*)"Models/RECAMARA/RECAMARA4.obj");
Model recamara5((char*)"Models/RECAMARA/RECAMARA5.obj");
```

Modelo de los seis ángeles

```
Model angel((char*)"Models/ANGEL/ANGEL.obj");
Model ala1((char*)"Models/ANGEL/ALA1.obj");
Model ala2((char*)"Models/ANGEL/ALA2.obj");
Model angel_brazo((char*)"Models/ANGEL/ANGEL_BRAZO.obj");

Model angel1((char*)"Models/ANGEL/ANGEL.obj");
Model ala11((char*)"Models/ANGEL/ALA1.obj");
Model ala21((char*)"Models/ANGEL/ALA2.obj");
Model angel_brazo1((char*)"Models/ANGEL/ANGEL_BRAZO.obj");

Model angel2((char*)"Models/ANGEL/ANGEL.obj");
Model ala12((char*)"Models/ANGEL/ALA1.obj");
Model ala22((char*)"Models/ANGEL/ALA2.obj");
Model angel_brazo2((char*)"Models/ANGEL/ANGEL_BRAZO.obj");

Model angel3((char*)"Models/ANGEL/ANGEL.obj");
Model ala13((char*)"Models/ANGEL/ALA1.obj");
Model ala23((char*)"Models/ANGEL/ALA2.obj");
Model angel_brazo23((char*)"Models/ANGEL/ANGEL_BRAZO.obj");

Model angel4((char*)"Models/ANGEL/ANGEL.obj");
Model ala14((char*)"Models/ANGEL/ALA1.obj");
Model ala24((char*)"Models/ANGEL/ALA2.obj");
Model angel_brazo24((char*)"Models/ANGEL/ANGEL_BRAZO.obj");

Model angel5((char*)"Models/ANGEL/ANGEL.obj");
Model ala15((char*)"Models/ANGEL/ALA1.obj");
Model ala25((char*)"Models/ANGEL/ALA2.obj");
Model angel_brazo25((char*)"Models/ANGEL/ANGEL_BRAZO.obj");
```

Regalos del árbol

```
Model Regalo1((char*)"Models/Regalo/regalo1.obj");
Model Regalo2((char*)"Models/Regalo/regalo2.obj");
```

Ventanas

```
Model MarcoVentana1((char*)"Models/Ventana/marcoVentana.obj");
Model Ventana1((char*)"Models/Ventana/ventana.obj");
Model MarcoVentana2((char*)"Models/Ventana/marcoVentana2.obj");
Model Ventana2((char*)"Models/Ventana/ventana2.obj");
```

Puertas de la casa

```
Model MarcoPuerta1((char*)"Models/Puerta/marcoPuerta1.obj");  
Model Puerta1((char*)"Models/Puerta/Puerta1.obj");  
Model MarcoPuerta2((char*)"Models/Puerta/marcoPuerta2.obj");  
Model Puerta2((char*)"Models/Puerta/Puerta2.obj");  
Model MarcoPuerta3((char*)"Models/Puerta/marcoPuerta3.obj");  
Model Puerta3((char*)"Models/Puerta/Puerta3.obj");
```

Parte exterior de la casa

```
Model Among((char*)"Models/Among/among.obj");  
Model Jardin((char*)"Models/Jardin/jardin.obj");  
Model Rosas((char*)"Models/Rose/rosas.obj");
```

Zona de la chimenea

```
Model Arbol((char*)"Models/Arbol/arbolNavideno.obj");  
Model Estrella((char*)"Models/Estrella/estrella.obj");  
Model Chimenea((char*)"Models/Chimenea/chimenea.obj");  
Model Adorno((char*)"Models/Chimenea/Venado/adornoVenado.obj");  
Model Sofa((char*)"Models/Sofa/sofa.obj");  
Model Calcetin1((char*)"Models/Calcetines/calcetin1.obj");  
Model Calcetin2((char*)"Models/Calcetines/calcetin2.obj");  
Model Mesa((char*)"Models/Mesa/mesa.obj");
```

Artículos sobre la mesa

