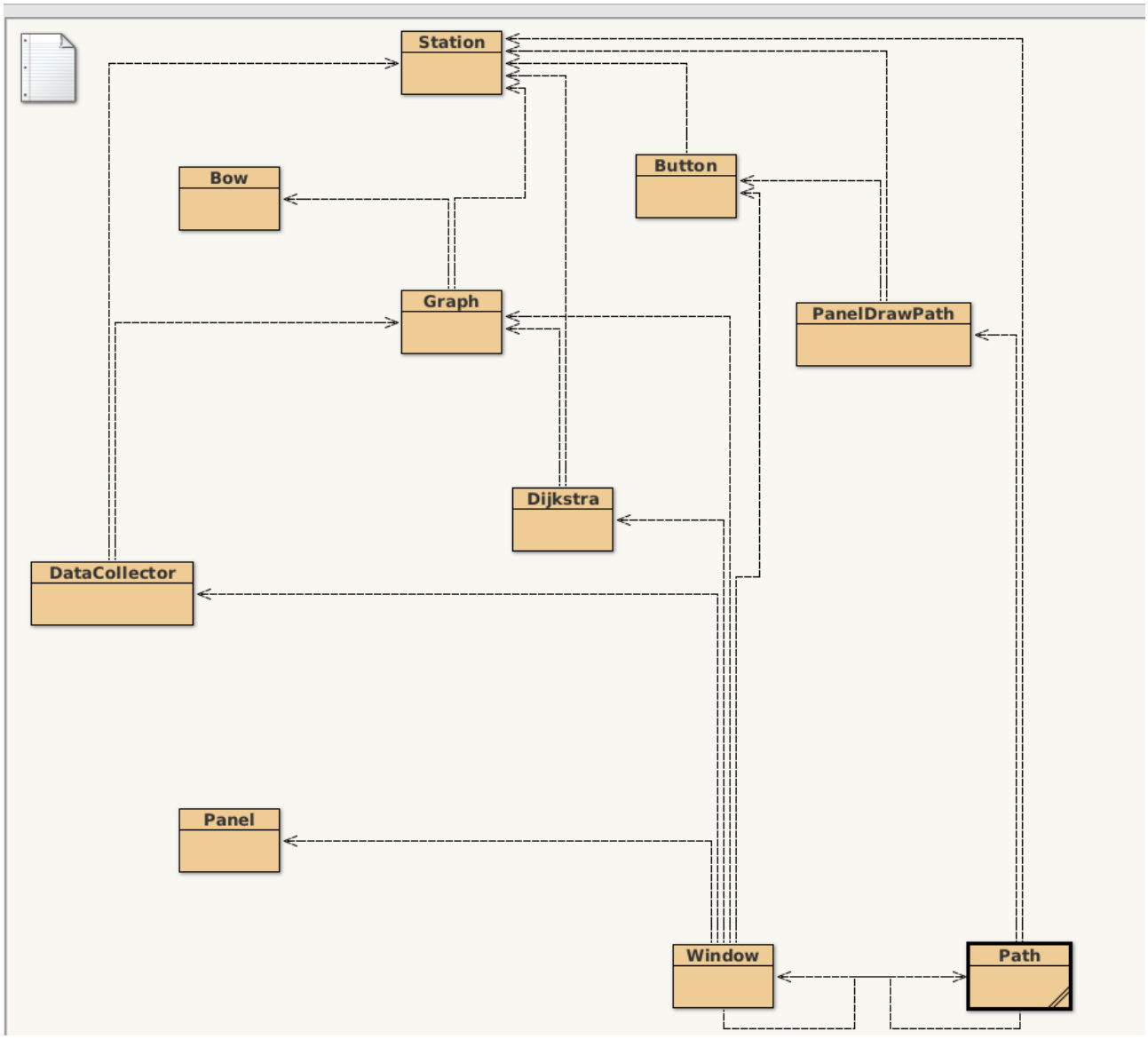
Rapport Du Projet Metro

HARAL DYLAN 21700650
FALL ABDOU AZIZ 21804685
BERKANI TINHINANE 21606277

I) Description des classes

Pour la description des classes nous avons généré la documentation des classes que nous avons réalisé tout au long de ce projet. La documentation se trouve dans le dossier documentation dans allclasses.html.



