



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Materia:

Computación grafica e interacción humano-computadora

Alumno:

318285553

Profesor:

Carlos Aldair Roman Balbuena

Semestre:

2026-1

Grupo:

05

Entregable:

Proyecto Final (Manual Técnico)

Índice

Objetivo.....	2
Diagrama de Gantt	2
Diagrama de flujo	4
Alcance	5
Limitantes	5
Metodología.....	6
Herramientas y recursos utilizados:.....	7
Análisis de costos	8
Características	9
Animaciones implementadas	10
Documentación del código	11
Glosario de funciones	12
Glosario de variables.....	13
Estado del arte.....	16
Conclusión	17
Referencias	17
Resultados finales del proyecto.....	19

Objetivo

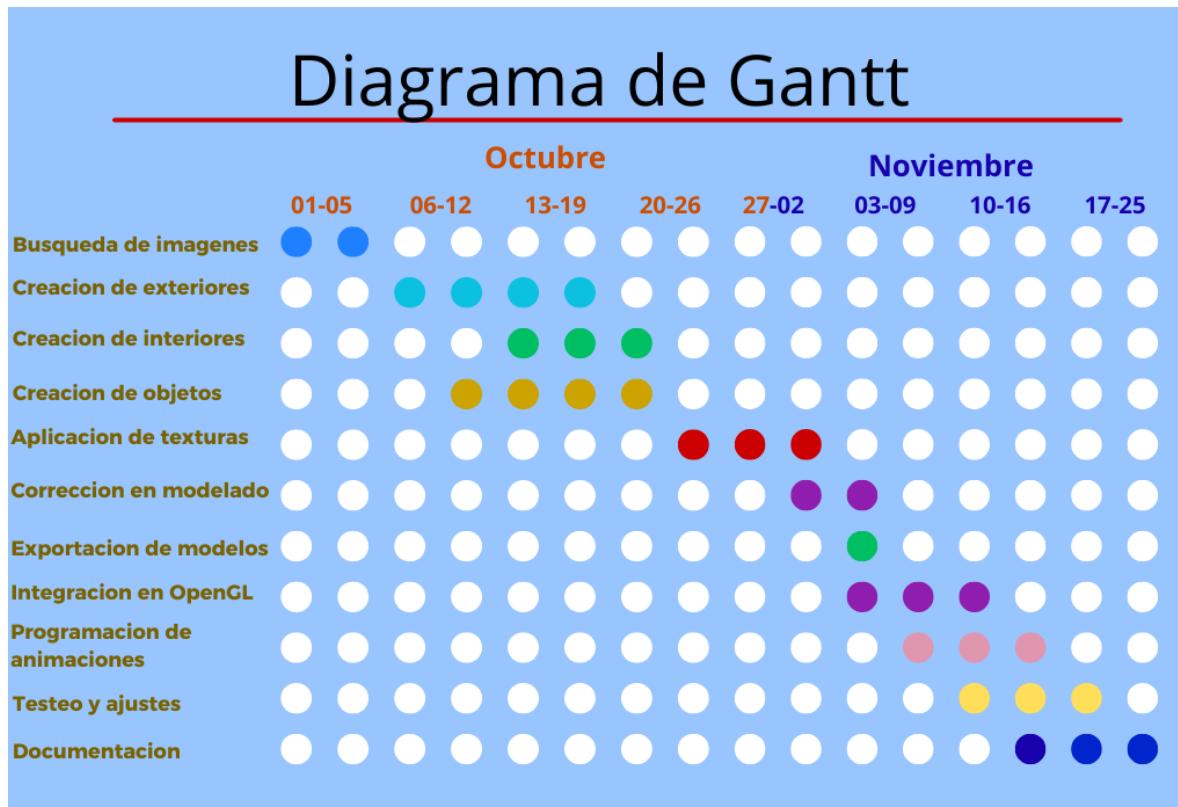
Recreación de un escenario: Modelar y renderizar con las herramientas proporcionadas la casa de la caricatura Billy and Mandy, incluyendo 2 cuartos (sala y baño) con 5 objetos en cada cuarto (toalla, taza, bañera, cortina, alfombra, televisión, sillón, lámpara, retrato, lámpara colgante), con sus respectivas texturas y mapas de normales.

Implementación de iluminación: Desarrollar un sistema de iluminación Phong que cumpla con los diferentes tipos de fuente de luz para dar realismo a la escena.

Desarrollo de animaciones: Programar comportamientos dinámicos en los objetos, animaciones de iluminación o movimiento.

Independencia del software: Generar un ejecutable final que sea funcional y portable, gestionando las dependencias de las librerías dinámicas para la ejecución en otro entorno.

