Práctica 1

Algoritmo A*

Ingeniería del Software

UCM

Andrei Ionut Vaduva

Lenguaje utilizado, procedimientos utilizados, partes opcionales y detalles necesarios.

Para la realización de esta práctica se ha utilizado Java, un lenguaje orientado a objetos. El código de esta práctica se ha estructurado en tres paquetes:

- main
    - Main
- dataStructures
    - Cell
    - Data
    - WayPointStructure
- aStar
    - Board
    - Star

El primer paquete corresponde al main, que es el que se encarga de instanciar las clases Board y Star.

El segundo tiene tres clases que corresponden a estructuras para almacenar información. Cell guarda información de la celda, es decir guarda la distancia estimada, la distancia acumulada, el color de la celda ya que puede ser un obstáculo, o una casilla con un coste mayor etc. La estructura Data se utiliza en el caso de las búsquedas, al buscar en abierta y en cerrada, se consideró oportuno tener una estructura que devolviera true si la celda se encontraba y la celda en cuestión o bien en caso contrario false y null. En el caso de WayPointStructure contiene una lista de celdas, un punto de inicio y un puto final. Esta estructura se utiliza en una de las partes opcionales y su uso esta relacionado con la visualización de los diferentes caminos en el tablero. No se utiliza en el algoritmo principal.

El tercer paquete y más importante contiene el tablero (Board), que se encarga de mostrar por pantalla una matriz, un panel de configuración y de

recibir los eventos o instrucciones para ejecutar el algoritmo A*. La clase Star contiene la implementación del algoritmo en si que consiste en:

Se escogen las celdas que corresponden a las posiciones de inicio y fin y se comprueba que no sean obstáculos. Si no son obstáculos se calculan las estimaciones y los diferentes atributos de la clase Cell para esa celda de inicio en concreto y se añade mediante un método privado a la lista abierta.

Dicho método se encarga de mirar si esa celda ya esta en abierta, en cuyo caso no la añadirá, y en caso de no estar añade dicha celda a la lista y ordena la lista por la distancia estimada + distancia acumulada.

Una vez añadida el algoritmo entra en un bucle que no parará hasta que no se produzca un fallo o se encuentre el final.

El algoritmo a gran escala lo que hace es, introducir en la lista abierta el nodo actual y si no es el nodo final se introducen sus hijos, es decir las celdas adyacentes en la matriz, sin considerarse a sí mismo, obstáculos, padres o si existe en cualquiera de las dos listas, abierta o cerrada.

El algoritmo funciona de tal manera que, si hay un camino mejor, ese será el camino elegido ya que se irán introduciendo nodos tanto en la lista abierta como cerrada de forma que los nodos que hay que evaluar están en la lista abierta y los que ya han sido evaluados y elegidos estarán en la lista cerrada.

Al acabar este bucle, ya sea por fallo o porque se ha encontrado el final, se obtiene una lista cerrada que contiene una serie de nodos cuyos padres han sido modificados o no en función de las circunstancias para encontrar el camino óptimo.

Para poder mostrar este camino (el camino bueno, el óptimo) se realiza una búsqueda recursiva a partir del ultimo nodo de la lista cerrada, de manera que recorriendo hacia atrás los padres de ese nodo se llega al inicio. Los nodos recorridos se almacenan en una lista auxiliar.

