

*Manual Técnico*

*Entorno virtual*

*Juguetería*

*Universidad Nacional Autónoma de México*

*Facultad de Ingeniería*

*Laboratorio de Computación Gráfica*

*e Interacción Humano-Computadora*

*Grupo 12*

*Elaborado por*

*316017862*

## Contenido

Objetivos .....	3
Introducción .....	3
Configuración .....	3
Alcance del proyecto.....	3
Limitantes .....	4
Metodología.....	4
Diagrama de Gantt .....	5
Documentación del código.....	6
Conclusiones.....	10

## *Objetivos*

Documentar y brindar la información necesaria para poder realizar la configuración necesaria para poder realizar la ejecución de la aplicación del entorno virtual.

Brindar la información de cómo fue desarrollado el entorno virtual, el funcionamiento de cada una de las variables y funciones dentro del programa.

Con los 2 objetivos anteriores y la aplicación del entorno virtual, el alumno demuestra y aplica los conocimientos adquiridos durante todo el curso.

## *Introducción*

A lo largo del documento, se proporciona la información necesaria para la ejecución de manera correcta de la aplicación del entorno virtual. Además, se realiza un reporte del proceso de elaboración del proyecto, como la explicación del funcionamiento que código empleado en la aplicación.

## *Configuración*

Para la ejecución de la aplicación, se requiere que en la carpeta donde se localice “ProyectoFinal.exe” se encuentren las bibliotecas glew32.dll y assimp-vc-140-mt.dll para que se puedan importar los modelos y evitar que nos provoque errores de carga. Adicional a estos archivos, es necesario tener de igual manera todos los modelos, Shaders y las imágenes de las texturas de los modelos para que la ejecución de la aplicación se realice de manera correcta.

## *Alcance del proyecto*

El alumno deberá seleccionar una fachada y un espacio que pueden ser reales o ficticios y presentar imágenes de referencia de dichos espacios para su recreación 3D en OpenGL.

Se deberá de contar con 7 objetos que el alumno deberá recrear virtualmente, los cuales deberán tener sentido en el contexto de la fachada seleccionada.

Se deberá de contar con cinco animaciones donde tres sean sencillas y 2 complejas.

El entorno deberá de contar con un correcto uso de la iluminación.

## *Limitantes*

Las principales limitantes durante el desarrollo del entorno fue el conocimiento teórico, como en el uso de las herramientas utilizadas. Adicional a esto, el paro ocurrido durante los días 19 de abril de 2024 al 4 de mayo de 2024 nos quitaron prácticas importantes para el desarrollo del proyecto afectando el desarrollo y algunas actividades esenciales para éste. Otra limitante importante, que muchas veces no se tiene en cuenta, es el equipo de cómputo con el que cuento, éste puede afectar a la eficiencia en la realización de cada actividad, debido a que no se cuenta con el equipo de cómputo óptimo para el uso de aplicaciones de modelado 3D y con el desconocimiento de su uso y en la elaboración de modelos 3D, provoca que el desarrollo, integración y procesamiento de nuestra computadora, con recursos limitados, al compilar nuestros programas no sea los más eficiente posible.

## *Metodología*

La metodología utilizada para el desarrollo del entorno virtual fue *Waterfall* (en cascada), esta es una metodología que se divide en distintas fases, cada una de las fases comienza solamente cuando ha terminado la fase anterior. El motivo por lo que se ha decidido tomar como base esta metodología es por la necesidad de tener que tomar el tema, para poder adquirir el conocimiento necesario para lograr realizar cada una de las actividades.

Además, la limitación del tamaño en equipo que provoca que no se puedan desarrollar fácilmente las actividades de manera simultánea. A pesar de que, en el diagrama de Gantt, se muestran actividades que se realizan de manera paralela, algunas de ellas están sujetas a otras, pero al desconocer el tiempo que se pueda tardar en desarrollar dicha actividad, se ha decidido dejar en el mismo periodo de tiempo ambas actividades, para no poner fechas límites entre esas actividades y tener la flexibilidad de tiempo entre ellas, ya que esas actividades se entregarían el mismo día.