Diagrama de Gantt



Este proyecto tuvo un plazo del 01 de octubre al 25 de noviembre de 2025 en el cual se desarrollaron diferentes tareas las cuales fueron:

Búsqueda de imágenes (01-05 de octubre): La fase inicial de este proyecto fue la búsqueda de modelos a realizar se eligió la caricatura de Billy y Mandy como inspiración para la ejecución de este proyecto. En esta búsqueda se obtuvieron las imágenes necesarias para modelar la fachada, sala, baño y objetos.

Creación de exteriores (13-23 de octubre): En este lapso se optó por utilizar blender como motor de modelado del proyecto, se realizaron diferentes pruebas para entender el funcionamiento de blender y se comenzó con el modelado exterior de la casa donde incluía: ventanas, puertas, chimenea y paredes.

Creación de interiores (10-23 de octubre): Continuando con el modelado en blender se modelaron los interiores de la casa, en donde se realizaron los marcos de las puertas, la separación por cuartos, los pisos y techos de cada uno de ellos.

Aplicación de texturas (23-02 de octubre): El proceso de texturizado se inició desde la finalización del modelado de los exteriores para facilitar la visualización del modelo, las texturas fueron diseñadas en Paint, Gemini y extraídas de internet.

Corrección en modelado (28 de octubre -05 de noviembre): Una vez finalizado el modelado de la casa, se revisaron fallos en los cuales existía duplicidad de vértices, error de ubicación o deformaciones en los objetos. Además, triangular cada una de las caras de los objetos correcciones de escalas y rotaciones.

Exportación de modelos (04-05 de noviembre): Los modelos se exportaron desde blender utilizando un formato .obj, con el objetivo de visualizarlo en OpenGL.

Integración en OpenGL (05 - 12 de noviembre): Los modelos realizados, se adaptaron para que se pudieran visualizar, de forma correcta en textura, ubicación y posición, además de los modelos obtenidos de internet.

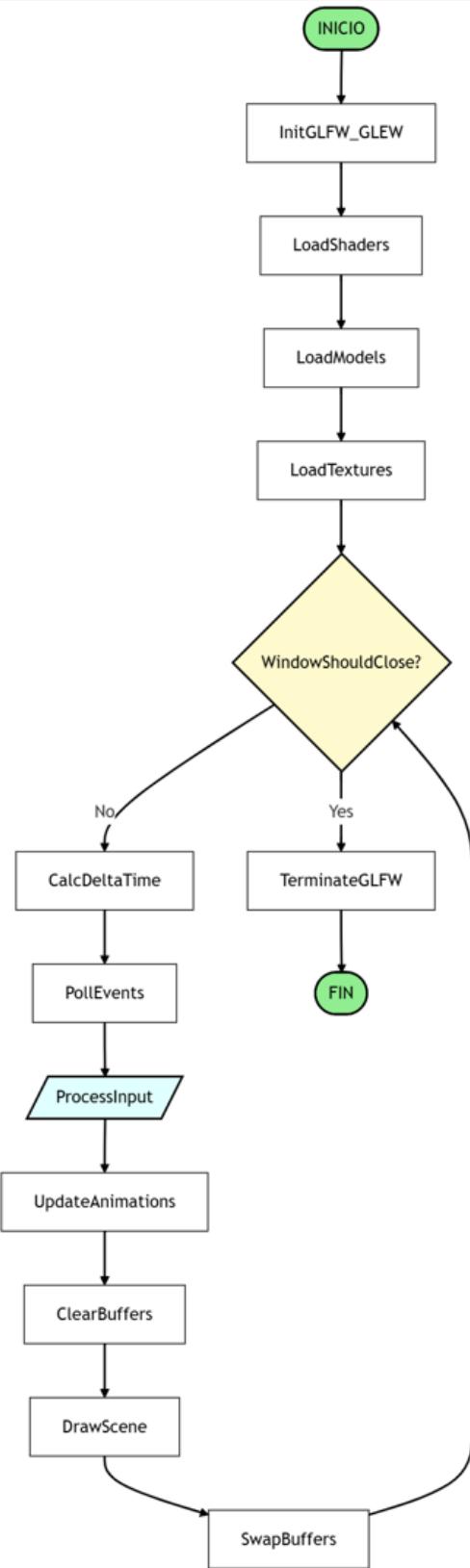
Programación de animaciones (07- 16 de noviembre): Las animaciones realizadas como el movimiento de lámpara encendido de la lámpara y apertura de puertas se realizaron con los objetos extraídos de blender.

