UNIVERSIDAD NACIONAL DE INGENIERIA

DIRECCION AREA DE CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN

PROGRAMA ACADEMICO DE INGENIERIA EN COMPUTACION

Proyecto de:

**ModelXplorer**

Presentado por:

Br. Dávila Sánchez Francisca Martina Carnet: 2022-0385U

Br. Pastran Moraga Jesly Isayana Carnet: 2022-0485U

Br.  Rivas Mendoza Eddy Jesús Carnet: 2022-1230U

Br. Mayorga Salinas Juridia del Rosario Carnet: 2022-0365U

Grupo:

3T1 - CO

Docente:

Ing. Danny Oswaldo Chávez Miranda

Managua, Nicaragua

Julio 2024

Contenido

[I. INTRODUCCIÓN 1](#_Toc171040471)

[I. OBJETIVOS 2](#_Toc171040472)

[**OBJETIVO GENERAL:** 2](#_Toc171040473)

[**OBJETIVO ESPECIFICO:** 2](#_Toc171040474)

[**Desarrollo de la Aplicación de Gráficos 3D** 3](#_Toc171040475)

[**Constructor y Configuración Inicial** 3](#_Toc171040476)

[**Funcionalidades de la Cámara** 3](#_Toc171040477)

[Clase Application 3](#_Toc171040478)

[Archivos de Código 4](#_Toc171040479)

[**Archivo .cpp (Application.cpp)** 4](#_Toc171040480)

[**Archivo .h (Application.h)** 4](#_Toc171040481)

[**Clase Camera** 4](#_Toc171040482)

[**Clase ImGuiView** 4](#_Toc171040483)

[**Funcionalidad Principal de ImGuiView** 4](#_Toc171040484)

[**Integración de ModelLoader** 5](#_Toc171040485)

[IIl. CONCLUSIÓN 6](#_Toc171040486)

[Bibliografía 7](#_Toc171040487)

# INTRODUCCIÓN

ModelXplorer es una plataforma innovadora que transforma cómo los desarrolladores y usuarios trabajan con modelos 3D. En un mundo donde el 3D es esencial en diseño, animación, videojuegos y realidad virtual, ModelXplorer ofrece herramientas avanzadas y accesibles para la exportación de modelos 3D.

La plataforma permite manejar coordenadas tridimensionales (Eje X, Eje Y, Eje Z) con precisión y ajustar cada aspecto de los modelos, incluyendo iluminación para dar realismo y profundidad.

Además, integra skybox para crear entornos envolventes y sonido para una experiencia sensorial completa. Así mismo, facilita la interacción con el teclado, permitiendo a los desarrolladores crear aplicaciones interactivas y dinámicas. Con una interfaz fácil y accesible, la plataforma es adecuada para usuarios de todos los niveles, asegurando modelos exportados eficientes y de alta calidad.

En resumen, ModelXplorer es una herramienta integral que empodera a los profesionales del 3D, combinando precisión, inmersión e interactividad en una sola plataforma.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

# OBJETIVOS

## **OBJETIVO GENERAL:**

Hacer que ModelXplorer sea Super Fácil de Usar: Queremos asegurarme de que cualquier persona, sin importar su experiencia, pueda usar ModelXplorer sin complicaciones. Quiero simplificar todo el proceso de importar modelos 3D para que sea rápido y sencillo.

## **OBJETIVO ESPECIFICO:**

Crear Herramientas de Análisis y Visualización Avanzada: Integrar herramientas que permitan a los usuarios analizar y visualizar datos complejos relacionados con los modelos 3D, facilitando la toma de decisiones y la optimización del diseño.

## **Desarrollo de la Aplicación de Gráficos 3D**

Este documento detalla el desarrollo de una aplicación de gráficos 3D utilizando C++ junto con las bibliotecas OpenGL, GLFW, GLAD y ImGui. El objetivo principal es crear una plataforma flexible para renderizar modelos tridimensionales y entornos de skybox, integrando una interfaz de usuario interactiva que facilita la manipulación y visualización de objetos en tiempo real.

