

Memoria

Grupo 57

- Aarón Cabero Blanco
- Xiao Peng Ye
- Daniel Tomás Sánchez
- Juan Diego Valencia
- Sergio Sánchez-Carvajales Franco

1. Desarrollo del algoritmo A*

A la hora de comenzar el desarrollo de la práctica hemos tenido que realizar un grafo en el que se especifican que estaciones están comunicadas entre sí y, además, cuál es la distancia entre dichas estaciones. Esta distancia, que es la distancia real entre las estaciones, la hemos buscado en Google y hemos creado el grafo a partir de dicha información.

Por otro lado, la distancia aérea entre las estaciones la hemos calculado mediante las coordenadas de Google Maps. Dicha distancia aérea ha sido calculada con las coordenadas nombradas anteriormente y con una función de nuestro lenguaje de programación que, en este caso, ha sido Python.

Una vez hemos extraído todos los datos en bruto, los hemos organizado en diferentes estructuras de datos para que fuera más fácil su uso.

A la hora de calcular $g(n)$, este se ha calculado con la distancia real que cuesta llegar a una estación partiendo desde la estación origen y pasando por las diferentes estaciones que nos ha ido diciendo el propio algoritmo A*. En cuanto a calcular $h(n)$, este ha sido calculado mediante un valor aproximado de la distancia aérea entre la estación en la que se encontraba el algoritmo en ese instante y la estación destino. Para este $h(n)$ se ha usado la función descrita anteriormente para calcular las distancias aéreas entre estaciones.

Con el cálculo de ambos, $g(n)$ y $h(n)$, hemos calculado el $f(n)$ que es la distancia aproximada para llegar a una estación partiendo desde el origen. En cada paso del algoritmo y, mediante una PriorityQueue (cola con prioridad) hemos ido extrayendo la siguiente estación a explorar que es la que menos coste tiene, por ello se ha usado una PriorityQueue. Tras iterar este algoritmo en algún momento llegamos al destino, puesto que todas las estaciones están conectadas con todas mediante otras estaciones o directamente entre ellas.

Para tener en cuenta aspectos como el transbordo, hemos tenido que almacenar la línea por la que se llega a una estación. En ese caso se añade una penalización en el tiempo y en distancia para así acercarnos más a la realidad y dar un resultado más certero.

2. Desarrollo de la GUI (Interfaz Gráfica)

El desarrollo de la interfaz gráfica ha sido un reto para nosotros, pues es la primera vez que la realizamos, añadiendo que se ha realizado Python con la librería PyQt5 y con un programa de diseño gráfico específico para esta librería, QtDesigner.

Para esta interfaz hemos abarcado varios aspectos importantes:

- Un diseño minimalista, simple y atractivo para el usuario.
- Una total visualización del trayecto, señalando el camino en el propio mapa.
- Versatilidad, siendo multiplataforma, ya que funciona en distintos sistemas operativos.
- Facilidad de uso, ya que la obtención de los resultados solo conlleva realizar dos simples pasos. Incluso existe una ventana de ayuda con los pasos a realizar.
- Visibilidad, ya que los resultados obtenidos se pueden observar inmediatamente al seleccionar las paradas deseadas.

3. Instrucciones de ejecución

1. Acceder a la carpeta bin desde el directorio del proyecto.
2. Hacer doble click en "Metro Atenas.exe"

4. Observaciones

Uno de los principales problemas que hemos encontrado para poder realizar este trabajo ha sido a la hora de poder calcular la distancia aérea entre las estaciones, ya que tuvimos que hacer uso de sus coordenadas mediante Google Maps, además de poner en datos todas las comunicaciones entre las estaciones junto a su distancia real. Ha sido un poco tedioso extraer todos los datos en bruto, pero finalmente ha sido satisfactorio ver que el algoritmo funcionaba correctamente.

El cálculo de la velocidad media del tren también fue otro de nuestros problemas, ya que tuvimos que informarnos y hacer un balance sobre la velocidad a la que iban los trenes en el metro de Atenas.

En cuanto a cuestiones de implementación, en vez de hacerlo en Java hemos realizado el proyecto en **Python** siguiendo los consejos de Vicente, el coordinador. Como este lenguaje de programación es nuevo para nosotros, nos hemos tenido que adaptar tanto a la hora de realizar el algoritmo como para realizar la GUI. Si bien es cierto que el hecho de ser un lenguaje nuevo ha sido un poco más trabajoso, hemos podido aprender mucho y explorar nuevos lenguajes para salir de nuestra zona de confort.

Por otro lado, esta es la primera vez que hemos realizado una GUI en nuestro grado, por lo tanto, esto ha sido lo más complicado. Aunque también es otro aspecto enriquecedor que nos llevamos, ya que son una parte muy importante de la programación.

5. Conclusiones

Hemos podido comprobar que el algoritmo A* se puede aplicar en ámbitos cotidianos como es el de este sistema de transporte. Por esta razón, resulta muy útil para la población ya que se aplica a un ámbito que se usa diariamente y con mucha frecuencia.

Además, hemos visto la gran relevancia que tiene una interfaz gráfica ya que los resultados obtenidos se observan en ella y vemos en cierto modo la fiabilidad del algoritmo a diferencia de solo presentar los resultados, pues nos veríamos obligados a creérmolos. Otro aspecto también relevante que hemos visto y nos los llevamos aprendido para el futuro, es la calidad y simpleza que ha de tener la interfaz gráfica, puesto que es con lo que el usuario interactúa y dependiendo de ello, se decantaría por una aplicación u otra.