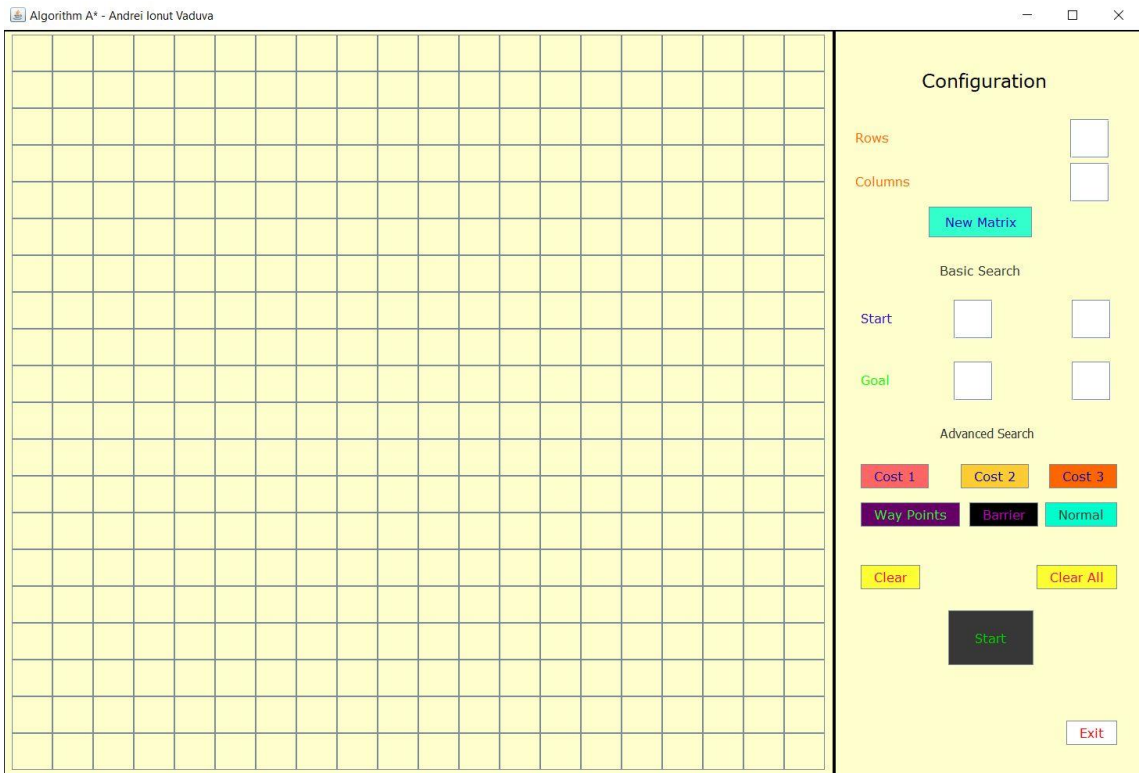
Dicha lista auxiliar es recibida por el tablero que se encarga de pintar el camino.

El código ejecutable entregado contiene dos partes opcionales:

1. Way Points: Lo que se hace en este caso es que se establece un punto de inicio y el resto de los puntos son considerados metas parciales, de forma que el algoritmo concluye cuando todas las metas parciales han sido alcanzadas. En este modo, el algoritmo NO contempla obstáculos, pero si contempla costes.
2. Costes: Se han definido tres tipos de costes. Cada coste esta identificado por un color. Cuanto mas oscuro es el color mayor es el coste. Coste1<coste2<coste3. Los valores de estos costes actualmente están en:
    a. Coste1 añade una penalización de +2;
    b. Coste2 añade una penalización de +4;
    c. Coste3 añade una penalización de +6;

Al ser una aplicación de escritorio, la creación de la parte visual consume bastantes recursos, crear una matriz mayor de 150x150 supone esperar cierto tiempo extra. Esta práctica esta optimizada para matrices de no mas de 100 o 150 filas y columnas. El algoritmo funciona para matrices de cualquier dimensión, pero la parte visual no. Si la aplicación se queda colgada o hace cosas raras, hay que reiniciar ya que los errores se deben a la interfaz gráfica.

Manual de usuario



Al iniciar la aplicación se abre una ventana que contiene en la parte derecha una matriz, donde se configurarán los diferentes casos, se mostrará el camino etc.

A la derecha tenemos un panel de configuración con varias opciones.

- Redimensionar matriz: Es la primera opción del panel de configuración. Simplemente hay que introducir la nueva dimensión y darle al botón New Matrix.
- Basic Search: Es el A* básico. Consiste en introducir un punto de inicio y un punto de final y pulsar el botón de Start.
- Advanced Search: Hay diferentes opciones.
    - Si no tocamos nada del panel de configuración y hacemos click en cualquier celda de la matriz se va a establecer esa celda como punto de inicio. Si hacemos otro click se establecerá la siguiente celda como punto de finalización. Si se siguen haciendo clicks a continuación, las celdas serán marcadas como obstáculos. Una vez establecidos el punto de inicio y fin