Posteriormente para la animación de billy caminando y el espectro saliendo de la televisión se trajeron los modelos de internet y se cargaron al código con sus debidas texturas y correcciones de posición.

Testeo y ajustes (10-20 de noviembre): Se realizaron pruebas para comprobar que la posición de los objetos fuera correcta y tuviera sus debidos ajustes, además se corrigieron cada uno de los errores que tenían, como iluminación y texturas.

Documentación (10-25 de noviembre): Esta tarea se llevó a cabo una vez finalizado el proyecto en su parte práctica, para tener una visión de todo lo realizado.

Diagrama de flujo



Alcance

El proyecto se desarrolla en una aplicación visual grafica en tiempo real, enfocada en la recreación de un escenario específico la casa de Billy and Mandy y la implementación de técnicas en la computación grafica.

Este sistema permite que el usuario final pueda hacer distintas acciones dentro del entorno virtual:

Navegación libre: Exploración del escenario en primera persona mediante el movimiento de teclado y mouse con libertad de movimiento en los diferentes ejes.

Interacción del entorno: Capacidad de modificar el estado de objetos específicos mediante eventos del teclado.

Visualización de animaciones: Observación de comportamientos dinámicos, tanto cílicos (personajes caminando, lámpara oscilando, etc.

El software implementa un motor gráfico basado en OpenGL con las siguientes capacidades técnicas:

Carga de recursos: Importación de modelos 3D, imágenes y objetos animados, pipeline de iluminación Phong, soportando una luz direccional, múltiples luces puntuales con atenuación cuadrática, 1 luz tipo Spotlight acoplada a la cámara, Sistema de animación: Calculo de matrices de modelo en CPU para animaciones procedimentales (traslación, rotación, escala).

Animación a partir de un esqueleto, Implementación de Hardware Skinning en el Vertex Shader para la deformación de mallas basadas en huesos.

Limitantes

El sistema presenta algunas restricciones técnicas y funcionales como lo son:

Restricciones de plataforma y software: La aplicación esta compilada para entornos Windows sin embargo no es compatible con otros entornos como lo son macOS o Linux.

Física de los objetos: Los objetos carecen de un relleno que impida el paso a los usuarios o que permita verlos por dentro.

ASSIMP es sensible a otros formatos que no sean jpg, lo que puede generar errores de visión o falta de texturas.

La fachada que se decidió implementar no contaba con las medidas reales para la fabricación de la casa por lo que se tuvieron que hacer un cálculo estimado de las medidas y de cada uno de los objetos.

La falta de experiencia en herramientas como blender complican la creación de modelos de forma avanzada, los cuales puedan tener una mejor forma y escala de la forma correcta.

Metodología

La metodología principal del proyecto fue modelo en cascada, este enfoque lineal permitió avanzar ordenadamente a través de fases bien definidas, asegurando que cada etapa se completara para continuar con la siguiente:

Fase 1 Análisis de requisitos:

Requisitos funcionales: Se identificaron las necesidades de un entorno 3D navegable, interactivo y con animaciones simples y complejas.

Selección de temática: Se visualizó el alcance visualizado en la serie Billy and Mandy, determinando la lista de 10 objetos en 2 cuartos los cuales son: sala (sofa, televisión, lámpara, lámpara colgante y retrato) y baño (taza, bañera, cortina, toalla y alfombra) para modelar y cumplir con la rúbrica.

Fase 2: Diseño del sistema

Diseño de la escena: En blender se diseñó una fachada e interiores lo más realista posible, construyendo cada uno de los elementos que lo componen como las ventanas y puertas.

Diseño de clases: Se estructuró el programa separando la lógica en clases especializadas para cámara, shader, model y modelanim.

Diseño de la interacción: Se definieron los mapas de las teclas para la navegación por medio de (WASD) y la activación de eventos con diferentes teclas.

Fase 3: Implementación

Configuración de un entorno virtual, inicializando ventanas con GLFW y carga de extensiones con GLEW, un motor de renderizado, con la programación de los shaders GLSL para soportar el texturizado de la iluminación Phong (Luz direccional, focal y spotlight).

Lógica de animación: Implementación de algoritmos matemáticos para las transformaciones geométricas (matrices de modelo).

Fase 4: Pruebas

Validación visual: Se verificó que la iluminación fuera la correcta y que los objetos se visualizaran de la mejor forma posible.

Se realizaron pruebas de animación, donde se comprobó la fluidez de las transiciones y la correcta ejecución de la secuencia compleja, además de valorar la fluides de cada animación de acuerdo con el uso de deltaTime.

