

PROYECTO PARTE 2. ENMASCARAMIENTO Y GESTIÓN DE ÁRBOLES

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	B	C	D	E
6					F
7					
8					
9					
10	I,X				
11					
12					

Inicial

	A	B	C	D	E
6					F
7		X			
8					
9					
10	I				
11					
12					

Despues de 4 pasos

- a) (25) Agregue enmascaramiento de mapas a su interfaz.

Árboles

(45) Para la siguiente entrega necesitarán la creación de árboles, por lo que deberá desarrollar una forma de:

- A. Agregar nodos
- B. Modificar información de nodos
- C. Eliminar nodos

Recuerde que cada casilla se convertirá en un nodo del árbol, es decir, cada vez que el ser avanza en el mapa se crea un nodo en el árbol. En ciertos casos (dependiendo del algoritmo) puede suceder que el nodo ya existe en el árbol pero debo cambiar la referencia de quién es su padre, es decir, el nodo puede “moverse” de lugar en el árbol generado. O también puede verlo como que elimino en el nodo anterior y creo uno nuevo con otro padre.

La información que debería guardar para cada nodo es:

- A. Padre
- B. Hijos
- C. Coordenadas (Fila y Columna)
- D. Número de visita
- E. Costo del nodo
- F. Cualquier otra información que considere relevante (ejemplo, si es inicial o final)

Note que el usuario final no podrá agregar, eliminar ni modificar nodos, ésto solo lo hace internamente el sistema cuando el ser se mueve en el laberinto.

(30) Por último, deberá agregar a la interfaz gráfica una manera de mostrar un árbol generado. (Por ejemplo, se puede mostrar el árbol como una estructura de carpetas y archivos, o pintarlo de manera gráfica con círculos y flechas).

Notas importantes:

La estructura de árbol es independiente del algoritmo y permitirá al sistema identificar un nodo cualquiera para modificar su información, agregarle uno o más hijos, o eliminarlo.

No existirán nodos repetidos en el árbol pero, dependiendo del algoritmo, si aparece un nodo repetido y éste tiene mejor función de evaluación, el nodo anterior se elimina y éste se agrega al árbol (o mueve de lugar el nodo).

Recordar que no se evaluarán o considerarán nodos cerrados, es decir que ya tengan todos sus hijos expandidos.

El árbol se guardará en memoria y será mostrado al finalizar un algoritmo o cuando se llegue a casilla final.

Para esta entrega, el ser avanzará con el teclado y al llegar a la casilla final se mostrará el árbol.

Recuerde que las casillas no alcanzables (es decir, con movimiento No aplica) NO son hijos del nodo actual y no deberán formar parte del árbol.

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 cerrados.

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						