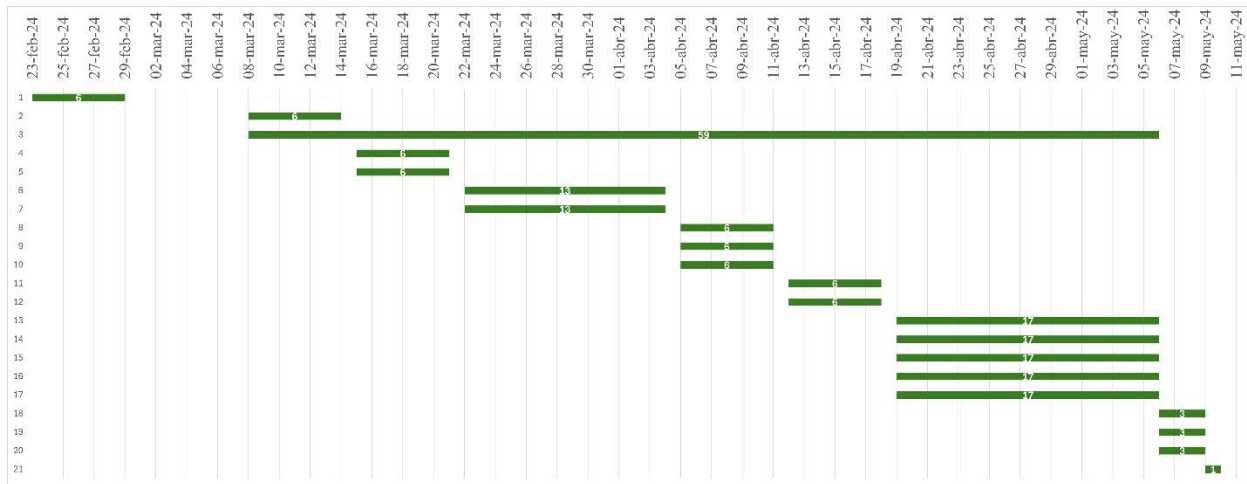
Aunque la metodología principal es *Waterfall*, en ocasiones tuvimos que realizar correcciones a ciertos modelos o configuraciones pasadas, pero sin tener que deshacer todo lo hecho. Por lo que también se ha utilizado de manera paulatina una metodología *Agile*, donde es un proceso cíclico, donde se tuvo un desarrollo, una integración y un mantenimiento en alguna de las actividades realizadas. Además, permitiéndome tener pasos incrementales e iterativos que me permitían tener pequeños entregables para la demostración del avance en el proyecto.

## Diagrama de Gantt

### Listado de actividades

# Act	Nombre Actividad	Fecha de Inicio	Duración en días	Fecha Fin
1	Toma de requerimientos y selección de fachada y modelos a recrear	23-feb-24	6	29-feb-24
2	Recreación de uno de los modelos (1er Modelo)	08-mar-24	6	14-mar-24
3	Recreación de la fachada	08-mar-24	59	06-may-24
4	Recreación de otro modelo (2do Modelo)	15-mar-24	6	21-mar-24
5	Texturizado del 1er y 2do modelo	15-mar-24	6	21-mar-24
6	Recreación de otro modelo (3er Modelo)	22-mar-24	13	04-abr-24
7	Texturizado del 3er modelo	22-mar-24	13	04-abr-24
8	Recreación de otro modelo (4er Modelo)	05-abr-24	6	11-abr-24
9	Texturizado del 4to modelo	05-abr-24	6	11-abr-24
10	Adecuar iluminación en OpenGL	05-abr-24	6	11-abr-24
11	Recreación de otro modelo (5to Modelo)	12-abr-24	6	18-abr-24
12	Texturizado del 5to modelo	12-abr-24	6	18-abr-24
13	Creación de animación por KeyFrames	19-abr-24	17	06-may-24
14	Creación de animaciones	19-abr-24	17	06-may-24
15	Recreación de otro modelo (6to Modelo)	19-abr-24	17	06-may-24
16	Recreación de otro modelo (7mo Modelo)	19-abr-24	17	06-may-24
17	Terminar de recrear todo el entorno e integrar correctamente los modelos	19-abr-24	17	06-may-24
18	Creación de Manual Técnico	06-may-24	3	09-may-24
19	Creación de Manual de Usuario	06-may-24	3	09-may-24
20	Configuración de proyecto	06-may-24	3	09-may-24
21	Entrega de proyecto	09-may-24	1	10-may-24

