Inteligencia Artificial- Práctica Metro Praga

María Díez Sanjuán (s100044)

Sandra Gómez Yágüez (t110057)

Rafael Escribano Romero (s100045)

Eduardo Quiñones Matesanz (r090034)

El objetivo de esta práctica es diseñar una aplicación para hallar el trayecto óptimo entre dos estaciones del metro de Praga, para ello hemos implementado el Algoritmo A* en Java.

El algoritmo A* se trata de un algoritmo de búsqueda que encuentra, siempre y cuando se cumplan unas determinadas condiciones, el camino de menor coste entre un nodo origen y un nodo objetivo/destino.

- **Algoritmo A***

  Hemos creado una clase en nuestro proyecto "practica_inteligencia" llamada AlgoritmoEstrella en la que implementamos dicho algoritmo además de calcular G(n) y H(n).

  Como hemos comentado anteriormente el algoritmo A* se basa en la búsqueda del camino de menor coste entre un origen y un destino. Para implementar dicho algoritmo primero creamos dos atributos llamados "Pila abierta" y "Pila cerrada" donde iremos metiendo y sacando las estaciones del metro del Praga.

  El algoritmo A* funciona de forma recursiva introduciendo en la pila abierta las estaciones adyacentes de menor coste utilizando la función G(n) y H(n) hasta que encuentre la estación destino verificando que no existe recorrido de menor coste.

  Las distancias coste se calculan mediante una función que calcula la distancia entre dos puntos utilizando como x e y la longitud y latitud descrita en Google Maps y obteniendo el resultado en metros.

  Calculamos la distancia recorrida con el sumatorio de distancias entre estaciones.

  Finalmente calculamos el tiempo estimado del recorrido suponiendo una velocidad media "x" del tren y un tiempo adicional de transbordo cuando este existe.

- **EstacionesPraga**

  En esta clase hemos creado todas las estaciones del metro de Praga, empezando por la línea A (verde), seguido de la B (amarilla) y por último la línea C (roja).

  En cada línea hemos implementado las estaciones como si de nodos se tratase, indicando siempre su anterior y su siguiente además de parámetros como el color de línea que es, si tiene transbordo, en el caso de lo que hubiese el tiempo de este mismo y las coordenadas en las que se encuentra esa parada. Para este ultimo parámetro hemos utilizado una herramienta de Google llamada "Obtener coordenadas de un punto en google maps" que nos da la latitud y longitud de cada parada del metro de Praga.

  Los transbordos los hemos tratado como una sola estación, es decir, si una parada aparecía en dos líneas diferentes como es por ejemplo "Mustek" que aparece en la línea A y en la línea B solo hemos creado tal estación en una de las líneas, ya que si la creábamos en las dos líneas nos surgían conflictos.
  En las estaciones con transbordos encontramos una pequeña diferencia, y es que hay que tratar cuatro vecinos y no dos como en el resto de estaciones.

  En esta misma clase, hemos definido un método en el que determinamos todas las estaciones del metro de Praga por orden alfabético para su posterior uso en la interfaz gráfica.

- **Imagen**

  Esta clase ha sido creada únicamente para importar la imagen que usaremos en la interfaz gráfica.

- **AtributosEstacion**

  En esta clase están definidos todos los atributos de las estaciones del metro de Praga entre los que se encuentran el nombre de dicha parada, la estación

siguiente, la estación anterior, el transbordo si lo hay, el tiempo de este mismo, las coordenadas, etc.

Creamos dos constructores con esos atributos teniendo en cuenta si existe o no transbordo.

En esta clase también están implementados los correspondientes getters y setters.

- **InterfazGrafica**

Esta clase, como su propio nombre indica, es la interfaz gráfica donde diseñamos como será la aplicación.

Primeramente para poder usar una interfaz gráfica en Java nos ha hecho falta la instalación de un plug-in llamado Swing que obtenemos de la herramienta Windows Builder.

Una vez instalado, hemos procedido a crear la interfaz gráfica introduciendo lo primero una imagen del metro de Praga para que sea más visual el trayecto mínimo a recorrer entre dos estaciones origen y destino.

Posteriormente hemos creado una serie de botones, el primero de ellos el botón “origen” donde podemos desplegar toda la lista de estaciones para elegir la estación de la que queremos partir. Creamos un botón exactamente igual pero al que llamamos “destino” y su función es seleccionar la estación a la que queremos llegar.

Diseñamos dos botones más, uno de ellos llamado “Hallar ruta” que pincharemos sobre él cuando ya hayamos seleccionado nuestra estación origen y nuestra estación destino para que nos muestre la lista resultado de las estaciones que debemos recorrer. Dicha lista aparecerá por orden de estaciones en un cuadro que hemos creado para mostrar por pantalla dicho recorrido.

El otro botón, citado anteriormente, se llama “Nueva búsqueda” que tendremos que pinchar sobre él siempre que queramos hacer otro nuevo recorrido, para restaurar todos los valores.

Hemos creado otro cuadro más en el que mostramos el tiempo estimado del recorrido escogido.

Y por último existe otro cuadro en el que damos a conocer la distancia recorrida en ese trayecto.

Con los diferentes comandos de esta herramienta hemos declarado el tamaño de los botones y de los cuadros, su posición, y su color. Además del color del fondo de la aplicación, el tamaño de la letra que hemos utilizado y el tipo de esta misma.

Por último hemos empleado una función llamada “fillOval” a la que se le pasan cuatro parámetros (x, y, ancho, altura). Esta función nos dibuja puntos en las coordenadas que le pasemos, en este caso las estaciones del metro de Praga. Por lo tanto cuando elijamos un recorrido y hallemos la ruta esta función nos dibujará sobre nuestra imagen insertada los puntos del recorrido. Hemos utilizado “setColor” para elegir el color de lo puntos, que hemos decidido que sean del color de su correspondiente línea menos los transbordos que los hemos coloreado de color negro.