

Manual Técnico

Estructura de datos
proyecto 1

DESCRIPCIÓN DEL PROGRAMA

Se le solicita desarrollar un juego basado en un tablero tridimensional, representado mediante matrices ortogonales. Un jugador explora este tablero en busca de un tesoro oculto mientras enfrenta desafíos como enemigos, trampas y pistas que indican la cercanía del objetivo. Se implementarán estructuras de datos como árboles binarios para gestionar la información de los enemigos y su ubicación, además de registrar los movimientos del jugador.

Tecnologías usadas

Ide: Visual estudio code

lenguaje: c++

Estructuras usadas

- Cola
- lista enlazada
- árbol binarios
- matriz ortogonal

Algoritmos utilizados

matrizOrtogonal

- **método buscarOCrear:**

Cuando busca su complejidad sera de $O(n)$, ya que depende de los elemento que contiene la lista de encabezados.

Cuando inserta de ser necesario su complejidad es de $O(1)$.

- **Método insertarNodo:**

Cuando se inserta un nodo, cada while tiene una complejidad $O(n)$, ya que cada while recorre un n cantidad antes de insertar el nodo.

- **Mostrar matriz:**

Este método tiene una complejidad de $O(f*c)$, osea la multiplicación de filas por columnas.

- **Buscar nodo:**

Este método tiene una complejidad de $O(f*c)$, osea la multiplicación de filas por columnas, porque va buscando las columnas de cada fila.

ListaEnlazadaPistas

- **método insertarPista:**

En el mejor caso cuando la lista este vacía, es $O(1)$, pero cuando no el while recorre n nodos hasta llegar al final entonces es $O(n)$.

- **ImprimirLista:**

El while imprime n nodos, por lo tanto es $O(n)$.

EncabezadoCapa

- **buscarOCrear:**

El while busca n nodos, por lo tanto es $O(n)$.
En la creación o devuelta es $O(1)$.

ConfiguracionInicialPartida

- **crearPociones**
- **crearPistas**
- **crearTablero**
- **crearTrampas**
- **crearEnemigos**

Estos método su complejidad es $O(n)$, ya que se creara n elementos.

- **CrearJugador**
- **crearTesoro**

Estos métodos son de complejidad $O(f*c*ca)$, osea $filas*columnas*capas$, ya que en el peor caso pasaran por todos los nodos existentes.

ArbolBinario:

- **mostrarArbolRecursivo**

Este método en el peor de los caso cuando el árbol esta desequilibrado hace iteraciones, de espacios $0+1+2+\dots+(n-1)$ y esto es igual a $(n(n-1)) / 2$, que esto se traduce a n^2 .

Si el árbol esta equilibrado, tenemos que $\log_2 n$, osea supongamos como tenemos que cada nodo tiene 2 nodos, y n la cantidad de nodos, entonces tenemos $\log_2 n$, esto sera en promedio por cada nodo solo, ahora si se toma en cuenta los n nodos totales, tendríamos $O(n\log_2 n)$.

- **InsertarNodoRecursivo**

En el peor de los casos seria $O(n)$, cuando el árbol esta desequilibrado.

Cuando no seria solo $O(\log_2 n)$, ya que no recorre todos los nodos.