Fase 5: Despliegue y mantenimiento

La fase final se centró en la generación de un producto entregable, se generó un manual técnico y uno de usuario, donde se plasmará todo lo elaborado durante el proyecto.

Herramientas y recursos utilizados:

Visual Studio 2022: Este fue el entorno de desarrollo principal, donde se escribió y compiló todo el código del proyecto. Aquí se configuraron las librerías externas necesarias (GLFW, GLEW, Assimp) para que OpenGL funcionara correctamente. Además, sirvió para generar el ejecutable final y sus herramientas de depuración ayudaron a detectar errores en el código cuando el programa no corría.

Blender: Esta herramienta funcionó para la creación de cada uno de los modelos, incluyendo la construcción de la estructura de la casa, sus ventanas y cada una de las puertas. También se usó para modelar los objetos que se encontraban dentro, los cuales eran: sillón, televisión, recuadro, lámpara de pie, lámpara colgante, taza, bañera, toalla, alfombra y cortina. A cada uno de estos objetos se le aplicó su propia textura en Blender, corrigiendo los mapas UV y asignando los materiales. Además, se utilizaron herramientas específicas como el biselado, transformaciones básicas, extrusión de caras, cortar bucles, unir vértices por distancia, triangular caras y finalmente la exportación de los modelos en formato .obj.

Gemini y ChatGPT: El uso de estas herramientas cumplió la función de apoyo en varias áreas: ayudaron a generar ideas para algunas texturas, resolver dudas sobre la sintaxis de programación en C++ y la exportación correcta de archivos .obj. También sirvieron para explicar a detalle funciones específicas de Blender que no conocía y para optimizar la lógica de algunos objetos que ya estaban construidos.

Paint: Este programa ayudó a editar y crear nuevas texturas, así como a adaptar imágenes extraídas de internet o generadas por IA. Fue útil para guardar las imágenes en los formatos adecuados y asegurar que tuvieran las dimensiones correctas para que no hubiera problemas al momento de cargarlas en OpenGL.

Youtube: Se utilizó esta herramienta como plataforma en donde se logró subir la vídeo demostración de la casa una vez finalizada, además de una fuente de conocimiento en donde se pudo observar tutoriales los cuales ayudaban a entender el funcionamiento de blender así como cada una de las funciones con las que se podían mejorar los objetos del modelado.

Análisis de costos

Costos de producción

Campo	Monto (MXN)	Duración	Indicaciones	Subtotal	Evidencia	Gasto real
Sueldo Desarrollador Principal	\$12,308	2 meses	Salario de Programador Jr.	\$24,616.00	Referencia 10	Inversión de tiempo
Licenciamiento de Activos 3D (Billy y Fantasma)	\$5,500.00	Pago Único	Costo estimado por la compra de derechos comerciales de modeladores freelance para los personajes.	\$5,500.00	Referencia 1 y 2	No (activos usados con fines académicos)
Consultoría Técnica y Shaders	\$6,000.00	Pago Único	Pago por el motor gráfico (Clases ModelAnim, Shaders complejos) desarrollados por terceros (Senior Dev/Profesor).	\$6,000.00	Referencia 11	No
Equipo de Cómputo	\$16,499.00	2 meses	Desgaste del equipo de desarrollo (Depreciación a 3 años).	\$916.61.00	Referencia 12	Si
Servicios y licencias	\$1,000.00	2 meses	Electricidad, Internet y herramientas de IA (ChatGPT/Gemini) para texturizado.	\$2,000.00	Referencia 13, 14 y 15	Sí
Renta de cuarto	\$4,100.00	2 meses	Oficina donde trabajara el desarrollador	\$8,200.00	Referencia 16	No

Costo total de desarrollo: \$47,232.61

Utilizando un margen de utilidad del 30%, por impuestos, soporte técnico y ganancia, nos da un precio de venta estimado de:

\$61,402.39

El precio final de \$61,402.39 MXN se estableció considerando la complejidad técnica de este desarrollo. Al ser un proyecto elaborado directamente en C++ y OpenGL (en lugar de usar motores comerciales como Unity o Unreal que facilitan el trabajo), el valor aumenta por los siguientes motivos:

Rendimiento optimizado: Como el software no depende de un motor gráfico pesado, el ejecutable final es mucho más ligero y eficiente.

Trabajo de Ingeniería: Se está cobrando por la programación manual de componentes complejos, como los shaders personalizados y la lógica para las animaciones de huesos (esqueletos), lo cual requiere conocimientos especializados de programación y no solo de diseño.

Licencias y Derechos: Dentro de este precio ya se contempla el pago a los creadores originales de los modelos (Billy y el Espectro) y a los desarrolladores de las librerías.