## Diagrama



## Documentación del código

### Diccionario de Include

<b>Include</b>	<b>Funcionamiento</b>
<i>iostream</i>	Proporciona funcionalidades para entrada y salida de datos.
<i>cmath</i>	Proporciona funciones matemáticas.
<i>GL/glew.h</i>	Proporciona funciones para cargar extensiones de OpenGL de forma dinámica.
<i>GLFW/glfw3.h</i>	Proporciona una API para crear ventanas, gestionar eventos de entrada y manejar contextos de OpenGL.
<i>stb_image.h</i>	Permite cargar imágenes en formato de mapa de bits (.jpg, .png, etc.).
<i>glm/glm.hpp</i>	Proporciona clases y funciones matemáticas diseñadas para trabajar con OpenGL. Incluye tipos de datos y operaciones para vectores, matrices y transformaciones.
<i>glm/gtc/matrix_transform.hpp</i>	Proporciona funciones para crear y manipular matrices de transformación, como matrices de vista, proyección y modelos.
<i>glm/gtc/type_ptr.hpp</i>	Proporciona funciones para convertir tipos de datos GLM en tipos de datos nativos de OpenGL.
<i>Camera.h</i>	Proporciona la funcionalidad necesaria para controlar una cámara en un entorno OpenGL
<i>Model.h</i>	Proporciona la funcionalidad necesaria para cargar y dibujar modelos 3D en un entorno OpenGL

*Texture.h*

Proporciona la funcionalidad necesaria para cargar texturas en OpenGL, usada principalmente para el Skybox

## Diccionario de Variables

<b>Variable</b>	<b>Funcionamiento</b>
<i>WIDTH</i>	Guarda el ancho de la ventana
<i>HEIGHT</i>	Guarda la altura de la ventana
<i>SCREEN_WIDTH</i>	Guarda el ancho de la pantalla
<i>SCREEN_HEIGHT</i>	Guarda la altura de la ventana
<i>camera</i>	Variable que nos permite realizar el manejo de la cámara
<i>lastX</i>	Guarda la posición Inicial del mouse sobre el eje X cuando se inicia la aplicación
<i>lastY</i>	Guarda la posición Inicial del mouse sobre el eje Y cuando se inicia la aplicación
<i>keys[1024]</i>	Es un arreglo de booleanos donde cada elemento representa un tecla y sirve para comprobar si una tecla ha sido presionada
<i>firstMouse</i>	Variable que sirve para entrar a una condicional donde se obtiene la posición inicial del mouse al ejecutarse la aplicación
<i>SpotPos</i>	Variable vec3 que almacena la posición de la luz de tipo Spot
<i>SpotDir</i>	Variable vec3 que almacena la dirección de la luz de tipo Spot
<i>pointLightPositions</i>	Arreglo de vec3 que almacena las posiciones de las 4 luces de tipo Point
<i>PosIni</i>	Variable que almacena la posición inicial del modelo del Robot que esta animado por keyframes
<i>PosIni2</i>	Variable que almacena la posición inicial del modelo de la avioneta que esta animado por keyframes
<i>deltaTime</i>	Variable que almacena la diferencia de tiempo entre el Frame actual y el anterior
<i>lastFrame</i>	Almacena el valor de tiempo del último frame
<i>posX, posY, posZ</i>	Variables que nos ayudan a modificar por medio de los keyframes la posición del robot
<i>posX2, posY2, posZ2</i>	Variables que nos ayudan a modificar por medio de los keyframes la posición de la avioneta

