

IMPLEMENTACIÓN DEL ALGORITMO A*

Por Guillermo Delgado Yepes



CONTENIDO

1. INTRODUCCIÓN	3
2. DETALLES DEL FUNCIONAMIENTO	3
3. DETALLES DE LA IMPLEMENTACIÓN	4
3.1 MODO CON ALTURAS	4
3.2 PENALIZACIONES.....	5
4. MANUAL DE USUARIO	6
4.1 MENÚ DE BOTONES	8

1. INTRODUCCIÓN

El algoritmo A* es un algoritmo de búsqueda clasificado dentro de los Algoritmos de búsqueda en grafos. Utiliza el coste real del recorrido y el valor heurístico de los nodos. Se realiza usando la siguiente función:

$$f(n) = g(n) + h(n)$$

Donde:

- $h(n)$ es el valor heurístico que se calcula de la siguiente manera

$$\sqrt{(x_a - x_m)^2 + (y_a - y_m)^2}$$

Donde:

- x_a : Es la fila del nodo actual
- x_m : Es la fila del nodo meta
- y_a : Es la columna del nodo actual
- y_m : Es la columna del nodo meta
- $g(n)$ es el coste de ir hasta ese nodo si te mueves arriba, abajo, izquierda y derecha o ir por diagonales:
 - 1 si vas arriba, abajo, izquierda o derecha
 - $\sqrt{2}$ si vas por diagonales

2. DETALLES DEL FUNCIONAMIENTO

En la primera interfaz gráfica se seleccionan el número de filas y el número de columnas para crear el tablero. En la segunda interfaz gráfica se introducen los diferentes elementos del tablero, se rellena la matriz, se realiza el algoritmo y se muestra el camino.



3. DETALLES DE LA IMPLEMENTACIÓN

El lenguaje para la implementación ha sido Java. Las interfaces gráficas se han desarrollado en Java Swing, la primera (introducción de fila y columna) se ha realizado en NetBeans, la segunda (tablero y botones) se ha realizado a mano.

Se ha creado una matriz de Celdas (que heredan de JLabel) y que contienen un enumerado que identifican de que tipo son (INICIO, LIBRE, BLOQUEADO, FINAL, PENALIZACIÓN y SAVEPOINT), la información de la fila y la columna que ocupa dicha celda dentro de la matriz, así como su altura y penalización.

Se usa un transfer para pasar el tamaño de una vista a otra, y un transfer para comunicar la vista del tablero con el algoritmo de búsqueda.

También se ha creado una clase Nodo, que contiene las coordenadas específicas de la casilla dentro de la matriz, su nodo padre, el nodo destino, y el cálculo del coste.

La clase Búsqueda recibe un transfer con la matriz y todos los datos necesarios y crea dos ArrayList<Nodo>, uno para guardar la lista de Nodos Abiertos y otra para guardar los cerrados. Se crea una función para introducir los nodos en la posición correcta de la lista de Abiertos. En esta clase se realiza el algoritmo, se devuelve null si no hay camino, o un ArrayList<Coordenadas> si existe un camino.

3.1 MODO CON ALTURAS

El modo con alturas se realiza de la siguiente manera:

1. El usuario introduce en el cuadro de texto la altura máxima por la que Mario puede pasar. Por ejemplo 2000, Mario no podrá pasar por casillas cuya altura sea mayor a 2000.
2. Se le asigna a cada celda una altura máxima de forma aleatoria dentro de un intervalo de valores que se calcula de la siguiente manera:
 - a. La altura mínima del intervalo será la mitad de la altura introducida por el usuario. En este caso 1000. La razón de este cálculo es que no queremos que haya celdas cuya altura sea 0.



- b. La altura máxima del intervalo será la altura introducida por el usuario + la cuarta parte de la altura introducida por el usuario, es decir, $2000 + \frac{2000}{4} = 2500$. La razón de este cálculo es porque deseamos que haya una mayor proporción de celdas que sean inferiores a 2000.
3. Se inicia el algoritmo teniendo en cuenta las alturas y se muestra el camino por pantalla.

3.2 PENALIZACIONES

La penalización de cada celda es un número aleatorio dentro del intervalo que se calcula de la siguiente manera:

- La penalización máxima es el cálculo de la distancia máxima (desde la esquina superior hasta la esquina superior), por ejemplo, para una matriz de 10x10 sería $\sqrt{10^2 + 10^2}$. A este cálculo se le hace el 80%.
- La penalización mínima es la mitad de la penalización máxima.