Características

Este proyecto cuenta con funciones las cuales están desarrolladas en el lenguaje c++ y OpenGL, sus características principales son:

Recreación del escenario: Se recreó la casa de la caricatura Billy y Mandy, modelando tanto la fachada exterior como los interiores.

Objetos: El escenario incluye modelos 3D de autoría propia (creados en Blender) como la casa, arbustos, puertas, sillón, lámparas, televisión, retrato, taza, bañera, cortina, toalla y alfombra. También se integraron modelos externos adaptados, como el Espectro y el personaje de Billy.

Texturizado completo: Todos los modelos cuentan con sus propias texturas y mapas UV para darles color y detalle a materiales como madera, tela y vegetación.

Iluminación:

Modelo Phong: Se implementó un sistema de iluminación que permite que los objetos reaccionen a la luz y sombras en las caras.

Fuentes de luz: El escenario funciona con dos tipos de fuentes de luz simultáneos:

Luz direccional que simula el sol y luces puntuales focos de las lámparas que iluminan en todas direcciones y con un rango establecido.

Interacción y Cámara:

Navegación libre: El usuario puede moverse libremente por toda la casa y el jardín usando una cámara sintética en primera persona controlada por el mouse y las teclas WASD.

Control de entorno: Se programaron interacciones para que el usuario pueda abrir y cerrar las puertas de las habitaciones, así como encender y apagar la luz de la lámpara.

Animaciones:

Animación Esquelética (Skinning): Se implementó la carga de modelos animados por huesos. Utilizando el personaje de Billy, que realiza un ciclo de caminata en el jardín.

Animaciones por Código: Se crearon animaciones matemáticas para objetos estáticos, como el tambaleo de la lámpara y la rotación de las puertas al abrirse.

Eventos Especiales: Se programó una secuencia compleja en la televisión, donde al presionar una tecla cambia su textura por la de un portal y sale un modelo 3D (el fantasma) con movimiento.

Animaciones implementadas

1.-Animación de Puertas: Para el movimiento de las puertas no se usó una función compleja, sino una interpolación lineal básica, la lógica consiste en aumentar o disminuir el ángulo de rotación hasta llegar a un límite.

Lógica: Se utiliza una bandera para saber si abrir o cerrar.

Fórmula:

$$\text{Angulo} = \text{Angulo}_{actual} + (\text{Velocidad} \times \text{deltaTime})$$

Implementación: Se aplicaron tres transformaciones en orden específico para simular una bisagra:

Traslación al origen: Se mueve la puerta para que su eje de rotación el cual es la bisagra esté en el punto central.

Rotación: Se aplica la rotación en el eje Y usando el ángulo calculado.

Traslación de regreso: Se devuelve la puerta a su posición original en la casa.

2. Animación de la Lámpara: Para lograr que la lámpara se tambaleara de forma suave y continua se utilizó la función trigonométrica Seno.

Lógica: Al usar el tiempo de ejecución como variable, el valor oscila entre -1 y 1 suavemente.

$$\text{Angulo} = \text{Amplitud} + \text{seno}(\text{Tiempo} \times \text{Velocidad})$$

Esto genera un movimiento de vaivén entre -5 y 5 grados, creando el efecto de tambaleo en la lámpara.

3. Animación del Espectro: La aparición del fantasma saliendo de la TV es una animación procedural que combina dos movimientos simultáneos el primero es el avance lineal (eje Z): El fantasma sale de la pantalla hacia el frente.

$$\text{Pos}_z = \text{Pos}_{ini} + (\text{Tiempo} \times \text{Velocidad})$$

Oscilación lateral (eje X): Para que parezca que flota, se le agregó un movimiento de izquierda a derecha usando Seno.

$$\text{Pos}_x = \text{seno}(\text{Tiempo} \times 8.0) \times 0.3$$

4. Animación de Billy: Esta es la animación más compleja del proyecto ya que a diferencia de las anteriores que mueven todo el objeto, esta deforma la malla del personaje.

El modelo tiene un "esqueleto" invisible (huesos). Cada vértice del modelo de Billy está asignado a uno o más huesos.

Funcionamiento: La clase ModelAnim lee el archivo .dae y extrae los keyframes. Calcula cuánto tiempo ha pasado e interpola la posición de los huesos entre dos movimientos. Envía al Shader un arreglo de matrices. El Vertex Shader multiplica cada vértice por la matriz de su hueso correspondiente, logrando que Billy camine deformando sus piernas y brazos de forma natural.

Documentación del código

Este proyecto fue elaborado en el entorno OpenGL, este proyecto este compuesto por un archivo main donde inicializaremos cada uno de los componentes, en este apartado tenemos la lógica principal de las animaciones y de la interacción con el usuario.

