

# OCTREES

Estructura de Datos para Partición Espacial  
Tridimensional.

# INTEGRANTES

Eliseo David Velasquez Diaz - 202410184

Jean Luka Terrazo Santiago - 202410422

# 4 PREGUNTAS CLAVE

1. ¿Qué problema resuelve?
2. ¿Cómo funciona?
3. ¿Cuales son las Operaciones y complejidades?
4. ¿Qué aporta la aplicación?

# REQUISITO 1

¿Qué problema resuelve?

# EL PROBLEMA

Es la búsqueda ineficiente en datos espaciales tridimensionales.



Búsqueda en 200,000 puntos 3D



¿Qué puntos están en esta región?



Método naive: revisar TODOS



Tiempo: 11 ms milisegundos para 200,000 puntos(muy lento)



En 60 FPS: solo 16 ms por frame

## ANALISIS



Si una consulta toma 11 ms = 68% del presupuesto total

# ¿POR QUÉ ES UN PROBLEMA?

MULTIPLES consultas por frame

Aplicaciones tiempo real:

- 4 consultas por frame
- $4 \times 11 \text{ ms} = 44 \text{ ms}$
- Resultado: 23 FPS ❌

Debería ser: 60 FPS ✅

# REQUISITO 2

¿Cómo funciona?

# OCTREE: División Espacial

- ✓ Divide espacio en 8 octantes
- ✓ Subdivisión recursiva
- ✓ PODA ESPACIAL:
  - Solo explora regiones relevantes
  - Ignora ramas irrelevantes

Resultado:  $O(\sqrt[3]{n})$  vs  $O(n)$

---



# DEMOSTRACIÓN

Código C++ funcionando en vivo 

# REQUISITO 3

¿Operaciones y complejidades?

# OPERACIONES PRINCIPALES

## 1 INSERCIÓN: $O(\log n)$

- Determinar octante (bits)
- Subdividir si excede umbral

## 2 BÚSQUEDA: $O(\log n)$

- Seguir camino desde raíz

## 3 RANGO: $O(\sqrt[3]{n} + k)$ ★

- Poda espacial (clave)
- $k$  = puntos encontrados

# ¿POR QUÉ $O(\sqrt[3]{n})$ ?

En 3D:

- Superficie del cubo  $\propto n^{2/3}$
- Solo exploramos superficie
- Interior se poda automáticamente

Resultado empírico:

- 200K puntos  $\rightarrow$  Speedup 9.2×
- Confirma teoría 

# TRACE PASO A PASO

Ejemplo: Insertar  $P3=(30,30,30)$

Centro:  $(50,50,50)$

$x=30 < 50 \rightarrow \text{bit } 2 = 0$

$y=30 < 50 \rightarrow \text{bit } 1 = 0$

$z=30 < 50 \rightarrow \text{bit } 0 = 0$

Octante =  $0 \times 4 + 0 \times 2 + 0 \times 1 = 0$



$P3 \rightarrow \text{octante } 0$

# REQUISITO 4

¿Qué aporta la aplicación?

# SISTEMA INTERACTIVO - 6 MODOS

1. Demo con visualización ASCII
2. Benchmark escalabilidad
3. Validación automática
4. Tests casos borde
5. Modo interactivo educativo
6. Demo completa

# LO CREATIVO



Modo interactivo:

- Insertar puntos en tiempo real
- Ver adaptación del Octree



Validación automática:

- Compara con búsqueda exhaustiva
- Garantiza correctitud



# RESUMEN

✓ Problema: Búsqueda  $O(n) \rightarrow O(\sqrt[3]{n})$

✓ Solución: Poda espacial

✓ Operaciones:  $O(\log n)$  y  $O(\sqrt[3]{n}+k)$

✓ App: Sistema interactivo 6 modos

✓ Demo: Speedup 9× confirmado