



Computación Gráfica e Interacción

Humano-Computadora

Profesor: Ing. Roman Balbuena
Carlos Aldair

Semestre 2025-2

Manual técnico

Grupo: 05

Alumno:

319033993

Índice o contenido

1. Objetivos	3
2. Alcance del proyecto.....	3
3. Diagrama de Gantt.....	5
4. Diccionario de funciones y variables	5
5. Diagrama de flujo	8
6. Análisis de costos	9
7. Conclusión	11

1. *Objetivos*

- **Desarrollar un recorrido virtual interactivo utilizando OpenGL 3.3**, que incluya la representación realista de una fachada y dos cuartos interiores, basados en imágenes de referencia seleccionadas cuidadosamente.
- **Integrar una cámara sintética manipulable en tiempo real**, que permita la exploración libre y natural del espacio virtual, facilitando una experiencia inmersiva para el usuario.
- **Crear y programar cuatro animaciones dentro del recorrido**, que aporten dinamismo y realismo al entorno, mostrando interacciones relevantes con los objetos y elementos del espacio.

2. *Alcance del proyecto*

Funcionalidades Incluidas

El presente proyecto consistió en el desarrollo completo de un recorrido virtual interactivo utilizando OpenGL 3.3, basado en la fachada y dos cuartos interiores del sitio “Random Play”.

Se logró incluir modelos 3D detallados, texturizados, así como diversas fuentes de iluminación, incluyendo iluminación direccional y spotlight, que aportan profundidad y ambiente al espacio virtual. Además, se integraron animaciones controladas por el usuario, permitiendo la interacción dinámica con objetos dentro del entorno, como aperturas de puertas y ventanas, que enriquecen la experiencia inmersiva.

El movimiento de cámara fue implementado de forma fluida y en tiempo real, ofreciendo libertad para explorar el espacio desde diferentes ángulos y perspectivas, mejorando la navegación y la interacción del usuario con el recorrido.

El proyecto abarca desde la modelación y texturizado de los elementos arquitectónicos hasta la programación de la lógica de interacción y animaciones, culminando con la entrega de un ejecutable funcional y la documentación técnica completa en dos idiomas.

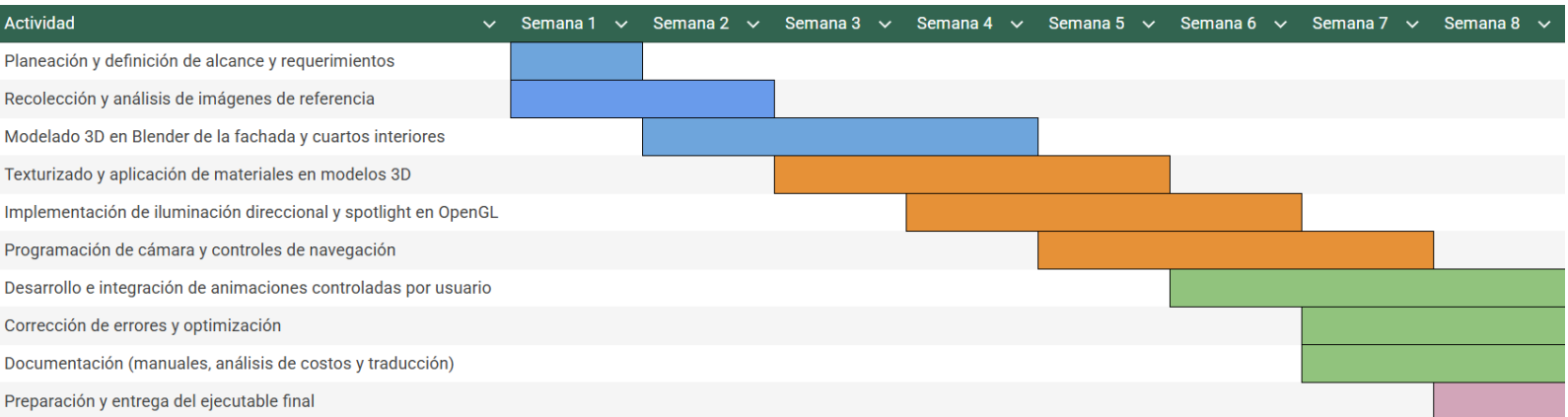
Limitaciones del Proyecto

A pesar de los logros alcanzados, el proyecto enfrentó ciertas limitaciones que afectaron su desarrollo y alcance:

- **Recursos técnicos:** La capacidad de procesamiento y memoria disponible en las estaciones de trabajo condicionó la complejidad de las texturas y el número de luces dinámicas que se pudieron implementar sin afectar el rendimiento en tiempo real.
- **Herramientas y entorno:** Aunque Blender y OpenGL 3.3 ofrecieron amplias posibilidades, algunas funcionalidades avanzadas como shaders personalizados o efectos visuales complejos no se exploraron a fondo debido a la curva de aprendizaje y el tiempo disponible.
- **Alcance funcional:** La interacción se limitó a cuatro animaciones controladas por el usuario; funciones adicionales como física avanzada, sonido o inteligencia artificial no fueron incluidas para mantener el proyecto manejable dentro de los recursos y tiempos disponibles.

Estas limitaciones fueron tomadas en cuenta desde la planeación para definir un alcance realista que permitiera entregar un producto funcional y de calidad, a la vez que respetaba las restricciones impuestas por el contexto del proyecto.

3. Diagrama de Gantt



4. Diccionario de funciones y variables

Funciones principales

int main()

Función principal donde se inicializan GLFW y GLEW, se crea la ventana, se configuran los shaders, se cargan los modelos 3D y se ejecuta el bucle principal de renderizado y animación. Controla el ciclo completo de ejecución del programa y la gestión de eventos.

void DoMovement()

Gestiona el movimiento de la cámara en respuesta a la entrada del usuario (teclas W, A, S, D, Q, E). Permite desplazamientos tridimensionales y rotaciones laterales, facilitando una exploración dinámica y fluida del entorno virtual.

void KeyCallback(GLFWwindow* window, int key, int scancode, int action, int mode)

Manejador de eventos del teclado que detecta pulsaciones y liberaciones de teclas. Actualiza el estado interno de las teclas y controla acciones específicas, como cerrar la ventana (ESC) o activar animaciones con teclas V, M, B y N, proporcionando interactividad al usuario.

void Animation()

Actualiza las posiciones y ángulos de rotación de los objetos animados (ventana, puerta, laptop) según estados y límites predefinidos. Implementa movimientos suaves y oscilatorios.

void MouseCallback(GLFWwindow* window, double xPos, double yPos)

Captura el movimiento del mouse para controlar la orientación de la cámara. Calcula el desplazamiento entre posiciones consecutivas para rotar la vista, mejorando la inmersión y el control intuitivo del usuario.

Variables

Shader lightingShader

Objeto shader encargado de controlar la iluminación en la escena. Gestiona cálculos en la GPU para aplicar efectos de luz que realzan el realismo visual de los modelos 3D.

Directional Light (dirLight)

Luz direccional que simula una fuente lejana con dirección fija $(-0.2, -1.0, -0.3)$. Proporciona iluminación uniforme como la luz solar, afectando todas las superficies. Sus componentes ambiental, difusa y especular definen cómo interactúa la luz ambiental, directa y los reflejos con los materiales, aportando profundidad y volumen a la escena.

Camera camera(glm::vec3(0.0f, 0.0f, 3.0f))

Objeto que representa la cámara 3D, ubicada inicialmente sobre el eje Z. Controla la vista y proyección en la escena, permitiendo la navegación y el encuadre dinámico.

bool keys[1024]

Arreglo que almacena el estado (presionada o liberada) de cada tecla, usado para procesar la entrada de usuario de manera eficiente y continua.

GLfloat deltaTime

Variable que registra el tiempo transcurrido entre el frame actual y el anterior. Se utiliza para normalizar movimientos y animaciones, asegurando que la velocidad sea constante independientemente de la tasa de fotogramas.

Variables de animación (ventcMovingRight, ventcPos, ventRotationAngle, etc.)

Controlan el estado y parámetros de las animaciones de objetos específicos, como la posición y rotación de ventanas, puertas y laptops. Permiten implementar movimientos naturales y sincronizados con el tiempo.

Shader lightingShader, Shader lampShader

Shaders encargados de los cálculos de iluminación general, responsables de efectos visuales en la escena.

Model compu, vent, ventc, puert, ran;

Modelos 3D cargados y renderizados en la escena, cada uno con geometría, texturas y transformaciones propias que contribuyen a la composición visual del entorno.

bool firstMouse = true;

Indicador que señala la primera captura de movimiento del mouse para evitar saltos abruptos en la cámara.