```
Model Pollo((char*)"Models/Pollo/PolloFinal/pollo.obj");  
Model Plato((char*)"Models/Plato/plato.obj");  
Model Dulce((char*)"Models/Dulce/dulceNavideno.obj");  
Model Botella1((char*)"Models/Botellas/botella1.obj");  
Model Botella2((char*)"Models/Botellas/botella2.obj");  
Model Botella3((char*)"Models/Botellas/botella3.obj");  
Model Botella4((char*)"Models/Botellas/botella4.obj");  
Model Botella5((char*)"Models/Botellas/botella5.obj");  
Model Botella6((char*)"Models/Botellas/botella6.obj");  
Model Botella7((char*)"Models/Botellas/botella7.obj");  
Model Botella8((char*)"Models/Botellas/botella8.obj");
```

Variables Implementadas para las animaciones:

Variables del tren

```
bool anim = false;
float rot_tren = 0.0f;
bool recorrido1 = true;
bool recorrido2 = false;
bool recorrido3 = false;
bool recorrido4 = false;
float movx = 0.0f;
float movz = 0.0f;
float rotvehiculo = 0.0;
bool active_tren;
```

Variables del segundo muñeco de nieve

```
float rot_brazo = 0.0f;
float rot_brazo2 = 0.0f;
float rot_gorro = 0.0f;
bool active_brazo;
bool anim_brazo = false;
bool recorrido1_brazo = true;
bool recorrido2_brazo = false;
bool recorrido3_brazo = false;
bool recorrido4_brazo = false;
bool recorrido5_brazo = false;
bool recorrido6_brazo = false;
float mov_gorro = 0.0f;
```

Variables del primer muñeco de nieve

```
float movy = 0.0f;
bool arriba1 = true;
bool arriba_giro = false;
bool arriba2 = false;
bool abajo_giro = false;
float rot_nieve = 0.0f;
float rot_nieve2 = 0.0f;
float movy2 = 0.0f;

bool active_nieve;
bool anim_nieve = false;
```

Variable para la animación del ángel

```
float rot_ala1 = 0.0f;
float rot_ala2 = 0.0f;
float rot_brazo_angel = 0.0f;
bool active_angel;
bool anim_angel = false;
float piel_angel1 = 0.0f;
float piel_angel2 = 0.0f;

bool recorrido1_angel = true;
bool recorrido2_angel = false;
bool recorrido3_angel = true;
bool recorrido4_angel = false;
bool recorrido5_angel = true;
bool recorrido6_angel = false;
bool recorrido7_angel = true;
bool recorrido8_angel = false;
bool recorrido9_angel = true;
bool recorrido10_angel = false;
```

Carga del escenario

```
faces.push_back("SkyBox/right.tga");
faces.push_back("SkyBox/left.tga");
faces.push_back("SkyBox/top.tga");
faces.push_back("SkyBox/bottom.tga");
faces.push_back("SkyBox/back.tga");
faces.push_back("SkyBox/front.tga");
```

Traslaciones y rotaciones de los modelos

```
model = glm::mat4(1);
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
Casa.Draw(shader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.701f, 11.833f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.899f, 9.781f, 1.267f));
model = glm::rotate(model, glm::radians(rot_brazo_angel), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel_brazo.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.283f, 9.032f, -0.5f));
model = glm::rotate(model, glm::radians(rot_ala1), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ala1.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.283f, 9.032f, -0.5f));
model = glm::rotate(model, glm::radians(rot_ala2), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ala2.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.701f, 11.833f, -10.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel1.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.899f, 9.781f, -8.767f));
model = glm::rotate(model, glm::radians(rot_brazo_angel), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel_brazo1.Draw(lightningShader);
```

```
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.283f, 9.032f, -10.5f));
model = glm::rotate(model, glm::radians(rot_ala1), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ala11.Draw(lightingShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.283f, 9.032f, -10.5f));
model = glm::rotate(model, glm::radians(rot_ala2), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ala21.Draw(lightingShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.701f, 11.833f, -20.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel2.Draw(lightingShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.899f, 9.781f, -18.767f));
model = glm::rotate(model, glm::radians(rot_brazo_angel), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel_brazo2.Draw(lightingShader);
```

```
model = glm::mat4(1);
model = glm::translate(model, glm::vec3(9.283f, 9.032f, -20.5f));
model = glm::rotate(model, glm::radians(rot_ala1), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ala12.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(10.283f, 9.032f, -20.5f));
model = glm::rotate(model, glm::radians(rot_ala2), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ala22.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-65.701f, 11.833f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel3.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-65.4f, 9.781f, 1.267f));
model = glm::rotate(model, glm::radians(rot_brazo_angel), glm::vec3(1.0f, 0.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
angel_brazo23.Draw(lightningShader);

model = glm::mat4(1);
model = glm::translate(model, glm::vec3(-66.083f, 9.032f, -0.5f));
model = glm::rotate(model, glm::radians(rot_ala1), glm::vec3(0.0f, 1.0f, 0.0f));
glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(model));
ala13.Draw(lightningShader);
```

Animaciones

(La localización de las animaciones se encuentra en el manual de usuario)

1.- Animación del tren

