Práctica de Ingeniería del Conocimiento

Algoritmo A*

Rodrigo de Miguel González

Universidad complutense de Madrid
Facultad de informática

Índice

1. Introducción
2. Funcionamiento
3. Detalles de funcionamiento
4. Detalles de implementación
5. Manual de usuario
6. Bibliografía

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.
Presentado por primera vez en 1968 por Peter E. Hart, Nils J. Nilsson y Bertram Raphael.
Se basa en una función de evaluación:

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

Donde:

- *h(n)* es la distancia (coste) al nodo n (actual) y
- g(n) es la distancia (coste) al nodo meta.

Además hace uso de dos listas de Nodos, una abierta y otra cerrada. En la abierta se guardan los nodos potenciales para el camino óptimo y la cerrada contiene los Nodos ya utilizados.

Funcionamiento

Su funcionamiento simplificado es el siguiente:

Se pretende implementar una versión simplificada como sigue:

1. Colocar el nodo de comienzo en la lista ABIERTA y calcular la función de coste f(n), siendo ahora g(n) = 0 y h(n) la distancia entre la posición actual y la meta.
2. Los obstáculos se incluyen directamente en la lista CERRADA.
3. Eliminar de la lista ABIERTA el nodo con la función de coste de mínimo valor y colocarla en CERRADA. Este es el nodo n. Si se produce empate entre dos nodos elegir uno de ellos aleatoriamente. Si uno de los nodos en ABIERTA es el nodo meta, entonces seleccionarlo, recuperar los punteros de los antecesores y generar la solución. Actualizar el coste para alcanzar el nodo padre con el fin de poder calcular posteriormente la nueva función de coste h(n). En caso contrario continuar en el punto 4.
4. Determinar todos los nodos sucesores de n y calcular la función de coste para cada sucesor que NO está en la lista CERRADA.
5. Asociar con cada sucesor que NO está ni en ABIERTA ni CERRADA el coste calculado, f(n), y poner esos sucesores en la lista ABIERTA. Asignar punteros a n (n es el nodo padre).
6. Con cualquiera de los sucesores que ya estaban en ABIERTA calcular el menor coste de entre el que ya tenía y el recién calculado: min(nuevo f(n), viejo f(n)).
7. Ir al paso 3

Detalles de funcionamiento

En la interfaz gráfica se seleccionan los parámetros (lugar de los obstáculos, el inicio y fin del camino o una nueva dimensión para el tablero), estos se envían a al mapa y posteriormente a la clase Algoritmo_A_Estrella, que se encarga de buscar el camino más corto; de introducir datos erróneos aparecerá un mensaje de error.
Además se calcula el tiempo que tarda el algoritmo en ejecutarse.

Detalles de implementación

El lenguaje para la implementación ha sido Java 7.0. La interfaz gráfica está desarrollada con Java Swing realizada a mano por la sencillez de la misma.

La interfaz gráfica se inicia con un tablero de botones de 15x10 (que se podrá redimensionar, ir al manual de uso). La ventana se lanza en un thread y a la hora de pintar el camino se realiza en otro Thread distinto para que que quede más dinámico. Tiene un ligero delay para que no se pinte demasiado rápido y pueda apreciarse.

Para el algoritmo he usado los arrays dinámicos ArrayList<Nodo> para la listas Abierta, Cerrada, y el camino final.

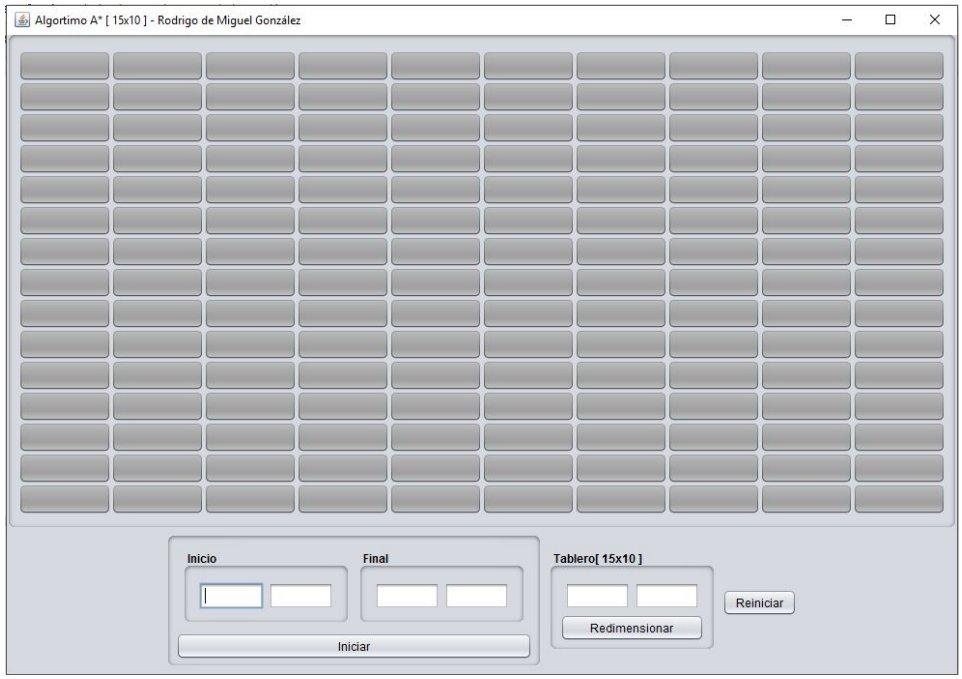
Cada Nodo tiene las coordenadas de la “casilla” del tablero que es, así como del tipo y las casillas circundantes