<i>rotRodIzq</i>	Variable usada para variar la rotación de brazos y piernas del robot para simular que camina.
<i>ERotZ</i>	Variable utilizada para rotar por medio de keyframes la hélice de la avioneta
<i>ARotY</i>	Variable utilizada para rotar por medio de keyframes la dirección de la avioneta
<i>MAX_FRAMES</i> , <i>MAX_FRAMES2</i>	Variables que almacenan el valor máximo de Frames para cada una de las animaciones por keyframes
<i>i_max_steps</i> , <i>i_max_steps2</i>	El número máximo de los cálculos entre cada frame
<i>i_curr_steps</i> , <i>i_curr_steps2</i>	Contador que nos indica cuantos pasos llevamos de los cálculos entre cada frame
<i>_frame</i>	Estructura de datos que contiene las variables necesarias para poder realizar las animaciones del robot
<i>_frame2</i>	Estructura de datos que contiene las variables necesarias para poder realizar las animaciones de la avioneta
<i>KeyFrame[MAX_FRAMES]</i> , <i>KeyFrame2[MAX_FRAMES2]</i>	Arreglo del tipo <i>_frame/_frame2</i> que almacenara los keyframes de las distintas animaciones.
<i>FrameIndex</i> , <i>FrameIndex2</i>	Variables que indican en que frame es el siguiente al estar guardando los keyframes de la animación
<i>play</i> , <i>play2</i>	Variables que permiten ejecutar las distintas animaciones
<i>playIndex</i> , <i>playIndex2</i>	Variables que nos indican cual es el frame que se esta dibujando de la animación
<i>window</i>	Variable que guarda la creación de la ventana
<i>skyboxVertices</i>	Arreglo de vertices que nos permite crear un cubo para texturizarlo como skybox del entorno
<i>VBO</i>	Vertex Buffer Object
<i>VAO</i>	Vertex Array Object
<i>faces</i>	Arreglo que almacena cada una de las caras del skybox
<i>projection</i>	Variable que almacena la información del tipo de perspectiva que tiene la cámara
<i>currentFrame</i>	Almacena el tiempo del frame actual
<i>model</i>	Variable que almacena toda la información de las transformaciones de nuestro modelos
<i>tmp</i> , <i>tmp2</i>	Variables que nos ayudan a setear las transformaciones de la variable <i>model</i> para reducir los cálculos que se tendrían que hacer desde 0 para volver a llegar al mismo punto.



<i>puerta</i>	Booleano que nos ayuda a activar la animación de la puerta
<i>P_Play</i>	Booleano que ayuda a que termine la animación de la puerta
<i>P_caso</i>	Entero que nos ayuda a indicar en qué estado se encuentra la puerta (cerrada o abierta)
<i>rotPuerta</i>	Flotante que nos ayuda a almacenar la rotación de la puerta para la animación
<i>rotTren</i>	Variable que ayuda a controlar la posición y rotación de la animación del tren

## Diccionario de funciones

<b>Función</b>	<b>Funcionamiento</b>
<i>saveFrame2</i>	Guarda los keyframes para la animación de la avioneta
<i>saveFrame</i>	Guarda los keyframes para la animación del robot
<i>resetElements2</i>	Resetea al estado del frame inicial de la avioneta
<i>resetElements</i>	Resetea al estado del frame inicial del robot
<i>interpolation2</i>	Calcula los incrementos necesarios entre cada frame de la animación de la avioneta.
<i>interpolation</i>	Calcula los incrementos necesarios entre cada frame de la animación del robot
<i>animacion2</i>	Ejecuta la animación de la avioneta
<i>animacion</i>	Ejecuta la animación del robot
<i>main</i>	Función principal donde se crean los buffers, se realiza la configuración de la ventana, la carga y dibujo de los modelos, luces, skybox.
<i>DoMovement</i>	Función que detecta las teclas wasd y genera el movimiento dentro de nuestro entorno. Además, contiene las animaciones sencillas que son activadas por una tecla.
<i>KeyCallback</i>	Detecta si alguna tecla ha sido pulsada.
<i>MouseCallback</i>	Calcula la posición del mouse que nos permite detectar su movimiento

## *Conclusiones*

El proceso de la creación de un entorno virtual requiere de demasiado tiempo y al no tener el conocimiento suficiente ralentiza todo el proceso de la creación. Sin embargo, considero que he logrado crear un entorno bastante bueno, aunque me hubiera gustado llenar los estantes de varios modelos, el tiempo que perdí por mi inexperiencia provoco que no pudiera acomodar de mejor manera la repetición de los modelos que ya tenía cargados. Estoy satisfecho con el resultado a pesar de lo anterior mencionado, ya que pude aplicar correctamente lo aprendido durante el curso.