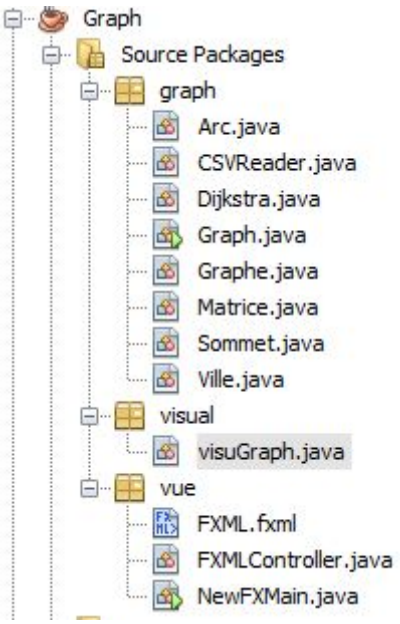
Compte rendu de projet de Graphe

Pittore Axel - Funes Héléna

Introduction :

Ce projet permet de représenter les villes de France avec des graphes. On fait un filtrage sur la population et on détermine à partir de quelle distance à vol d'oiseau deux villes sont reliées, puis on crée le graphe.

1. Implémentation de la structure de graphe :



Arborescence du projet

Pour implémenter la structure de graphe, nous avons décidé de créer une classe Graphe qui possède comme attribut deux ArrayList, une remplie à l'aide de plusieurs objets de classe Sommet (les noeuds) et l'autre remplie d'objets de classe Arc. Un Sommet a un String nom, et des valeurs de latitude et longitude. Enfin, un Arc a en attribut deux sommets et une valeur d'arc. On a bien ainsi une structure de graphe implémentée correctement. Les classes CSVReader et Ville servent à lire le fichier CSV pour la première, et stocker les données lues ainsi pour la deuxième. Enfin, il existe un constructeur qui permet de transformer directement un ArrayList de Ville en Graphe. Il y a en plus un deuxième paramètre qui est la distance qui à partir de laquelle on considère que deux ville ne seront pas reliées par un arc. La classe Matrice correspond à une matrice d'adjacence que l'on crée à partir d'un Graphe. Elle est utilisée par la classe Dijkstra qui implémente l'algorithme éponyme. La classe visuGraph sert à lier un graphe de la classe que nous avons créée à un Graph de la librairie GrapheStream, que nous avons utilisée pour modéliser en 2D nos graphes. Enfin, le package Vue contient un début d'interface graphique, mais celle-ci a été abandonnée à causes de problèmes que nous aborderons plus tard.

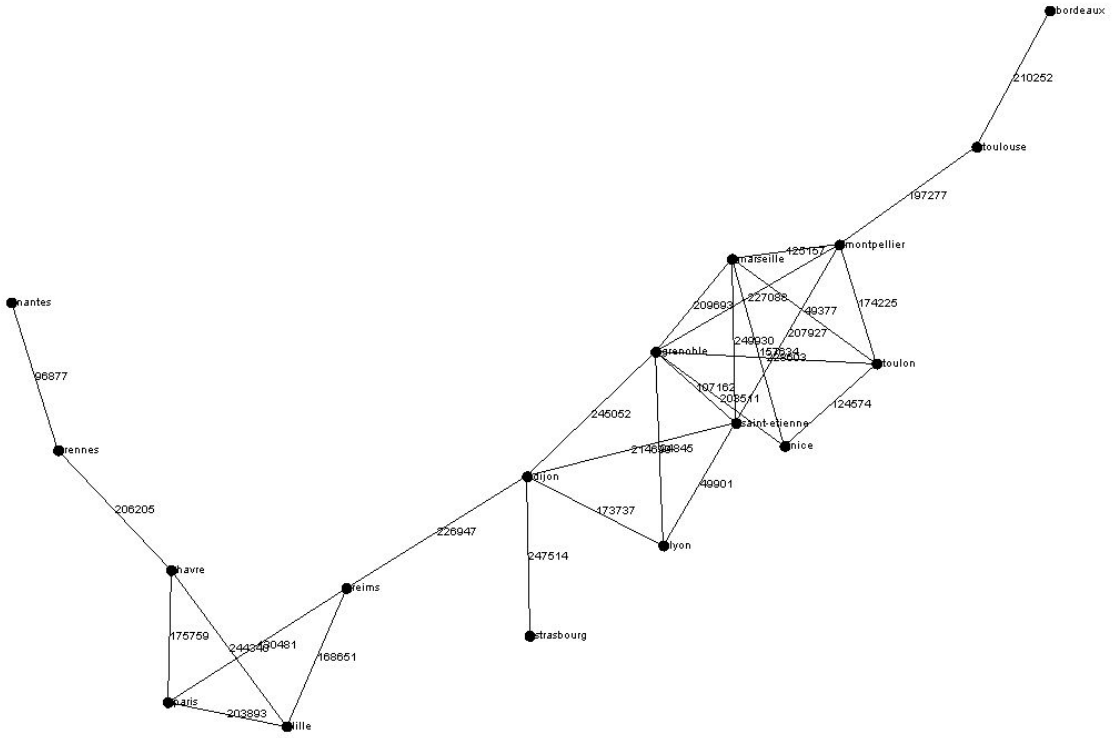
On peut afficher le graphe en console :

```
****** graphe ******
**Villes**
nice
marseille
dijon
toulouse
bordeaux
montpellier
rennes
grenoble
saint-etienne
nantes
reims
lille
strasbourg
lyon
paris
havre
toulon
**Arcs**
Arc reliant nice à marseille de valeur 157634 mètres
Arc reliant nice à montpellier de valeur 271082 mètres
Arc reliant nice à grenoble de valeur 203511 mètres
Arc reliant nice à saint-etienne de valeur 296816 mètres
Arc reliant nice à lyon de valeur 297655 mètres
Arc reliant nice à toulon de valeur 124574 mètres
Arc reliant marseille à montpellier de valeur 125157 mètres
```

