

PRÁCTICA 1

SIMULADOR

ALGORITMO A*

```
1 | | | | | | | | | | | |
2 | | | | | | | | | | | |
3 | | | | | | | | | | | |
4 | | | | | | | | | | | |
5 | | | | | | | | | | | |
6 | | | | | | | | | | | |
-----
      1      2      3      4      5      6
-----
1 | | | | | | | | | | | |
2 | | | | | | | | | | | |
3 | | | | | | | | | | | |
4 | | | | | | | | | | | |
5 | | | | | | | | | | | |
6 | | | | | | | | | | | |
-----

MOSTRAMOS LA LISTA CERRADA DEFINITIVA:
(5,2) (5,3) (4,3) (4,2) (6,4) (5,5) (4,6)
MOSTRAMOS LA LISTA ABIERTA DEFINITIVA:
(4,1) (5,1) (6,1) (6,2) (3,2) (3,3) (3,4)
RECORRIDO FINAL QUE SE LLEVA A CABO:
(5,2) -> (5,3) -> (6,4) -> (5,5) -> (4,6)
```

Objetivo, Lenguaje utilizado y detalles de implementación.

El objetivo principal de esta práctica era implementar el algoritmo A estrella en un determinado lenguaje de programación elegido por el alumno, donde el resultado final debe mostrar el trayecto que el dispositivo debe seguir para llegar a una determinada meta marcada inicialmente.

En mi caso, el lenguaje de programación que he utilizado es C++ junto con el IDE Visual Studio.

Mi algoritmo consiste principalmente en llevar paralelamente dos listas o arrays de estructuras, una abierta y otra cerrada. Estos arrays en cada posición contienen la fila y columna del nodo correspondiente que se va añadiendo a cada lista y además la fila y columna de su nodo padre. En el caso de la lista abierta también he contemplado el valor de $f(x)$ de cada nodo que es el valor estimado a la meta más el valor estimado que llevamos recorrido (valor hasta el inicio desde ese nodo).

Para ir rellenando la lista abierta, se van calculando las posibles expansiones del nodo actual en el que nos encontramos y estas posibles expansiones se añaden al array de lista abierta junto con su valor de $f(x)$ y el valor de su padre;

Según el valor de $f(x)$ que vaya teniendo cada nodo en las diferentes expansiones, el que se añade a la lista cerrada es el que menor valor de esta “ f ” tiene, que se incluye el nodo junto con su padre, de esta forma se va construyendo la lista cerrada y se va almacenando en el array.

Cuando el algoritmo encuentra que una de sus posibles expansiones es la meta, es cuando se sale de bucle y termina el programa.

Para mostrar el camino final del dispositivo, vamos recorriendo desde la meta los padres que van teniendo los diferentes nodos, es decir, se coge la meta de la lista cerrada y se visualiza su padre, se muestra su padre, y ahora el nodo actual es este padre, que mostrara a su vez su padre y así sucesivamente. (Es por decirlo de alguna forma una llamada recursiva que va mostrando el padre del nodo hasta que encuentra el nodo inicial).

Para poder mostrar estos valores de los nodos que va recorriendo el dispositivo, he copiado los valores de la lista cerrada a un array denominado array final que es el encargado de mostrar el camino que ha seguido el dispositivo, este array contiene los valores de la lista cerrada junto con el de los padres y solamente se encarga de mostrar estos valores. (He implementado este array por si acaso en las partes opcionales tenía que tocar la lista cerrada para tener guardados los datos sin perderlos).

Partes Opcionales

En lo relativo a las partes opcionales, únicamente he dispuesto de tiempo para implementar la parte en la que obligamos al dispositivo a pasar por un punto intermedio antes de llegar a la meta, de tal forma que el programa nos pedirá al comienzo la fila y columna de la que queremos iniciar el recorrido, la fila y la columna por la que obligamos al dispositivo a pasar, y la fila y columna de la meta.

Nota: Esta parte esta implementada y aunque funciona correctamente, me he dado cuenta de que en algún caso el resultado no es del todo correcto o el programa falla, aunque en la mayoría de las situaciones que he probado su funcionamiento es el adecuado.

Manual de Usuario

Para poder ejecutar el programa hay que descomprimir el proyecto en la carpeta raíz de proyectos de V.S. Dentro de Visual Studio pincharíamos archivo->abrir proyecto y seleccionamos el proyecto que hemos guardado. (Dentro del proyecto se incluye un fichero de texto llamado "autor.txt" que incluye el texto que se muestra en el menú en la opción 3, de tal manera que si queremos ejecutar la opción 3 del menú, debemos asegurarnos que este fichero está dentro del proyecto.

