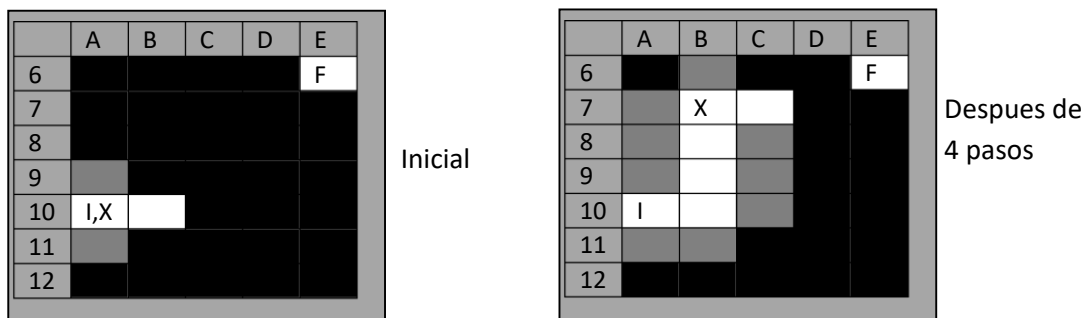


PROYECTO PARTE 2. ENMASCARAMIENTO, ÁRBOLES Y BÚSQUEDA CIEGA

Enmascarando mapas

Una cosa es el mapa realizado por el creador y otra es el conocimiento que tiene un ser dentro del mapa sobre éste. En realidad, el conocimiento del mapa que tiene un ser es solo la parte que sus sensores le permiten saber y el estado final.

Además considere que un ser X colocado dentro del mapa es capaz de memorizar el mapa previamente visitado y es capaz de ver solamente una celda de distancia (arriba, abajo, derecha e izquierda). Inicialmente, el ser no conoce el mapa entero y lo va descubriendo conforme viaja en él. Por ejemplo, en el siguiente mapa de laberinto, el mapa desconocido es marcado con negro.



a) (15) Agregue enmascaramiento de mapas a su interfaz.

Búsqueda ciega y árboles

En esta parte del proyecto se deben resolver automáticamente laberintos o mapas usando el algoritmos de búsqueda ciega, generar y mostrar su árbol correspondiente.

Integrar a su sistema lo siguiente:

1. (10) El usuario puede establecer la prioridad direccional para el algoritmo de búsqueda. Para el caso de búsqueda ciega, esta prioridad determina el orden de visita de los nodos hijos.
2. (20) Resolver automáticamente el laberinto o mapa utilizando el algoritmo de profundidad.
3. (5) Al llegar al estado final debe mostrar en el mapa la ruta de solución encontrada (note que la solución no es lo mismo que el recorrido llevado a cabo para encontrarla).
4. (50) 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, y costo (si el algoritmo es heurístico). Tome en cuenta que este árbol se puede ver como los que hacemos en pizarrón cuando resolvemos un problema con cualquier algoritmo de búsqueda, independientemente de si lo está graficando o mostrando como carpetas y archivos.

Note que:

- El árbol generado durante el proceso de búsqueda contiene TODOS los nodos expandidos, hayan sido visitados o no.
- Recuerde que cada casilla puede convertirse en un nodo del árbol, es decir, cada vez que el ser avanza en el mapa se crean nodos en el árbol dependiendo de cuáles son los nodos hijos alcanzables, es decir, que su costo de movimiento es diferente de N/A.
- Cuando el ser llega a una casilla, ésta se marca como visitada tanto en el mapa como en el árbol.
- No se repetirán nodos en el proceso de búsqueda.
- El árbol se guardará en memoria y será mostrado al finalizar un algoritmo o cuando se llegue a casilla final.
- Note que el usuario final no podrá agregar, eliminar ni modificar nodos, esto solo lo hace internamente el sistema cuando el ser se mueve en el laberinto.

Ejemplo:

A continuación se muestra un pequeño laberinto donde la casilla inicial es A,10 y la final es D,9. El orden de expansión de nodos es arriba, izquierda, abajo, derecha y no se repiten nodos.

Los números muestran el movimiento que realizó el ser en el laberinto con el algoritmo de profundidad.

El árbol de búsqueda final se muestra en la siguiente página.

	A	B	C	D	E	F
3						
4						
5						
6						
7		5	6	7		8
8		4		9		
9		3		F,10		
10	I,1	2				
11						
12						