La base del proyecto es la clase Application, que centraliza todas las funciones críticas de la aplicación. Esto incluye la inicialización de la ventana principal mediante GLFW para la gestión de ventanas y el contexto OpenGL. Adicionalmente, se utiliza GLAD para cargar dinámicamente todas las funciones de OpenGL necesarias, asegurando compatibilidad y rendimiento óptimo con la tarjeta gráfica del sistema.

La configuración de shaders es esencial para definir cómo se procesan los vértices y fragmentos de los objetos renderizados en la escena. Se emplean archivos de shader específicos para lograr efectos visuales personalizados y realistas.

La interfaz de usuario se construye utilizando ImGui, una herramienta que permite crear controles gráficos interactivos de manera eficiente. Esto incluye desde herramientas de edición hasta la visualización de datos complejos, como los ejes de coordenadas, y los controles de configuración de skybox, todo integrado directamente sobre la ventana de OpenGL.

## **Constructor y Configuración Inicial**

El constructor de Camera recibe parámetros como la posición inicial de la cámara, el vector de dirección hacia arriba y los ángulos de rotación. Utiliza estos valores para inicializar variables internas como la velocidad de movimiento, la sensibilidad del ratón y el nivel de zoom. La dirección inicial hacia la que apunta la cámara se establece hacia adelante.

## **Funcionalidades de la Cámara**

Este código proporciona funcionalidades para controlar una cámara virtual en un entorno 3D. Sus métodos incluyen la inicialización con parámetros iniciales como posición y orientación, el procesamiento de entrada del teclado para el movimiento, la gestión del movimiento del ratón para ajustar la vista, y la manipulación del zoom. Además, actualiza internamente los vectores de dirección y calcula la matriz de vista necesaria para renderizar la escena desde la perspectiva correcta. Esta clase es crucial para aplicaciones gráficas que requieren interacción dinámica del usuario con la vista del mundo virtual.

# Clase Application

La clase Application en este proyecto se encarga de inicializar y configurar GLFW y OpenGL, así como preparar shaders para el renderizado. La clase también incorpora una cámara para controlar la vista dentro de la escena. El método run dirige el ciclo principal de la aplicación, gestionando la entrada del usuario, actualizando modelos y la cámara en base al tiempo transcurrido, y renderizando modelos y un skybox con los shaders adecuados. La interfaz de usuario, implementada mediante ImGui, proporciona herramientas interactivas para ajustes en tiempo real sobre la escena renderizada. Callbacks estáticos manejan eventos como el redimensionamiento de la ventana y la entrada del ratón y teclado, interactuando eficientemente con métodos de instancia para ajustar la cámara y gestionar la interacción del usuario. Al finalizar, el método cleanup libera recursos, cerrando la ventana GLFW y liberando la memoria utilizada por objetos y shaders, asegurando una terminación ordenada de la aplicación.

# Archivos de Código

### **Archivo .cpp (Application.cpp)**

Este archivo contiene la implementación concreta de las funciones y métodos declarados en el archivo de encabezado (.h). Aquí se encuentran todos los detalles de cómo se ejecutan las funciones, cómo se manipulan los datos y cómo se interactúa con las bibliotecas y APIs externas.

### **Archivo .h (Application.h)**

Este archivo, conocido como archivo de encabezado, contiene las declaraciones de las clases, funciones y variables que serán utilizadas en otros archivos .cpp donde se incluya este archivo. En el caso de Application.h, contendría la declaración de la clase Application, sus métodos públicos y privados.

### **Clase Camera**

La clase Camera se usa principalmente en aplicaciones gráficas y simulaciones tridimensionales. El archivo camera.cpp no solo simplifica la implementación del control de la cámara en entornos 3D, sino que también mejora la usabilidad y la eficiencia del desarrollo de aplicaciones gráficas. Proporciona un mecanismo robusto para manejar la interacción del usuario con la visualización de la escena.