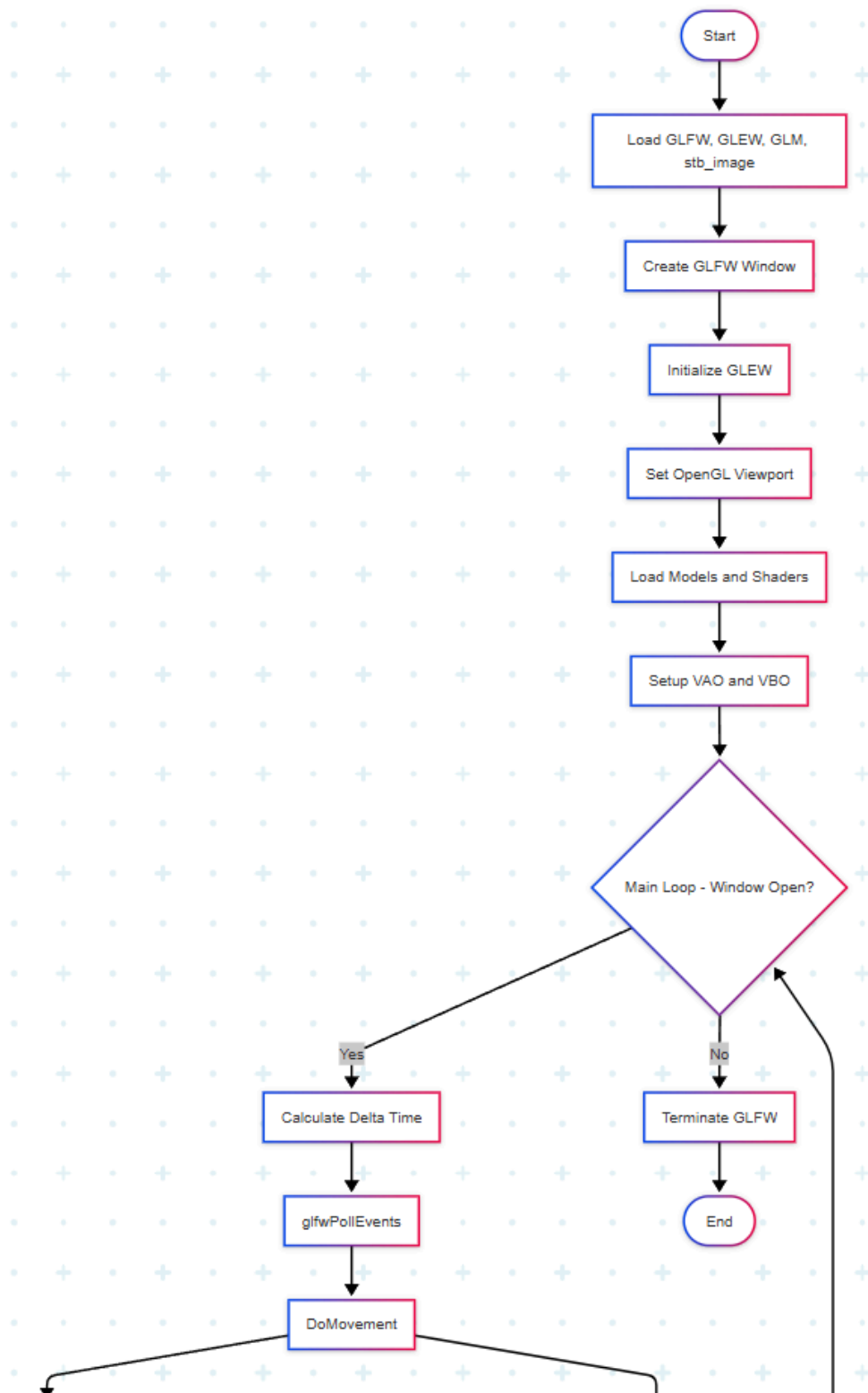
float vertices[]