2. Création de plusieurs graphes :

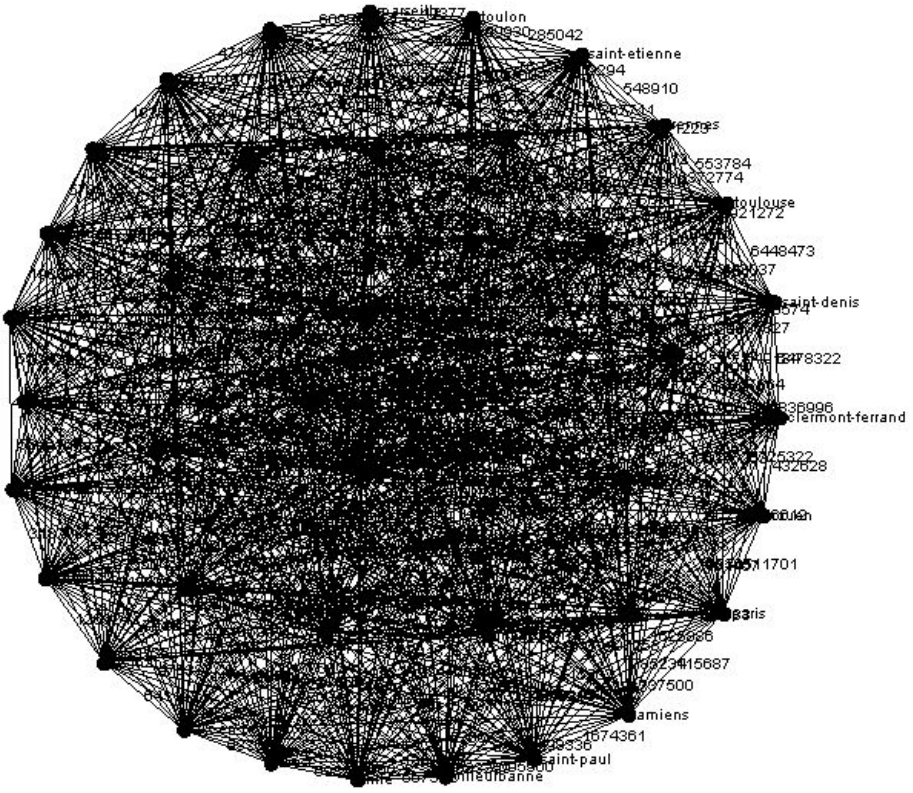
- Essayer de modéliser un graphe avec toutes les communes a pris trop de temps sur notre machine, nous avons donc réalisé des graphes plus petits.

- Avec les villes de population supérieure à 150.000 et un arc si la distance à vol d'oiseau est de moins de 250 km, nous avons un bon résultat

