

PROYECTO PARTE 3. RESOLVER LABERINTOS O MAPAS USANDO BÚSQUEDA HEURÍSTICA

En esta parte del proyecto se deben resolver automáticamente laberintos o mapas usando algoritmos de búsqueda heurística.

Integrar a su sistema lo siguiente:

1. Permita elegir la medida de distancia para sus algoritmos de búsqueda heurística:
 - a. (10) Implemente la medida Manhattan
 - b. (10) Implemente la medida Euclidiana
2. Permita elegir el algoritmo de búsqueda con el que se resolverá el problema:
 - a. (22) Resuelva automáticamente el mapa usando el algoritmo de coste uniforme.
 - b. (22) Resuelva automáticamente el mapa usando el algoritmo de búsqueda voraz primero el mejor.
 - c. (21) Resuelva automáticamente el mapa usando el algoritmo A*.
 - d. Algoritmo de profundidad (que ya fue implementado en la segunda entrega del proyecto).
 - *Tome en cuenta que los algoritmos heurísticos evalúan todas las casillas(nodos) alcanzables y en caso de encontrar un nodo repetido que tiene mejor $f(n)$, se quedará con este nuevo nodo y eliminará el nodo repetido anterior. No se evaluarán nodos que ya hayan sido cerrados/visitados.*
 - *El usuario puede establecer la prioridad direccional para el algoritmo de búsqueda (mismo que fue implementado en la entrega 2 del proyecto). En caso de búsqueda heurística esta prioridad determina la elección de nodos empatados en costo. Note que este desempate se da en el mismo nivel, es decir, si hay nodos del nivel 3 con mismo costo que nivel 2, los primeros en entrar en la lista (nivel2) son los primeros en salir. En otras palabras, la prioridad direccional me desempata nodos con mismo costo en el mismo nivel. Para el caso de búsqueda ciega, esta prioridad determina el orden de visita de los nodos hijos.*
3. Al llegar al estado final debe mostrar en el mapa la ruta de solución encontrada (ya implementado en la segunda entrega).
4. (15) Al finalizar la búsqueda, se debe mostrar el árbol de búsqueda generado en donde se pueda observar el nivel de los nodos (con el fin de identificar nodos padres e hijos), nombre de los nodos, orden de visita de los nodos, estado del nodo (visitado o no), costo del nodo $f(n)$ (para el caso de A* separar por $g(n)$ y $h(n)$).
 - *Tome en cuenta que, aunque ya implementó árboles en la segunda entrega del proyecto, éstos no consideraban nodos repetidos. Es necesario verificar que su estructura le permite manejar los nodos repetidos con menor $f(n)$. Recuerde que, para el caso de los algoritmos de búsqueda heurística, puede suceder que el nodo ya existe en el árbol, pero debo cambiar la referencia de quién es su padre, es decir, el*

[illegible]

En el ejemplo, los nodos expandidos se encuentran desenmascarados, los nodos cerrados-visitados contienen el número de visita y cada nodo tiene entre paréntesis el costo total o $f(n)$. La "X" indica la posición actual del ser en el mapa, la "I" indica la casilla inicial y "F" la casilla final.