Diagramme

II) Préparation du projet

Nous avons choisi le langage java pour réaliser l'application.

L'application permet d'afficher le plus court chemin entre deux stations de métro. Ces dernières

sont sélectionnées par l'utilisateur par le biais de la souris en cliquant sur deux stations de métro. Cet affichage nécessite au préalable de connaître les coordonnées de chaque station de métro sur la carte de métro. C'est la raison pour laquelle nous avons rajouté quelques données supplémentaires comme les coordonnées de chaque station de métro dans le fichier métro.txt. En plus des coordonnées des stations, nous avons ajouté le terminus de chaque ligne de métro pour connaître les directions.

Pour le choix d'algorithme, l'algorithme de Floyd-Warshall est le plus adopté et facile à implémenter pour calculer le plus court chemin entre des stations car il calcule le plus court chemin entre tous les pairs de sommets avec une complexité $O(n^3)$, n le nombre de sommets. Avec celui de Dijkstra, on traite pour chaque itération un sommet en O(n) et à chaque traitement de sommet, on parcourt le tableau des plus courtes distances en O(n) ce qui donne une complexité $O(n^2)$ et un plus court chemin entre deux stations, si s'était celui de Floyd-Warshall on aurait n plus chemins différent ce qui n'est pas réaliste pour plus de 300 stations. C'est la raison pour laquelle nous avons préféré Dijkstra pour calculer le plus court chemin entre stations.

L'implémentation de l'algorithme est assez complexe mais nous avons pris le soins de bien découper le code en plusieurs bouts. L'implémentation se déroule en 4 étapes :

- on initialise les données avec le fait qu'aucun sommet n'est était traité et qu'aucun sommet n'a de père. Les plus petites distances ont pour valeur l'infini.

- on traite le sommet de départ et on met à jour les plus petites distances avec ses successeurs en utilisant les pondérations.

- tant que tous les sommets ne sont pas traités, on prend un sommet non traité qui a la plus petite distance dans le tableau de distance et on regarde son successeur avec la distance la plus petite. On passe par ce successeur ce dernier devient un sommet traité et le prédécesseur de ce sommet devient le père.

- le tableau de père est mis à jour.

III) Fonctionnement général de l'application

Le lancement de l'application se fait via la commande « make » si cette commande est installée sinon il faut compiler le fichier principale « Window.java » avec la commande javac Window.java puis exécuter le programme par « java Window ».

Lorsque l'application est lancé toutes les données du graphe (le tableau des stations de métro, la matrice qui représente les arcs) sont initialisées par la classe DataCollector qui récupère les données depuis metro.txt. Une fois les données initialisées, la carte du métro est chargée depuis la racine du projet et les « buttons » représentant les stations de métro sont instanciées avec les coordonnées de chaque stations à partir du tableau des stations (ArrayList) qui se trouve dans la classe Graph. La classe Dijkstra est instancié par défaut.

C'est ainsi que l'interface graphique qui est la fenêtre principale (Window : classe) apparaît avec la carte de métro et les boutons sur la carte.

Lorsqu'on survole un bouton ou on pose la souris sur le bouton, l'étiquette du bouton apparaît avec le nom de la station.

Une fois que l'utilisateur clique sur une station de départ et une station d'arrivée, le plus court chemin

entre les deux stations est calculé et la fenêtre principale est fermée.

Une nouvelle fenêtre apparaît contenant la carte du métro avec le chemin reliant les deux stations sélectionnées précédemment par l'utilisateur avec un bouton rouge étiqueté 'Close Window'.
La station de départ est représentée par un gros bouton de couleur vert de même que la station d'arrivée avec une couleur bleu. Une étiquette qui contient le nom de chaque station apparaît lorsqu'on survole le bouton ou que l'on pose la souris sur le bouton.

La description du trajet apparaît dans le terminale ainsi que la durée du trajet.

En fin pour fermer la fenêtre il faut cliquer sur le bouton Close Window qui se trouve en bas de l'interface graphique.

IV) Améliorations

Ajouter des fonctionnalités comme l'effet de zoom sur la carte ou faire un programme multi-threads pour gérer la navigation entre les deux interfaces graphiques. Pour l'effet de zoom il faut recalculer les coordonnées la nouvelle carte et forcer le composant à ce repeindre.