### **Clase ImGuiView**

El archivo ImGuiView.h define la clase ImGuiView, diseñada para gestionar la interfaz de usuario mediante la biblioteca ImGui. La clase facilita la configuración y el uso de ImGui junto con OpenGL y GLFW para crear una interfaz gráfica interactiva en aplicaciones de gráficos 3D. El método init inicializa ImGui para trabajar con OpenGL y GLFW, configurando el contexto de renderizado y ajustando los estilos visuales según las necesidades del usuario. El método shutdown libera los recursos utilizados por ImGui al cerrar la aplicación, asegurando una gestión adecuada de la memoria. Durante cada ciclo de renderizado, el método newFrame prepara ImGui para un nuevo frame, permitiendo la actualización y renderizado de la interfaz de usuario. La función render se encarga de dibujar la interfaz gráfica utilizando OpenGL, asegurando que los elementos diseñados con ImGui se integren de manera eficiente en la aplicación 3D.

### **Funcionalidad Principal de ImGuiView**

El corazón de la funcionalidad de ImGuiView reside en drawImGuiContent, que permite al usuario interactuar con la carga de modelos 3D desde archivos seleccionados. Además, proporciona controles intuitivos para ajustar la posición, rotación y escala de los modelos cargados, facilitando así la manipulación y visualización de los mismos en tiempo real. Adicionalmente, drawSkyboxControl se encarga de gestionar la carga y ajuste de skyboxes, permitiendo al usuario seleccionar archivos y configurar la escala del skybox cargado. Esta funcionalidad complementa la visualización de modelos 3D al proporcionar un entorno de fondo envolvente.

### **Integración de ModelLoader**

En términos de implementación, ImGuiView utiliza ModelLoader para la carga y gestión de modelos 3D y skyboxes. Esta integración permite manipular de manera efectiva propiedades como la posición, rotación y escala de los objetos en la escena, optimizando así la experiencia de usuario dentro de aplicaciones gráficas complejas. En resumen, ImGuiView ofrece una solución robusta y amigable para la manipulación y visualización de modelos 3D y skyboxes mediante una interfaz gráfica intuitiva**.**

## **Enlace de video y repositorio:**

[**https://youtu.be/YErqipcB9vM**](https://youtu.be/YErqipcB9vM) **https://github.com/Eddy030501/ModelXplorer**

# CONCLUSIÓN

Este proyecto representa un avance significativo en la creación y exportación de modelos 3D, ofreciendo herramientas avanzadas para la manipulación precisa de coordenadas y la creación de entornos envolventes. Con un enfoque en la usabilidad y la innovación tecnológica, se promueve la creatividad y se facilita la producción de experiencias visuales y sensoriales inmersivas. Este enfoque no solo mejora la eficiencia del proceso de diseño, sino que también abre nuevas posibilidades para aplicaciones en diversos campos como la realidad virtual, la educación y el entretenimiento digital.

Bibliografía  
  
-Victor Gordan. (2020, 20 noviembre). *OpenGL Tutorial 0 - Install* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=XpBGwZNyUh0>  
-Code, Tech, and Tutorials. (2021, 22 octubre). *Why and How To Use ImGui For Your Coding Projects | Tutorial & Case Study* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=U1BnzWX194Q>  
-ChatGPT. (n.d.). "<https://chatgpt.com/c/e374acd5-09bf-42f3-bdd2-4793d9785e17>". Retrieved [7/5/2024], from <https://chatgpt.com/c/e374acd5-09bf-42f3-bdd2-4793d9785e17>

-AlonsoCerpa. (s. f.). *GitHub - AlonsoCerpa/NeonEngine: 3D Graphics Engine in C++ using OpenGL, glm, ImGui and Assimp.* GitHub. https://github.com/AlonsoCerpa/NeonEngine