El simulador comienza con un menú en el que se nos muestran las siguientes opciones:

- 1- Simulador Algoritmo A* con tablero*
- 2- Simulador Algoritmo A* con paso intermedio*
- 3- Autor y aplicaciones en la vida real*
- 4- Salir*

1-La primera opción es para simular el algoritmo A estrella normal sin pasos por puntos intermedios. El programa comenzará pidiéndonos la fila y columna donde queremos colocar la salida del dispositivo, posteriormente nos pedirá donde queremos colocar la meta, y finalmente si queremos poner obstáculos en el camino. En el caso de que queramos poner obstáculos, nos irá pidiendo la fila y columna donde queremos ponerlos.

```

Elige una opcion:
1- Simulador algoritmo A* con tablero
2- Simulador algoritmo A* con paso por un punto intermedio
3- Autor y aplicaciones en la vida real
4- Salir
1
Donde colocamos el punto de partida?:
Fila: 1
Columna: 1
Donde colocamos la meta?:
Fila: 5
Columna: 5
-->Quieres poner obstaculos? (S/N)

s
En que casilla quieres poner el obstaculo?
Fila: 3
Columna: 3
¿Quieres poner mas obstaculos? (S/N): n
    
```

En el momento en el que pulsamos la tecla “n” de que no queremos poner más obstáculos, nos mostrara todas las expansiones en el tablero que el dispositivo ha tenido durante el recorrido, estas expansiones están indicadas en el tablero con la letra “c” que es la letra que simula al dispositivo. En el tablero estará también la letra “m” que simboliza la meta, y las letras “X” que haya en el tablero indican los obstáculos que hay en el tablero y por donde no puede pasar el dispositivo.

¿Quieres poner mas obstaculos? (S/N): n

	1	2	3	4	5	6
1	c					
2						
3			X			
4						
5					m	
6						

	1	2	3	4	5	6
1						
2		c				
3			X			
4						
5					m	
6						

	1	2	3	4	5	6
1						
2			c			
3			X			
4						
5					m	
6						

	1	2	3	4	5	6
1						

Por último, el programa mostrará como ha quedado la lista abierta al finalizar el programa, y la lista cerrada, y posteriormente nos mostrará el recorrido que ha seguido el dispositivo para llegar a la meta. (los nodos por los que ha pasado para llegar hasta la meta)

```
C:\Users\usuario\documents\visual studio 2013\Projects\AlgoritmoA\Debug\AlgoritmoA.exe

MOSTRAMOS LA LISTA CERRADA DEFINITIVA:
(1,1) (2,2) (2,3) (3,2) (3,4) (4,4) (5,5)

MOSTRAMOS LA LISTA ABIERTA DEFINITIVA:
(1,2) (2,1) (1,3) (3,1) (1,4) (2,4) (4,1) (4,2) (4,3) (2,5) (3,5) (4,5) (5,3) (5,4)

RECORRIDO FINAL QUE SE LLEVA A CABO:
(1,1) -> (2,2) -> (2,3) -> (3,4) -> (4,4) -> (5,5)
```

2- La segunda opción nos da la posibilidad de ejecutar el algoritmo A pasando por un punto intermedio que elijamos nosotros. Al comienzo el programa igual que antes nos pedirá la fila y columna de salida del dispositivo, la fila y columna de meta del dispositivo, y la fila y columna del paso intermedio que queremos hacerle pasar, una vez hemos introducido los datos, se ejecuta el algoritmo estrella dos veces, uno hasta el punto de paso y otro hasta la meta desde este punto de paso y nos muestra el resultado que ha seguido para llegar hasta la meta. Después de mostrarnos este recorrido nos muestra el tablero con las diferentes letras el cual simbolizan:*

- “m” meta del dispositivo
- “p” punto por el que tiene que pasar el dispositivo
- “c” ultimo nodo expandido antes de llegar a la meta y la otra letra “c” es el punto de partida de donde salió el dispositivo

C:\Users\usuario\documents\visual studio 2013\Projects\AlgoritmoA\Debug\AlgoritmoA.exe

```
Elige una opción:
1- Simulador algoritmo A* con tablero
2- Simulador algoritmo A* con paso por un punto intermedio
3- Autor y aplicaciones en la vida real
4- Salir
2
Donde colocamos el punto de partida?:
Fila: 1
Columna: 1

Donde ponemos el punto por el que tiene que pasar el dispositivo?
fila: 3
columna: 3

Donde ponemos la meta del dispositivo?
fila: 5
columna: 5
-->Quieres poner obstaculos? (S/N)
n
RECORRIDO FINAL QUE SE LLEVA A CABO:
(1,1) -> (2,2) -> (3,3) -> (4,4) -> (5,5)

  1   2   3   4   5   6
-----
1 | c |   |   |   |   |   |
2 |   |   |   |   |   |   |
3 |   |   | p |   |   |   |
4 |   |   |   | c |   |   |
5 |   |   |   |   | m |   |
6 |   |   |   |   |   |   |
-----
```

3- La tercera opción es un complemento que nos muestra algunas aplicaciones en la vida real del algoritmo, en qué consiste, y el nombre del autor del programa. (Este apartado u opción del menú es meramente informativo y como complemento a este manual de usuario proporcionado).

Conclusión

He de decir que me ha llevado bastante tiempo programar este algoritmo y durante su desarrollo, aunque al comienzo de empezar a programarlo no parecía tan complicado, en la medida en la que iba avanzando me iban surgiendo más y más problemas y me costaba más resolverlos ya que aumentaba su complejidad, por ejemplo, cuando fallaba uno de los nodos intermedios de las expansiones que no guardaba bien el padre, o no hacía una correcta expansión...etc.