La clase Algoritmo_A_Estrella contiene el tablero lógico (con los obstáculos) de la aplicación y las listas y la Interfaz Heurística.

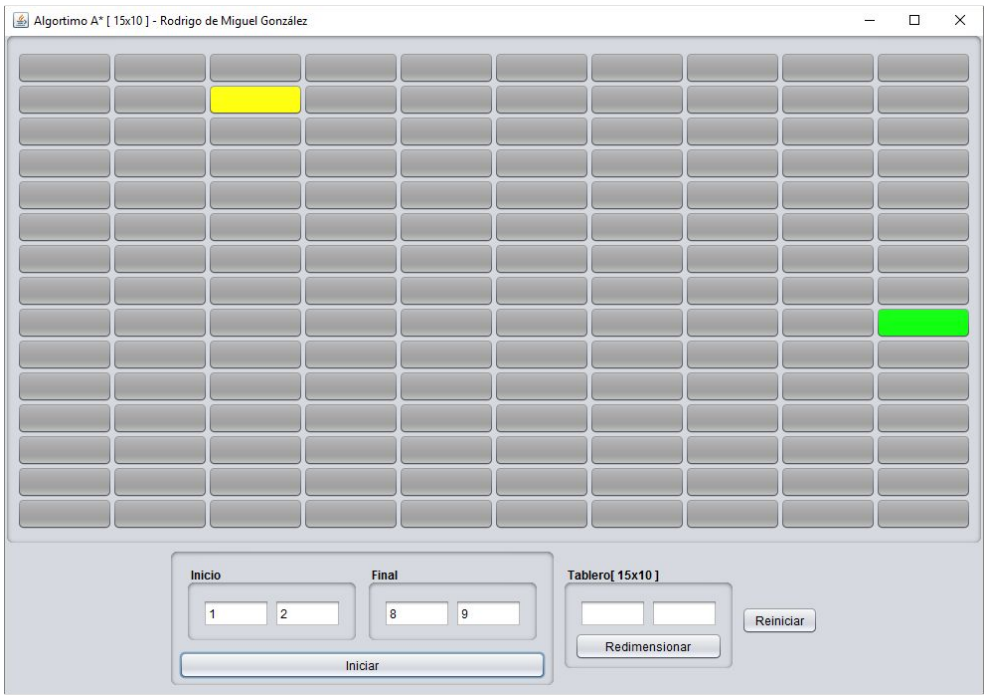
El cronómetro del algoritmo se inicia cuando se inicializan los datos del tablero lógico y acaba cuando se obtienen los resultados, el tiempo no incluye en retardo de comprobaciones de datos correctos ni de pintado de la interfaz.

Manual de usuario

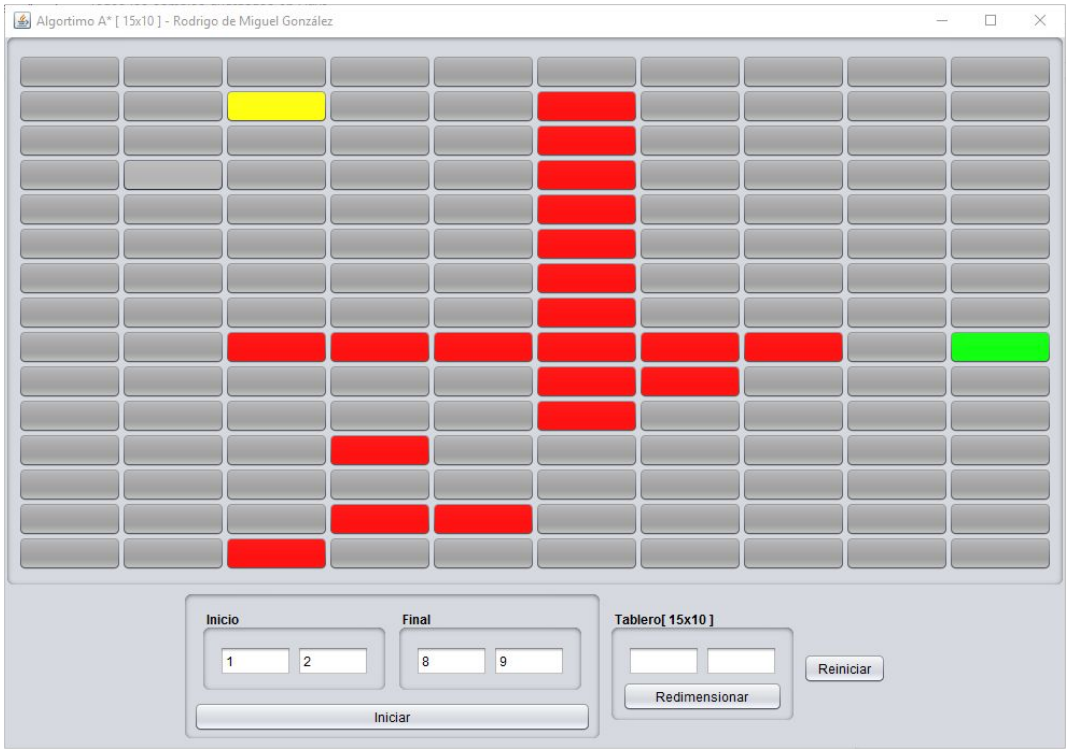
Para ejecutar la aplicación hay que hacer doble click sobre el archivo .jar. Se abrirá una ventana con el tablero y las opciones disponibles:



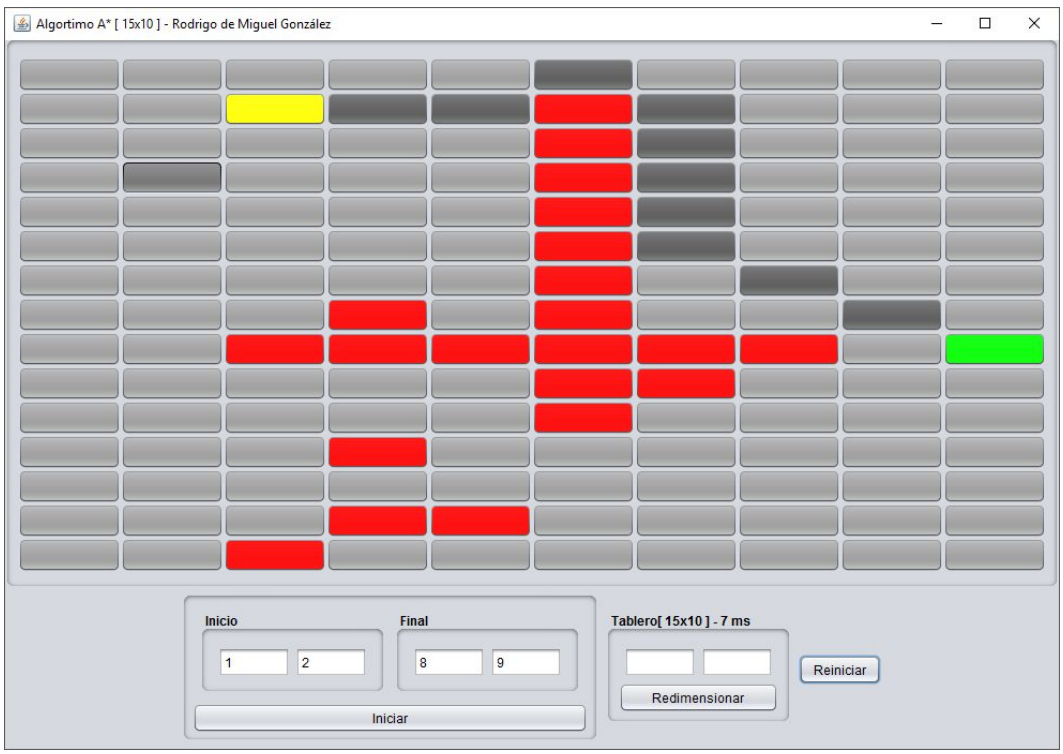
En la parte de abajo de la ventana se pueden meter las coordenadas de inicio y de final del camino, una vez metidas las dos coordenadas de cada uno se pintará en en el tablero las casillas correspondientes:



Posteriormente hacer click sobre las casillas del tablero para ir colocando los obstáculos



Una vez terminado hacer clic sobre el botón inicio o pulsar la tecla Enter.
El algoritmo se ejecuta y comienzá a pintarse el camino más corto.



Una vez terminado puede volver a colocar nuevos obstáculos y darle a iniciar, de esta forma se calculará un nuevo camino y se volverá a pintar.
Pulsar el botón reiniciar para reiniciar la ventana

En la caja de tamaño se puede indicar un nuevo tamaño de tablero. Así como ver el tiempo en milisegundos que ha tardado el algoritmo en ejecutarse.

Bibliografía

- Apuntes de la asignatura
- Wikipedia
- Cristalab