



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.

1.

Funciones Principales:

- cargarArchivo: Carga datos de objetos.
- envolverte: Calcula cajas envolvertes.
- 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: envolver nombre_objeto

comando: envolver

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



KDTREE.HXX

Implementaciones de KDtree

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 coordenadas px,py,pz
para los nodos del vertice cercano.

Solucion: Todos a la vez

```
// Almacenar los vértices de todos los objetos en el Kd-Tree
for (std::list<Objeto>::iterator itObj = objetosPrograma.begin(); itObj != objetosPrograma.end(); itObj++) {
    std::string nombre_objeto = itObj->obtenerNombreObjeto(); // Obtener el nombre del objeto
    std::list<Cara> caras = itObj->obtenerCaras(); // Obtener las caras del objeto
    for (std::list<Cara>::iterator itCara = caras.begin(); itCara != caras.end(); itCara++) {
        std::list<Arista> aristas = itCara->obtenerListaAristas(); // Obtener las aristas de la cara
        for (std::list<Arista>::iterator itArista = aristas.begin(); itArista != aristas.end(); itArista++) {
            std::list<Vertice> vertices = itArista->obtenerListaVertices(); // Obtener los vértices de la arista
            for (std::list<Vertice>::iterator itVertice = vertices.begin(); itVertice != vertices.end(); itVertice++) {
                std::vector<double> punto = {
                    static_cast<double>(itVertice->obtenerX()),
                    static_cast<double>(itVertice->obtenerY()),
                    static_cast<double>(itVertice->obtenerZ())
                };
                kdTree.insertar(punto, nombre_objeto); // Insertar el vértice en el Kd-Tree
            }
        }
    }
}
```

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.



GRACIAS