```
if (anim)
{
    if (recorrido1)
    {
        if (movx < -130.0f)
        {
            recorrido1 = false;
            recorrido2 = true;
            rotvehiculo = -90.0f;
        }
        else
        {
            rot_tren += 0.3f;
            movx -= 0.1f;
        }
    }

    if (recorrido2)
    {
        if (movz < -150.0f)
        {
            recorrido2 = false;
            recorrido3 = true;
            rotvehiculo = -180.0f;
        }
        else
        {
            rot_tren += 0.3f;
            movz -= 0.1f;
        }
    }
}
```



```

if (recorrido3)
{
    if (movx > 130.0f)
    {
        recorrido3 = false;
        recorrido4 = true;
        rotvehiculo = 90.0f;
    }
    else
    {
        rot_tren += 0.3f;
        movx += 0.1f;
    }
}

if (recorrido4)
{
    if (movx > 150.0f)
    {
        recorrido4 = false;
        recorrido1 = true;
        rotvehiculo = 0.0f;
    }
    else
    {
        rot_tren += 0.3f;
        movx += 0.1f;
    }
}

```

Código en donde se muestra cómo se activad la animación:

```

if (keys[GLFW_KEY_V])
{
    active = !active;
    anim = !anim;
}

```

2.- Animación del primer muñeco de nieve

```
if (arriba1)
{
    if (movy > 8.0f && movy2 > 16.0f)
    {
        arriba1 = false;
        arriba_giro = true;
    }
    else
    {
        movy += 0.1f;
        movy2 += 0.2f;
    }
}
```

```
if (arriba_giro)
{
    if (rot_nieve < -360.0f && rot_nieve2 > 360.0f)
    {
        arriba_giro = false;
        arriba2 = true;
    }
    else
    {
        rot_nieve -= 1.0f;
        rot_nieve2 += 2.0f;
    }
}
```

```
if (arriba2)
{
    if (movy < 0.0f && movy2 < 0.0f)
    {
        abajo_giro = true;
        arriba2 = false;
    }
    else
    {
        movy -= 0.1f;
        movy2 -= 0.2f;
    }
}
```

```
if (abajo_giro)
{
    if (rot_nieve > 0.0f && rot_nieve2 < 0.0f)
    {
        abajo_giro = false;
        arriba1 = true;
    }
    else
    {
        rot_nieve += 1.0f;
        rot_nieve2 -= 1.0f;
    }
}
```

Tecla para llevar acabo la animacion

```
if (keys[GLFW_KEY_Z])
{
    active_nieve = !active_nieve;
    anim_nieve = !anim_nieve;
}
```

3.- Animación del segundo muñeco

```
if (recorrido1_brazo)
{
    if (rot_brazo > 52.0f) {
        //rot_gorro = -1.0f;
        recorrido1_brazo = false;
        recorrido2_brazo = true;
    }
    else
    {
        rot_brazo += 0.1f;
    }
}

if (recorrido2_brazo)
{
    if (mov_gorro < -1.0f) {
        //rot_gorro = -1.0f;
        recorrido2_brazo = false;
        recorrido3_brazo = true;
        recorrido4_brazo = true;
    }
    else
    {
        mov_gorro -= 0.01f;
    }
}
```

```
if (recorrido3_brazo)
{
    if (rot_brazo < 0.0f) {
        recorrido3_brazo = false;
        //recorrido4_brazo = true;
    }
    else
    {
        rot_brazo -= 0.1f;
    }
}
```

```

if (recorrido4_brazo)
{
    if (rot_brazo2 < -60.0f) {
        recorrido5_brazo = true;
        recorrido4_brazo = false;
    }
    else
    {
        rot_brazo2 -= 0.1f;
    }
}

```

```

if (recorrido5_brazo)
{
    if (mov_gorro > 0.0f) {
        recorrido5_brazo = false;
        recorrido6_brazo = true;
    }
    else
    {
        mov_gorro += 0.01;
    }
}

```

```

if (recorrido6_brazo)
{
    if (rot_brazo2 > 0.0f) {
        recorrido6_brazo = false;
        recorrido1_brazo = true;
    }
    else
    {
        rot_brazo2 += 0.1;
    }
}

```

Tecla para llevar acabo la animacion