Shader.h :Este ayuda a extraer la lectura, compilación y conexión de los archivos .frag y .vs.

Camera.h: Esta clase es fundamental ya que ayuda a crear una cámara sintética, con la cual podemos navegar en primera persona a través de los modelos realizados. Además, procesa la entrada del teclado y el mouse para generar una matriz de vista.

ModelAnim.h/meshAnim.h: Estas clases son un aporte del profesor Carlos Aldair Roman Balbuena, que gestionan la carga de modelos 3D animados como lo es la animación de Billy (movimiento.dae), esta clase se basa en la jerarquía de huesos y calculan su transformación en cada fotograma.

Shaders

Este proyecto está compuesto por distintos shaders los cuales cumplían con una función específica en los proyectos.

Lighting: Es un shader principal que cumple con la función de iluminar los objetos siguiendo el modelo de iluminación Phong y texturas en todos los tipos de fuentes de luz (direccional, focal y linterna).Ademas facilitan la carga de los modelos 3D estáticos como lo son los creados por autoría propia (casa, arbustos, puertas, sillón, lámparas, televisión, retrato, taza, bañera, cortina, toalla, alfombra) y los extraídos de diversas fuentes (Espectro y Billy). Estos utilizan bibliotecas Assimp para leer los archivos y asignar sus texturas.

modelLoading: Fue utilizado para la textura básica de la casa y las puertas.

anim: Aplica el funcionamiento de la piel del esqueleto del personaje animado multiplicando la posición del vértice por la posición de la matriz, donde realiza una suma ponderada de las transformaciones de los huesos que afectan el vértice.

lamp: Este shader dibuja un color sólido y genera una luz usada para el debug.

Glosario de funciones

Función	Tipo	Parámetros	Descripción
main	int	void	Es la función principal del programa. Aquí se inicializan las librerías, se cargan los modelos, texturas y shaders, se ejecuta el bucle principal (Game Loop) donde se dibuja todo.
KeyCallback	void	GLFWwindow* window, int key, int scancode, int action, int mode	Esta función ayuda a detectar cuando presionamos una tecla una sola vez. Sirve para activar los "interruptores" como abrir puertas, prender la luz o activar la animación de la TV.

MouseCallback	void	GLFWwindow* window, double xPos, double yPos	Registra el movimiento del mouse en la ventana. Sirve para calcular hacia dónde estamos mirando y actualizar la cámara en primera persona.
DoMovement	void	void	Esta función detecta si mantenemos presionada una tecla. Se usa para el movimiento continuo de la cámara (WASD) para que el desplazamiento sea fluido.
Animation	void	void	Aquí se encuentra la lógica matemática de las animaciones. Calcula los ángulos de apertura de las puertas, el tambaleo de la lámpara y el tiempo de la animación del fantasma.
ObjetoDraw	void	glm::mat4 base, glm::vec3 escala, glm::vec3 traslado, GLint uniformModel	Es una función que ayuda a dibujar cubos (Recibe la matriz base, el tamaño y la posición para dibujar objetos compuestos sin repetir tanto código).

Glosario de variables

Variables de sistema y ventana: Ayudan a controlar la configuración inicial de la ventana gráfica y el flujo de tiempo del renderizado.

Nombre	Tipo	Descripción
WIDTH	const GLuint	Ancho constante de la ventana (1600 px)
HEIGHT	const GLuint	Alto constante de la ventana (900 px)
SCREEN_WIDTH	int	Almacena el ancho real del framebuffer de la ventana
SCREEN_HEIGHT	int	Almacena el alto real del framebuffer de la ventana
deltaTime	GLfloat	Variable que mide el tiempo transcurrido entre el fotograma actual y el anterior
lastFrame	GLfloat	Guarda el tiempo del último fotograma procesado

Variables de entrada y cámara: Variables utilizadas para la manipulación de la imagen, así como cada una de las imágenes o movimientos.

Nombre	Tipo	Descripción
camera	Camera	Permite navegar por el escenario simulando una cámara en primera persona.
keys	bool	Arreglo que registra cuáles teclas se están presionando y cuáles no en cada momento.
lastX	GLfloat	Almacena la ubicación del cursor en el eje X del fotograma anterior para calcular el movimiento de la cámara.
lastY	GLfloat	Almacena la ubicación del cursor en el eje Y del fotograma anterior.
firstMouse	bool	Variable que evita saltos bruscos de la cámara al entrar la ventana por primera vez.
nKeyProcessed	bool	Bandera auxiliar que evita que una acción se repita muchas veces si se mantiene una tecla presionada.