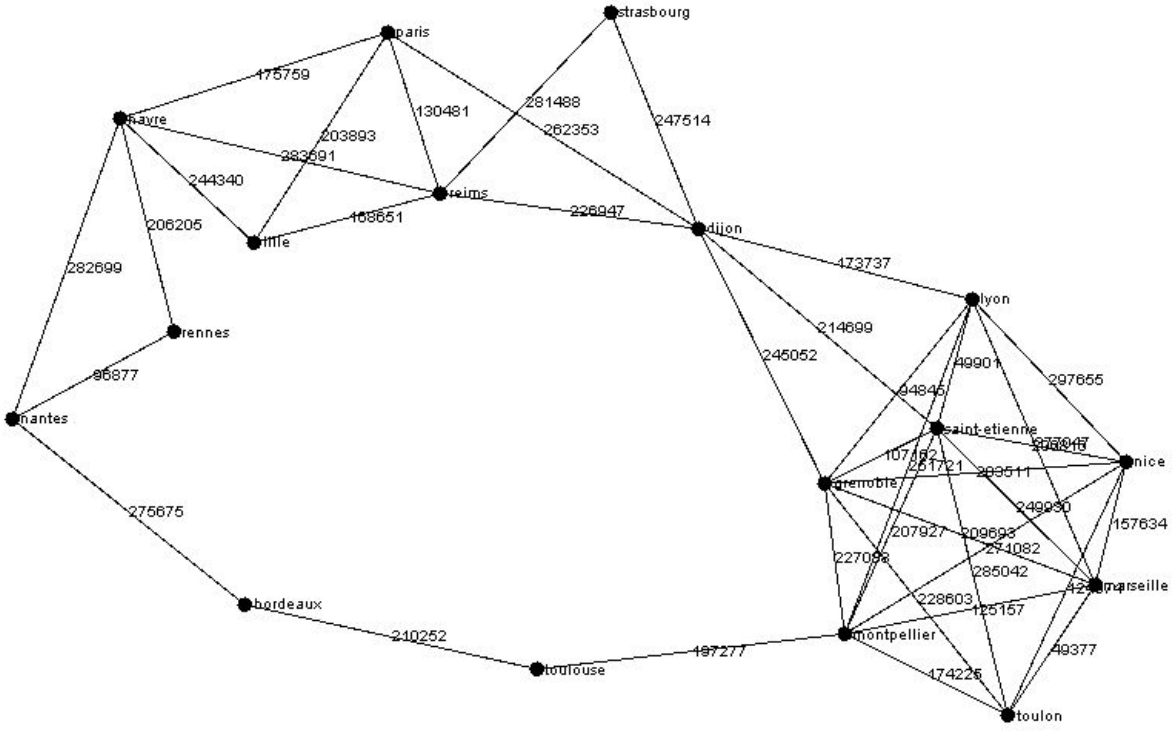


Graphe avec les villes de population supérieure à 150.000 et un arc si la distance à vol d'oiseau est de moins de 250 km

- Avec les villes de population supérieure à 100 000 et un arc si la distance à vol d'oiseau est de moins de 100 000 km, nous avons le graphe complet suivant