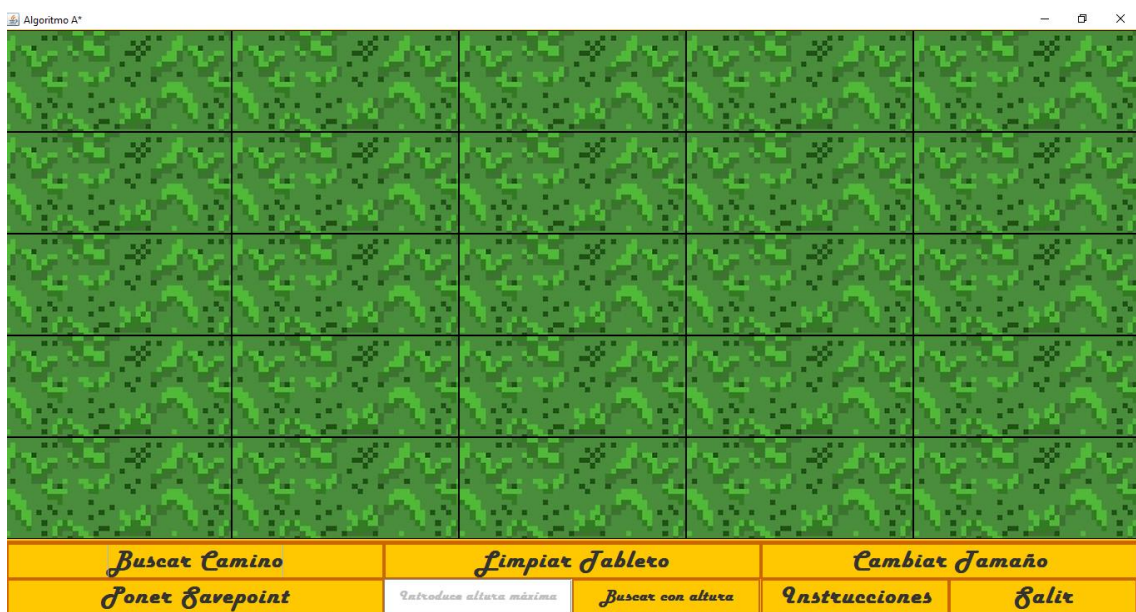


4. MANUAL DE USUARIO

Para ejecutar la aplicación hay que hacer doble click sobre el archivo .jar. Se abrirá la siguiente ventana:



Introduce el número de filas y el número de columnas en los recuadros correspondientes y pulsa el botón "Crear Matriz" para que se pinte el tablero inicial. Si todos los datos son correctos te aparecerá una ventana como esta:



Para colocar los distintos elementos se deberá pulsar encima de la casilla deseada.

Estos son los elementos en orden de aparición:

- **Inicio:** La casilla de inicio está representada por Mario. Será el primer elemento en aparecer (siempre que no este puesto ya, solo puede haber un inicio).



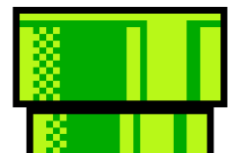
- **Meta:** La casilla de meta está representada por la Princesa Peach. Será el segundo elemento en aparecer (siempre que no este puesto ya, solo puede haber una meta). Será el primer elemento en aparecer si el inicio ya está puesto).



- **Bloqueo:** La casilla de bloqueo está representada por Bowser. Será el tercer elemento en aparecer (siempre que inicio y meta no estén puestos, será el segundo si está puesto uno de los dos y será el primero si ambos están puestos). Se pueden poner tantos como se quiera.



- **Penalización:** La casilla de penalización está representada por una tubería con planta piraña. Como Mario tiene que esperar a que la planta se esconda, tardará más tiempo al pasar por esas casillas. Aparecerán después de los bloqueos. Se pueden poner tantas como se quiera.



- **Savepoint:** Estas son casillas por las que Mario deberá pasar antes de llegar a la meta. Están representadas por los castillos. Para que aparezcan se debe pulsar el botón "Poner Savepoints" del panel inferior (el botón se pondrá de color rojo cuando esté activado), desde ese momento podrás poner todos los Savepoint que quieras en el orden por el que quieras que pase Mario.



¡¡CUIDADO!! No puedes borrarlos ni añadir más después. Vuelve a pulsar dicho botón para dejar de poner Savepoints y volver a los elementos

normales. Si lo vuelves a activar, se borrarán todos los Savepoints que pusiste anteriormente.

A continuación, pulsa el botón “Buscar camino” para que te encuentre el mejor camino posible.



4.1 MENÚ DE BOTONES

1. **Buscar Camino:** Busca el camino desde el inicio hasta la meta sin tener en cuenta las alturas de las celdas.
2. **Limpiar tablero:** Quita todas las casillas ocupadas y caminos pintados
3. **Cambiar tamaño:** Abre la ventana para cambiar la altura y anchura de la matriz
4. **Poner Savepoint:** Al activar esta acción puedes poner todas las metas que quieras en el tablero
5. **Poner alturas:** Introduce la altura máxima por la que puedes pasar e inicia el modo con alturas.
6. **Instrucciones:** Muestra las instrucciones
7. **Salir:** Salir de la aplicación