Fast-Metro

Présentation du logiciel

Fast-Metro est un logiciel permettant de trouver le plus court chemin entre deux stations de métro. Ce logiciel utilise la puissance de Java, de Jframe (bibliothèque graphique native de Java), Maven (pour les dépendances) et du standard Json, librairie Gson (lire et écrire sur un fichier) pour faciliter les transferts de données.

Execution

java -jar releases/app/*.jar

Mode d'Emploi

On selectionnne les deux stations qu'on souhaite voir le plus court chemin. On clic. Le logiciel va afficher le plus court chemin. On clic pour reinitialiser la carte.

Explication de la structure du projet

Voici un résumer de la structure du projet:

1) fastmetro package: contient la classe main et les classes de structures de données
   a) Main.class: class principal.
   b) Carte.class: Tous ce qui concerne la carte.
   c) Station.class: Tous ce qui concerne une station (numéro de la stations,ligne). Un station est traversée par une ligne dans les deux sens de circulations.
   d) Gare.class: Une gare est un ensemble de stations. Elle contient un nom et une liste de stations.
   e) Lecture.class: Pour lire un json.
   f) Dijkstra: Implémentation de l'algo et utilisation d'un tableau pour faire les opérations.
   g) Ecriture.class(non utilisé): Pour écrire sur un json(initialisé des stations sur une nouvelle carte et faire un nouveau graphe)
2) GUI package: pour l'interface graphique utilisateur (voir dans la javadoc dans Documentation/index.html avec un navigateur pour plus de detail). Mais en gros, on a une fenetre, un panel(station pannel) pour dessiner les stations(changer les couleurs etc. . .) et circle qui représente un cercle utiliser pour l'affichage des stations.

Algorithme du plus court chemin: Dijkstra

Structure de données

On a deux tableaux:

- Matrice de Dijkstra Dynamic, c'est-à-dire en abscisse les id des stations et en ordonnée les étapes. A chaque case, on a un couple de valeur: <le cumul du temps, le père>
- Liste dynamic des stations coloriés. Quand une station est colorié on la supprime de la liste.

Dérouler de l'algorithme

Voici l'algo adapter pour le programme:

- On initialise la première ligne de matrice (+inf est 2 à la puissance 31, c'est la valeur max d'un int).
- On boucle tant que le tableau des stations coloriés n'est pas vide.
- On copie la première ligne à la deuxième.
- On cherche tous les voisins de la première station.
- Pour chaque voisin on vérifie si elle inferieur à celle entrée. Et on écrase la valeur.
- Puis on cherche les pères successif du tableau de la dernière ligne.

Détail de l'utilisation de l'interface graphique

clic

Le clic est mis en écoute. Quand on clic on fait appel à l'objet dijkstra initialisé dans la carte. Les tableaux sont temporaires et change à chaque itérations.

Contenu

Cartes

On peut changer de carte dans image/carte.gif (faut changer le main en conséquence)

Stations et Gare

On peut changer les stations, elles sont sauvegardées dans data carteStation.json et carteStation.json

Source

src/*

Documentation

javadoc dans Documentation/index.html

Annexe

Screencapture

Disponible sur github

https://github.com/dytq/Fast-Metro