Variables de animación (Puertas): Estas variables no ayudan a gestionar los ángulos, estados necesarios, así como la ubicación de sus bisagras y estados necesarios para abrir o cerrar puertas.

Nombre	Tipo	Descripción
doorE_angle	float	Almacena el ángulo actual de apertura de la puerta de la Entrada.
doorS_angle	float	Almacena el ángulo actual de apertura de la puerta de la Sala.
doorB_angle	float	Almacena el ángulo actual de apertura de la puerta del baño.

doorE_open	bool	Indica si la puerta de la Entrada debe abrirse o cerrarse.
doorS_open	bool	Indica si la puerta de la Sala debe abrirse o cerrarse.
doorB_open	bool	Indica si la puerta del Baño debe abrirse o cerrarse.

Variables de la animación (lampara): Estas variables ayudaran a controlar el movimiento físico simulado y el estado de la luz.

Nombre	Tipo	Descripción
lampWobble	float	Valor numérico que define qué tan inclinada está la lámpara
lampAnimationActive	bool	Funciona como interruptor para activar o desactivar el movimiento de la lámpara
lampLightActive	bool	Variable lógica que permite habilitar o deshabilitar el control de la luz
lampState	bool	Indica si el foco de la lámpara se encuentra encendido (verdadero) o apagado (falso)

Variables de animación (TV): Variables que controlan la aparición del fantasma.

Nombre	Tipo	Descripción
tvAnimationActive	Bool	Interruptor principal que inicia la animación del fantasma saliendo de la TV.
tvAnimTime	float	Temporizador que cuenta el tiempo transcurrido de la animación para determinar cuándo detenerla.

Estado del arte

El recorrido virtual desarrollado es un seguimiento de manera simulada porque se realiza de forma digital mediante programación en OpenGL y sirve para que se puedan realizar exploraciones en la computadora, se hace el recorrido como si se estuviera dentro de la caricatura en un espacio físico y sirve para crear formas, videos e imágenes en tercera dimensión. En este proyecto se puede manipular el movimiento de la cámara y ver objetos que se mueven o se quedan estáticos, trabajando con este tipo de recorridos similar a como se hace en la industria de los videojuegos o la arquitectura.

Algunas de las principales características de este proyecto son las siguientes:

La simulación: Donde se puede hacer una recreación de un lugar que es ficticio, trabajando con algo que no existe en el mundo real pero sí en la animación.

Referente a la tecnología: Trabaja con renderizado de gráficos en tiempo real que permiten una visualización de 360 grados y también fueron programados con imágenes (texturas) y modelos con movimientos (animaciones).

En la parte de la interactividad: El usuario tiene la libertad de poderse mover de manera libre por toda la casa y el jardín usando el teclado y el mouse.

El acceso: Es desde un equipo de cómputo (PC) mediante un archivo ejecutable que procesa los gráficos al momento.

El principal propósito: Es que sirve para hacer exploraciones en espacios modelados, recrear imágenes de la serie y demostrar el funcionamiento de las luces y sombras de forma remota.

Este recorrido virtual es una recreación del escenario de Billy y Mandy y es totalmente visual; sirve para simular un recorrido real dentro de la casa donde el usuario se puede mover libremente, simulando el movimiento como si caminara y observando la recreación de formas, figuras y personajes como Billy o el fantasma donde puede interactuar con ellos de forma libre.

Un recorrido virtual como este posibilita la facilidad de recrear un espacio que no existe físicamente un dibujo animado 2D pasado a 3D y también se pueden manipular en la creación de elementos que se deseen, agregando formas, colores, iluminación y movimientos según sean las necesidades del proyecto.

Conclusión

Este proyecto es un recopilatorio de todo lo aprendido a lo largo del semestre, creando un entorno virtual interactivo basado en las aventuras de Billy y Mandy. A través de una herramienta de renderizado, se cumplió con el objetivo de integrar modelado, texturizado, iluminación y animación en tiempo real.

En este proyecto se utilizaron distintas herramientas fundamentales en OpenGL. El desarrollo de un modelo desde cero en Blender fue interesante, ya que es un mundo que genera infinitas posibilidades, aprender a usar esta herramienta fue muy positivo, debido a que me ayuda a visualizar una nueva área de desarrollo profesional.

Dominar los pipelines gráficos es algo complejo, pero permiten un control sobre la visualización, logrando efectos de iluminación detallados utilizando múltiples fuentes de luz que otorgan profundidad y realismo. Por otro lado, las animaciones ayudan a observar cómo funciones matemáticas pueden crear trayectorias no lineales y comportamientos gráficos dinámicos.