```

if (keys[GLFW_KEY_X])
{
    active_brazo = !active_brazo;
    anim_brazo = !anim_brazo;
}

```

4.- Animación del ángel

```
if (recorrido1_angel)
{
    if (rot_ala1 > 30.0f)
    {
        recorrido1_angel = false;
        recorrido2_angel = true;
    }
    else
    {
        rot_ala1 += 0.1;
    }
}
```

```
if (recorrido2_angel)
{
    if (rot_ala1 < 0.0f)
    {
        recorrido2_angel = false;
        recorrido1_angel = true;
    }
    else
    {
        rot_ala1 -= 0.1;
    }
}
```

```
if (recorrido3_angel)
{
    if (rot_ala2 < -30.0f)
    {
        recorrido3_angel = false;
        recorrido4_angel = true;
    }
    else
    {
        rot_ala2 -= 0.1;
    }
}
```

```

if (recorrido4_angel)
{
    if (rot_ala2 > 0.0f)
    {
        recorrido4_angel = false;
        recorrido3_angel = true;
    }
    else
    {
        rot_ala2 += 0.1;
    }
}

```

```

if (recorrido5_angel)
{
    if (rot_brazo_angel < -15.0f)
    {
        recorrido5_angel = false;
        recorrido6_angel = true;
    }
    else
    {
        rot_brazo_angel -= 0.1;
    }
}

```

```

if (recorrido6_angel)
{
    if (rot_brazo_angel > 0.0f)
    {
        recorrido6_angel = false;
        recorrido5_angel = true;
    }
    else
    {
        rot_brazo_angel += 0.1;
    }
}

```

Tecla para la animacion

```

if (keys[GLFW_KEY_C])
{
    active_angel = !active_angel;
    anim_angel = !anim_angel;
}

```

Para el desarrollo de este proyecto se está tomando un periodo de trabajo de un mes aproximadamente tomando en cuenta un total de 16 horas de trabajo para cada desarrollador.

Tabla para tiempo normal					
Puesto	%Trabajo TN	Horas Trabajadas N	Sueldo N	Sueldo Mensual	Meses Trabajados
Desarrollador uno	10	16	313	5008	1
Desarrollador dos	10	16	313	5008	1

Tabla para tiempo acelerado					
Puesto	%Trabajo TA	Horas Trabajadas A	Sueldo A	Sueldo Mensual +15%	Meses Trabajados
Desarrollador uno	10	16	313	5759,2	1
Desarrollador dos	10	16	313	5759,2	1

En donde al igual ya se muestra y se agrega los costos de las herramientas a usar durante el desarrollo de este proyecto.

Tabla para tiempo normal			
Descripción	Costo Directo	Costo Indirecto	Costo Normal
Desarrollador uno	5008	250,4	5258,4
Desarrollador dos	5008	250,4	5258,4
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
Total Sueldos	10016	500,8	10516,8
Software	1509	75,45	1584,45
Equipo de computo	5000	250	5250
Material Extra	3400	170	3570
Total Costos	19925	996,25	20921,25

Esta es la tabla en donde ya se presenta una aceleración al proyecto.

Tabla para tiempo acelerado			
Descripción	Costo Directo	Costo Indirecto	Costo Acelerado
Desarrollador uno	5759,2	287,96	6047,16
Desarrollador dos	5759,2	287,96	6047,16
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
		0	0
Total Sueldos	11518,4	575,92	12094,32
Software	1509	75,45	1584,45
Equipo de Computo	5000	250	5250
Material Extra	3400	170	3570
Total Costos	21427,4	1071,37	22498,77

Requerimientos de Hardware:

Para el desarrollo de un proyecto de esta magnitud es necesario tener una computadora con buenas características como es el caso de un i5 de séptima generación o su equivalente en un ryzen, al igual una tarjeta de video de 2 vram y al menos un disco ssd para que los archivos se puedan cargar más fáciles.

Mencionamos estas características ya que investigando y el tiempo trabajo hemos notado que los componentes de nuestros equipos no son los mas adecuados por lo que optamos hacer una pequeña investigación den diversos foros y fue las mejores características para tener un proyecto más fluido.

Liga al repositorio:

[https://github.com/LuisRomeroLares/Proyecto CG_Equipo8](https://github.com/LuisRomeroLares/Proyecto	CG_Equipo8)