

Algoritmo A* en C++

1) Conceptos Importantes:

Cola de Prioridad (Heap): Es una estructura de datos la cual utiliza el concepto de árbol binario ordenado, es utilizado para la lista abierta, pues coloca el elemento de menor peso en la raíz del árbol y es sencillo llegar a él,

Lista Abierta: Lista de nodos por recorrer.

Lista Cerrada: Lista de nodos ya recorridos.

Heurística: Es una técnica o acercamiento aplicada para resolver problemas / acercase de una forma más cercana a la solución.

- Para este trabajo se utilizó la heurística de la distancia euclidiana entre dos puntos, en este caso del nodo al destino.

2) Implementación del algoritmo:

Todo el código está basado en el pseudocódigo disponible en Wikipedia, el cual funciona de la siguiente forma:

- Se toma el inicio o punto de partida.
- Se agrega a la lista abierta o por recorrer.
- Mientras la lista abierta no este vacía.
- Recorrer cada vecino del nodo de la lista abierta.
- Para cada nodo:
 - Si el nodo es una pared o ya se encuentra en la lista cerrada: Ignorar
 - Si la "h" del nodo sumado a la distancia del nodo y el vecino es menor que la "g" del vecino o si no está en la lista abierta, tomar como hijo del "nodo" padre y actualizar los valores de "f" ($g + h$).
 - Si no está en la lista abierta, se agrega.
- Una vez el nodo actual sea igual a la meta, se retorna el camino de la meta al inicio trazado por todos los nodos.

3) Justificación de eficiencia

Realizando la prueba, se puede observar que en promedio para cada laberinto de 1000x1000 se demora 0.03[s] aproximadamente.

Razones de por qué es eficiente:

- 1) Al usar una Heap, se evita buscar el elemento de menor costo.
- 2) Al usar una matriz de $n \times n$, de complejidad $O(n^2)$ en generarse, y que es utilizada solamente 1 vez al generar todo el laberinto, nos evitamos recorrer

nuevamente en cada iteración si el nodo está en la lista cerrada y abierta, ya que el costo para buscarlo es $O(1)$.

Dibujos Ilustrativos

#: WALL

blank_space: SPACE

X: ENTRY

G: EXIT

O: PATH

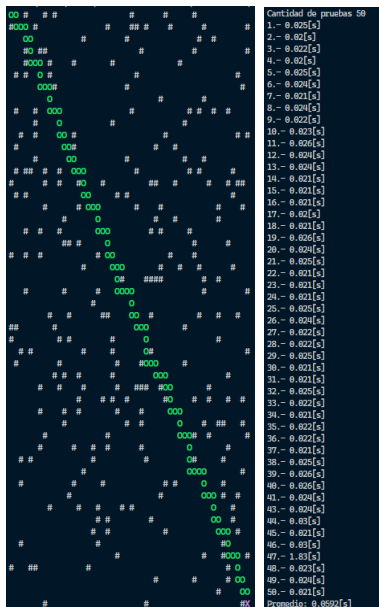


Imagen 1: Laberinto encontrando una solución

Imagen 2: Tiempo que tarda por cada laberinto de 1000 x 1000.

Fuentes:

Todo el código creado fue producto del pseudocódigo disponible en Wikipedia:

https://en.wikipedia.org/wiki/A*_search_algorithm

Y usando este artículos como referencia:

<https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2>

<https://advanceintelligence.wordpress.com/2014/10/07/algoritmo-a-estrella/>

Las clases Node, Container, son de autoría del docente de esta asignatura.

Para la heurística, distancia euclidiana:

https://en.wikipedia.org/wiki/Euclidean_distance