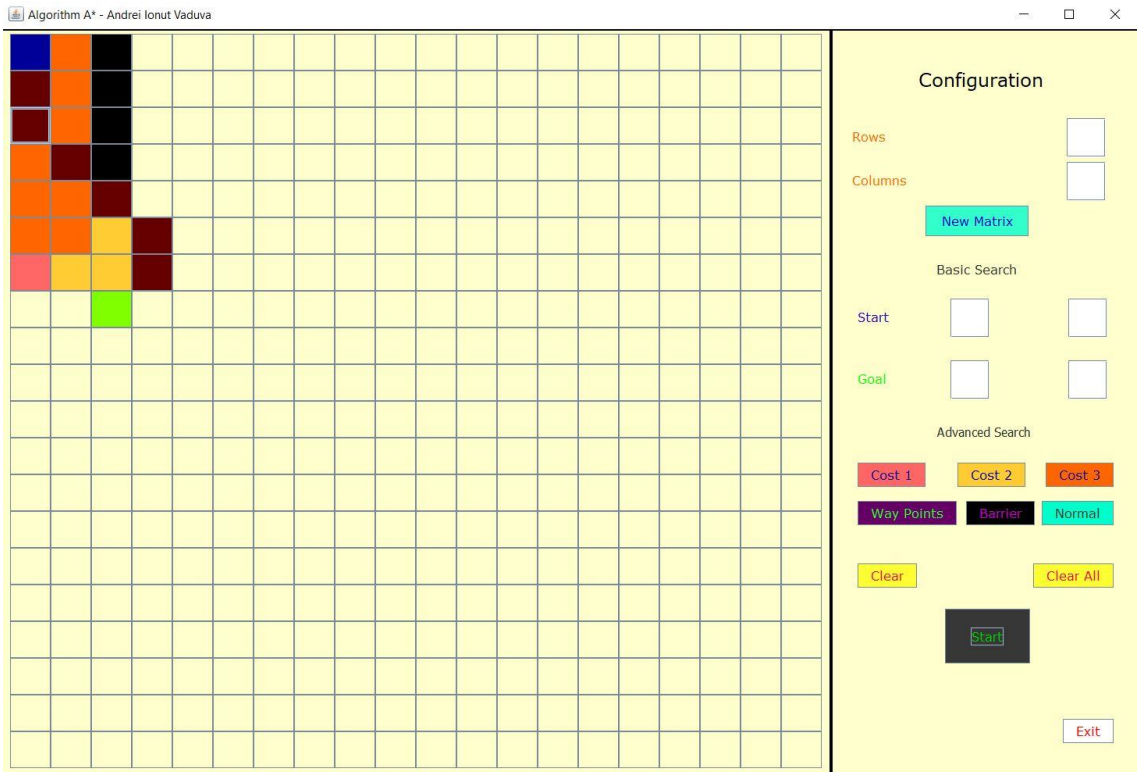
y los obstáculos en caso de haberlos hay que pulsar el botón de Start.

- En el panel de configuración disponemos de una fila con tres botones llamados “Coste 1”, “Coste 2” y “Coste 3”. Cada color tiene un color de fondo diferente. Dicho color será el que aparecerá en la matriz. Al hacer click en “Coste 1” y después click en la matriz, la celda del click quedará marcada como una celda que tiene coste 1 y se pondrá de su correspondiente color. Lo mismo para “Coste 2” y “Coste 3”. Todos los clicks realizados una vez hecho click en cualquiera de los tres botones solo tendrán esa funcionalidad, poner costes.

  Si ya hemos puesto todos los costes en las celdas elegidas y querremos establecer un inicio y un final hay que pulsar en el botón “Normal”, que cambiará el modo de los clicks al de la funcionalidad básica de: un click marcamos la celda como inicio, otro click marcamos la celda como fin y los siguientes clicks obstáculos.

- Si por el contrario, solo querremos poner obstáculos hay que hacer click en el botón con fondo negro “Barrier”.
- El botón con fondo morado, “Way Points” al hacer click sobre él la aplicación se pone en modo Way Points, es decir el primer click establece el punto de partida y los siguientes clicks serán metas parciales. Para comenzar a buscar el camino siempre hay que pulsar el botón de “Start”.
- Una vez realizado un caso es muy importante utilizar los botones “Clear” o “Clear All”.
- El botón Clear All borra todo el tablero incluido obstáculos (casillas negras).
- El botón “Clear” borra todo a excepción de los obstáculos. (Las casillas de coste si las borra).
- El botón de “Exit” cierra la aplicación.

También hay que tener en cuenta que el camino encontrado se pinta de color marrón y si una celda del camino tiene cierto coste, su color cambiará al color del camino al ser atravesada, a excepción de los obstáculos que no serán atravesados bajo ningún concepto.



En la imagen de arriba podemos observar celdas marcadas de diferente color. A continuación, vamos a asociar cada color a un caso.

- Azul: Punto de inicio.
- Verde: Punto de finalización.
- Negro: Obstáculos por los que no se puede pasar.
- Naranja: Casillas que al ser atravesadas tienen un coste mayor que el normal o que el coste 2.
- Amarillo raro: Casillas que al ser atravesadas tienen un coste mayor que el normal o que el coste 1.
- Rosa: Casillas que al ser atravesadas tienen un coste mayor que el normal.
- Marrón: Son las casillas que forman el camino desde el inicio azul hasta el final verde. El inicio y el fin no se marcan en marrón.