Finalmente, este proyecto me ayudó a ver de una forma diferente los gráficos 3D, sobre todo al analizar animaciones antiguas donde ahora puedo detectar errores de geometría, iluminación o texturizado. Sin embargo, con el paso del tiempo, las nuevas herramientas nos han ayudado a mejorar gráficamente hasta tener proyectos visuales donde la línea entre el realismo y lo generado por un motor gráfico es cada vez más delgada.

Referencias

Enlace en el cual se obtuvo el modelo del fantasma utilizado en el proyecto:

1.-Cruz Ponce, A. A. (2021, 17 de abril). Fantasma [Modelo 3D]. Sketchfab. <https://sketchfab.com/3d-models/fantasma-8928e3d13773497291605e5ef1fc0aed>

Enlace en el cual se obtuvo el modelo del personaje utilizado en el proyecto:

2.-calvinwil5782. (2023, 13 de noviembre). Billy [Modelo 3D]. Sketchfab. <https://sketchfab.com/3d-models/billy-04a755991fee4d99ba6027a80e6aeb65>

3.-Integración de GitHub Desktop con el Proyecto de Computación ... [Video]. (n.d.). YouTube. <https://www.youtube.com/watch?v=XZYgHmnB1vU>

4.-Animación Básica en OpenGL [Video]. (n.d.). YouTube. <https://www.youtube.com/watch?v=luybf6bTzhc>

- 5.- Carga de Texturas en OpenGL Introducción a las UV s [Video]. (n.d.). YouTube. https://www.youtube.com/watch?v=21jOb3w_T8k
- 6.-Animación por Keyframes en OpenGL: Movimiento y Rotación [Video]. (n.d.). YouTube. <https://www.youtube.com/watch?v=5l1A6Lxa3GQ>
- 7.-IEP. (2022, 7 de julio). Metodología Waterfall: Modelo de gestión de proyectos en cascada. IEP. Recuperado de <https://iep.edu.es/metodologia-waterfall/>
- 8.-Universidad de Valladolid. (s. f.). Metodología Waterfall (modelo en cascada). Universidad de Valladolid. Recuperado de <https://uvadoc.uva.es/bitstream/10324/71366/1/TFG-G6918.pdf>
- 9.-Rominaescalonab. (s. f.). 59 ideas de Billy y Mandy [Tablero de imágenes]. Pinterest. Recuperado de <https://cl.pinterest.com/rominaescalonab/billy-y-mandy/>
- 10.- Indeed. (2025, 14 de noviembre). *Sueldo de Programador/a junior en México* [Página web]. <https://mx.indeed.com/career/programador-junior/salaries>
- 11.- Indeed. (s. f.). Sueldos de CONSULTORIA TI en México: ¿Cuánto paga CONSULTORIA TI? [Página web]. <https://mx.indeed.com/cmp/Consultoria-Ti/salaries>
- 12.- Liverpool. (s. f.). Laptop gamer HP Victus 15-FA2701WM 15.6 pulgadas Full HD Intel Core i5 NVIDIA GeForce RTX 4050 16 GB RAM 512 GB SSD [Página de producto]. <https://www.liverpool.com.mx/tienda/pdp/laptop-gamer-hp-victus-15-fa2701wm-15-6-pulgadas-full-hd-intel-core-i5-nvidia-geforce-rtx-4050-16-gb-ram-512-gb-ssd/1179817655?gclidc=aw.ds&...>
- 13.- Grupo Industronic. (2025, 9 de junio). Costo de electricidad en México: tarifas y cómo reducir la ... [Página web]. <https://grupoindustronic.com/costo-de-electricidad/>
- 14.- Telmex. (s. f.). Paquetes de Internet para hogar [Página web]. https://telmex.com/web/hogar/mkt/paquetes-de-internet?utm_source=performance_max&utm_medium=cpa&utm_campaign=performance&utm_id=ads001&utm_term=80megas.netflix.sincosto.cdmx&...
- 15.- Google. (s. f.). AI y herramientas de productividad – Google One AI [Página web]. https://one.google.com/ai?utm_source=g1&utm_medium=paid_media&utm_campaign=sem_aiplans&...
- 16.- El Economista. (2024, 20 de julio). ¿Cuánto cuesta rentar una habitación con servicios incluidos en la CDMX? EconoHábitat. Recuperado de <https://www.eleconomista.com.mx/econohabitat/Cuanto-cuesta-rentar-una-habitacion-con-servicios-incluidos-en-la-CDMX-20240719-0078.html>

Resultados finales del proyecto