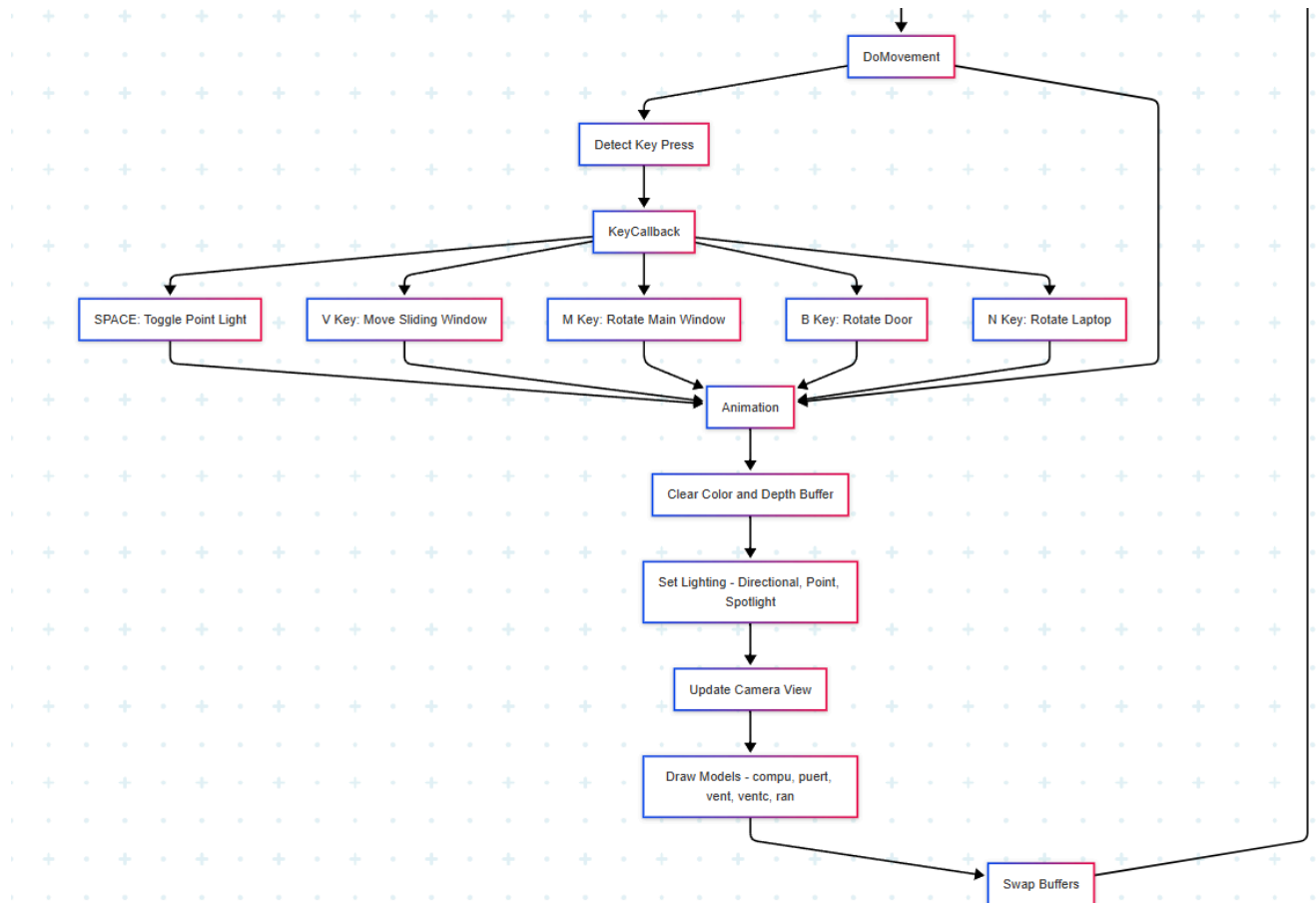
Arreglo que define la geometría base de un cubo 3D, incluyendo posiciones y normales para el cálculo correcto de iluminación. Sirve como estructura fundamental para construir objetos en la escena.

Matrices de transformación (glm::mat4 modelCompu, modelPuert, etc.)

Matrices que gestionan la posición, rotación y escala de cada modelo 3D. Son importantes para animar y colocar objetos en el espacio tridimensional, facilitando movimientos y cambios visuales.

5. Diagrama de flujo





6. Análisis de costos

Para el desarrollo del recorrido virtual interactivo utilizando OpenGL 3.3, se realizó un análisis detallado de los costos involucrados, contemplando los recursos humanos, herramientas, tiempo estimado, riesgos y márgenes necesarios para asegurar la calidad y sostenibilidad del proyecto.

Objetivo del Proyecto

Desarrollar un recorrido virtual interactivo utilizando OpenGL 3.3, que incluya:

- Modelado 3D realista de una fachada y dos cuartos interiores.
- Cámara interactiva en tiempo real para exploración libre.
- Cuatro animaciones (puertas, ventanas, objetos interactivos).
- Iluminación dinámica (directional light + spotlight).
- Ejecutable funcional + documentación técnica bilingüe.

