PROYECTO ESTRUCTURAS DE DATOS

Daniel Rosas

Juan Forero Moreno
Santiago Camargo

Manuel Rincon

Componentes 1 y 2 completos y funcionales

 Objetivo del Proyecto: Crear un sistema para cargar, manipular y realizar cálculos sobre objetos tridimensionales.

Estructuras de Datos Utilizadas:

Listas

Vectores

Árboles KD

Funcionalidades Clave:

Creación de cajas envolventes Búsqueda de vértices cercanos Cálculo de rutas cortas

Estructura del Código

Modularidad: Diseño basado en funciones para escalabilidad y mantenimiento. 1.Bibliotecas Importantes:

<iostream>: Interacción con el usuario.

<fstream>: Lectura y escritura de archivos.

<vector>: Manejo de listas dinámicas.

Funciones Principales:

cargarArchivo: Carga datos de objetos.

envolvente: Calcula cajas envolventes.

encontrarObjeto: Busca objetos cargados.



COMPONENTE 1

Objetivo: Los algoritmos implementados en este componente servirán para gestionar la información

básica de los objetos 3D, a partir de los archivos de mallas. Este componente se implementará con las siguientes funciones:

comando: cargar nombre_archivo

comando: listado

comando: envolvente nombre_objeto

comando: envolvente

comando: descargar nombre_objeto

comando: guardar nombre_objeto nombre_archivo

comando: salir



COMPONENTE 2

Objetivo: Los algoritmos implementados en este componente servirán para identificar puntos (vértices) más cercanos en los objetos. Este componente se implementará con las siguientes funciones:

comando: v_cercano px py pz nombre_objeto

comando: v_cercano px py pz

comando: v_cercanos_caja nombre_objeto

Caja Envolvente

¿Qué es?

El menor cubo que contiene completamente un objeto.



- Búsqueda de límites mínimos y máximos.
- Comparación con coordenadas de vértices.
- Uso: Definir límites del objeto en los ejes X, Y, Z.





KDTREE.HXX

Recursivo

Busqueda de lado cercano

Busqueda nodo cercano

Distancia euclidiana

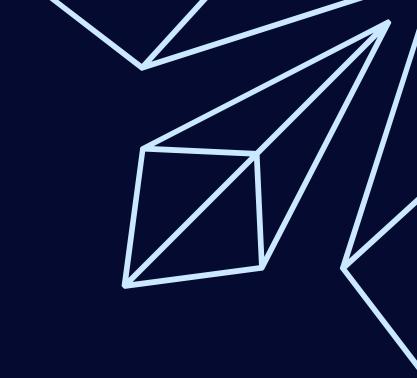


NODO.HXX

Almacena cordenadas de puntos Nombre del punto 2 hijos

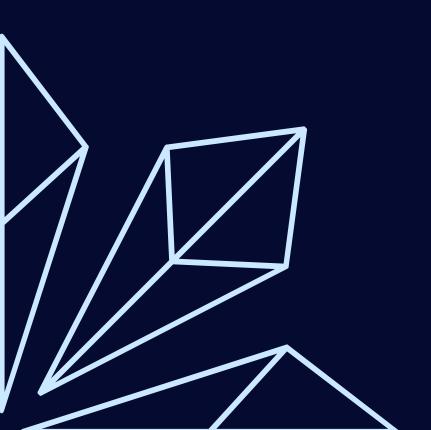


PROBLEMAS



Establecer los puntos en las cordenadas px,py,pz para los nodos del vertice cercano.

Solucion: Todos a la vez



Diseño Modular: Cada parte del sistema está distribuida en funciones independientes, lo que permite una fácil depuración y actualización del código.

El uso de estructuras como los vectores y los árboles KD garantiza que el sistema pueda manejar grandes volúmenes de datos tridimensionales de manera eficiente. El KD-Tree, en particular, permite realizar búsquedas rápidas en espacios multidimensionales

Manejo de Errores: Se implementan mecanismos robustos para la validación de archivos y el manejo de errores, garantizando que solo se procesen datos correctos y válidos. Esto protege al sistema de datos corruptos o mal formateados.

