Découpage fonctionnel

Projet Algorithmique 2A - Lyon maps

CHAHINE Maroun
HABIB Daniel
CHORBAGI Ahmad
MARCHI MEKARI Gabriel

Promotion 2022-2023
Lanière N, Groupe 53

Extraction de données

La classe lecture permet d'extraire les données des fichiers csv et créer un dictionnaire structuré de manière efficace pour son exploitation.

Calculs

Nous avons décidé d'utiliser l'algorithme de Dijkstra pour créer le chemin. Cette méthode nous semble plus adaptée au projet. Chaque station est un sommet, et à chaque mode de transport correspond un poids, représentant le temps réel (calculé avec la distance et la vitesse) entre deux stations. La fonction creer_graphe() permet de créer le graphe qu'utilise l'algorithme de Dijkstra.
On trouve d'autres fonctions de calcul dans notre programme:

- transport_from_station(): prend en argument une station, et retourne les différents transports qui passent par celle-ci sous forme de liste.

- coord_clicked_depart() et coord_clicked_arrivee() permettent de récupérer les coordonnées des stations sélectionnées lorsque l'utilisateur appuie sur la carte pour définir son trajet.

- plus_proche_station() permet de retrouver la station la plus proche par rapport aux clics de l'utilisateur sur la carte. Celle-ci retourne un tuple correspondant à la station de départ et la station d'arrivée.

- lat_lng_from_nom() prend en argument une station, et retourne sa latitude et longitude.

- xy_repere_cartesien() retourne les coordonnées d'une station en fonction de sa latitude et longitude. On y trouve aussi un calcul d'erreur et une échelle qui serviront au placement de la station sur la carte. Cette méthode nécessite d'autres fonctions telles que distanceKm() qui renvoie la distance entre deux points repérés par leur latitude et longitude; distance() qui renvoie la distance entre deux points repérés par leurs coordonnées cartésiennes; et convertRad() qui prend un angle en degrés et retourne sa valeur en radians.

- angle_between_points() prend en argument la latitude et longitude entre deux points, et grâce à un calcul trigonométrique, retourne l'angle entre ces deux points sur la carte.
  extreme_arc() : est une fonction qui prend en input les coordonnées de la station de référence (IUT Feyssine), la distance entre la station de référence et une station donnée qu'on veut représenter, et end_angle qui est l'angle de la boussole de la station qu'on veut dessiner en prenant comme centre de la boussole la station de référence. Cette fonction renvoie les coordonnées de la station qu'on veut représenter.
  Ces deux méthodes sont nécessaires au placement des stations sur la carte.

Visualisation et IHM

Nous avons créé deux classes, correspondant aux deux fenêtres qui s'affichent. La première classe Ouverture correspond à l'écran d'accueil, et Ecran2 correspond à la fenêtre dans laquelle l'utilisateur fait ses choix.
Dans la classe Ouverture, nous avons l'initialisation et une méthode qui permet d'accéder au deuxième écran quand l'utilisateur appuie sur un bouton.

Dans la classe Ecran2, nous avons, outre l'initialisation:

- creer_widg(): permet de créer les boutons et les zones de texte sur la fenêtre pour que l'utilisateur fasse ses choix. Les combobox sont des zones de texte avec une liste déroulante intelligente. Celles-ci sont créées dans une autre classe, mais appelées dans cette méthode. Ainsi, l'utilisateur pourra choisir les stations à l'aide du combobox, ou sur la carte.

- Ces fonctions permettent de faire fonctionner le logo qui s'affiche lorsque 10 secondes passent sans que l'utilisateur bouge son curseur. task_sleep() permet de vérifier que 10 secondes sont passées depuis l'inactivité de l'utilisateur, photo_go() permet d'afficher le logo, lancer() permet de débuter l'animation, move() fait bouger l'image, timer_loop() est une boucle qui permet le déplacement régulier de l'image. Wake_up() est appelée lorsque le curseur est déplacé et permet d'effacer l'image et d'arrêter le mouvement, et stop() d'arrêter l'animation. La classe TimerPeriodic permet de vérifier régulièrement si les 10 secondes d'inactivité sont passées.

- effacer_message_depart() et effacer_message_arrive() servent à effacer les messages qui s'affichent lors de l'ouverture de la carte.

- dessiner_trajet(): cette méthode permet de dessiner le trajet sur la carte.

Fonctions principales

- trajet_combobox(): Cette méthode de l'écran 2 permet de récupérer les stations écrites dans les zones de texte. A l'aide de l'algorithme de Dijkstra, elle permet de calculer le plus court chemin sous forme de liste, et puis dessine le trajet sur la nouvelle fenêtre graphique. Aussi, cette fonction permet d'afficher notre logo si l'utilisateur ne bouge pas son curseur pendant 10 secondes.

- creer_fen(): Cette fonction a un rôle primordial si l'utilisateur veut choisir le point de départ et le point d'arrivée graphiquement sur la carte. Elle permet de créer la carte et indique à l'utilisateur de choisir son point d'arrivée. Elle est appelée lorsque l'utilisateur appuie sur "Choisir sur la carte", et elle fait appel aux fonctions permettant de calculer le trajet entre deux stations choisies par l'utilisateur.