Desglose de Costos

Concepto	Descripción	Horas Estimadas	Costo (USD)
1. Modelado 3D	Creación de modelos 3D (fachada, cuartos, objetos) en Blender.	60 horas	\$2,400
2. Texturizado	Aplicación de texturas realistas (paredes, muebles, pisos).	30 horas	\$1,200
3. Programación OpenGL	Implementación de renderizado, shaders, cámara, iluminación y animaciones.	100 horas	\$5,000
4. Integración de Animaciones	Programación de 4 animaciones (puertas, ventanas, laptop, etc.).	40 horas	\$2,000
5. Optimización	Ajuste de rendimiento (gestión de memoria).	20 horas	\$1,000
6. Pruebas y Depuración	Validación de interacciones, corrección de bugs.	30 horas	\$1,500
7. Documentación	Manual técnico (español + inglés) y guía de usuario.	20 horas	\$800
8. Licencias y Herramientas	Blender (gratis), Visual Studio (gratis), posible licencia de assets.	-	\$300
9. Contingencias (10%)	Imprevistos (cambios de diseño, problemas técnicos).	-	\$1,520
Total		300 horas	\$15,720.00

Concepto	Descripción	Precio (USD)
Costo Total	Suma de todos los recursos y actividades.	\$15,720
Margen de Ganancia (30%)	Utilidad esperada.	\$4,716
Precio Base	Costo total + margen.	\$20,436
IVA/Taxes (16%)	IVA (16%)	\$3,270
ISR (10% sobre utilidad)	ISR	\$472
Precio Final de Venta	Total a facturar al cliente.	\$24,178

- Costo total (15,720)+*Ganancia*(4,716) =\$20,436.
- IVA se aplica sobre 20,436→3,270.
- ISR se aplica solo sobre la ganancia (4,716)→472.
- **Total:** 20,436+3,270 + 472=24,178

Observaciones y Justificación

Uso de herramientas gratuitas: El desarrollo se apoya en herramientas open source como *Blender* y *Visual Studio Code*, eliminando gastos por licenciamiento y permitiendo mayor inversión en desarrollo técnico y visual.

Texturas libres o propias: Se emplearon recursos de dominio público o generados manualmente, minimizando gastos en bancos de recursos.

Contingencias: Se consideró un 10% del costo base para cubrir imprevistos como rediseños, bugs complejos o peticiones adicionales del cliente.

Ganancia proyectada: El margen del 30% asegura una compensación adecuada por la especialización del trabajo, inversión de tiempo, soporte y mantenimientos menores posteriores a la entrega.

Costos adicionales posibles:

- Compra de modelos 3D premium si se desea un mayor nivel de detalle.
- Uso de herramientas de pago como *Substance Painter* para texturizado profesional.
- Soporte post-venta o actualización de funcionalidades a futuro.

Este análisis no solo proporciona transparencia en los recursos utilizados, sino que permite establecer un precio justo y competitivo que garantiza la viabilidad económica del proyecto tanto para el desarrollador como para el cliente final.

7. Conclusión

El desarrollo del recorrido virtual interactivo utilizando OpenGL 3.3 ha cumplido con los objetivos planteados, logrando una experiencia inmersiva y dinámica basada en la representación realista de una fachada y dos cuartos interiores del sitio “Random Play”.

Se implementaron modelos 3D detallados y texturizados con precisión, aplicando técnicas de iluminación direccional y spotlight. La integración de una cámara sintética manipulable en tiempo real permitió una navegación fluida.

Durante el desarrollo, se siguió una metodología estructurada que permitió planear, implementar, probar y documentar cada fase de manera organizada, lo que se tradujo en un producto final funcional. La documentación técnica completa en español e inglés facilita la comprensión y uso del sistema por parte de usuarios y futuros desarrolladores.

El análisis de costos realizado demuestra una estimación realista y detallada, que contempla todos los recursos, materiales, licencias y gastos indirectos, además de impuestos y márgenes comerciales. Esto asegura la viabilidad económica del proyecto y su posicionamiento competitivo en el mercado.

A pesar de ciertas limitaciones técnicas y funcionales —como la ausencia de efectos visuales avanzados y la limitación del hardware disponible— estas fueron consideradas desde la planeación para establecer un alcance realista y manejable. Así, se entregó un producto coherente con los tiempos y recursos disponibles, sin comprometer la calidad y el cumplimiento de los requisitos.

En conclusión, este proyecto representa una sólida aplicación práctica de conceptos y técnicas avanzadas en computación gráfica e interacción humano-computadora, evidenciando dominio en modelado, animación y programación gráfica. Además, el proceso completo —desde el diseño hasta la documentación— refleja una gestión adecuada y profesional del desarrollo de software.