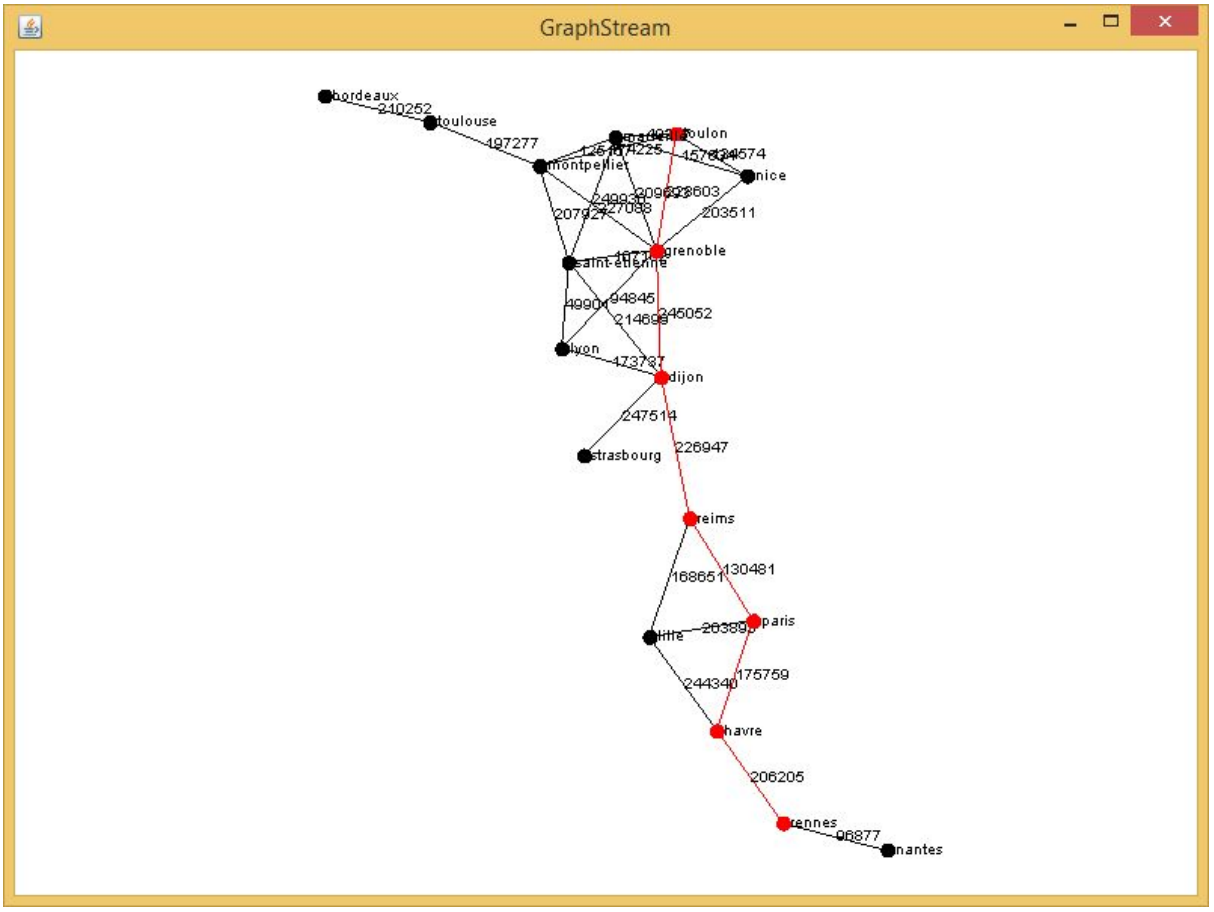


- Avec les villes de population supérieure à 150 000 et un arc si la distance à vol d'oiseau est de moins de 300 km, nous avons le graphe complet suivant

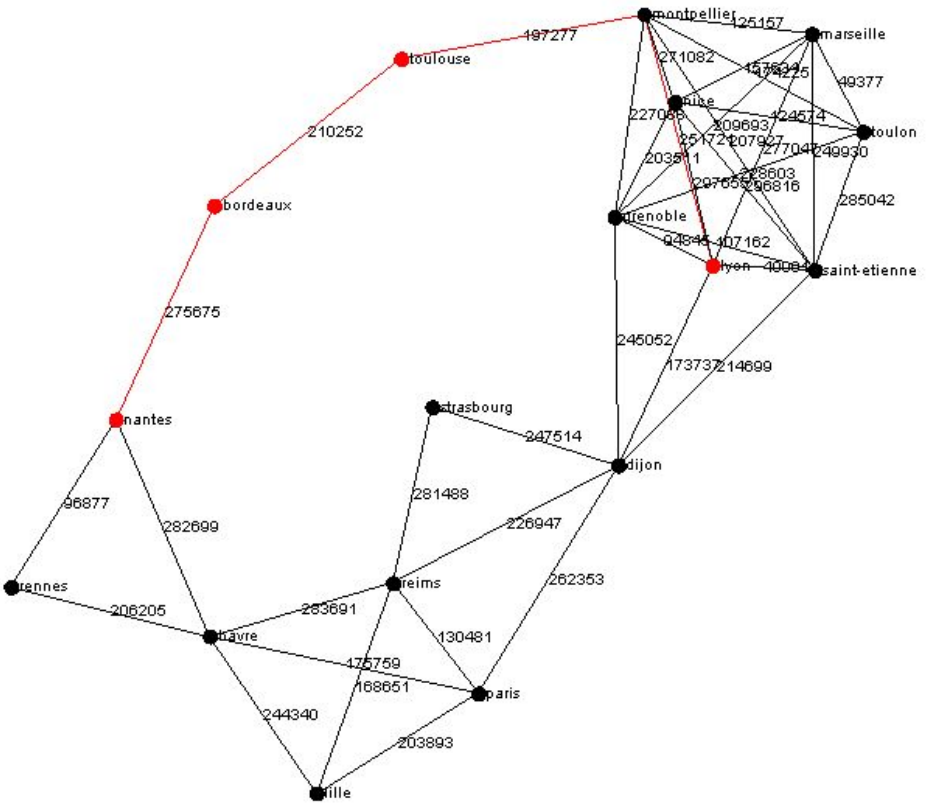


3. Implémentation des algorithmes :

Nous avons seulement implémenté l'algorithme de Dikjstra. Nous avons eu pendant longtemps des problèmes dessus qui étaient au final liés à la condition d'arrêt. En effet, si il existait plusieurs composantes connexes dans le graphe, il plantait. Mais ce problème est maintenant réglé, et l'algorithme s'applique à tous les graphes. Le code est disponible dans la classe Dijkstra, principalement la méthode plusCourtChemin qui est commentée.



Plus court chemin de rennes à toulouse sur le premier graphe

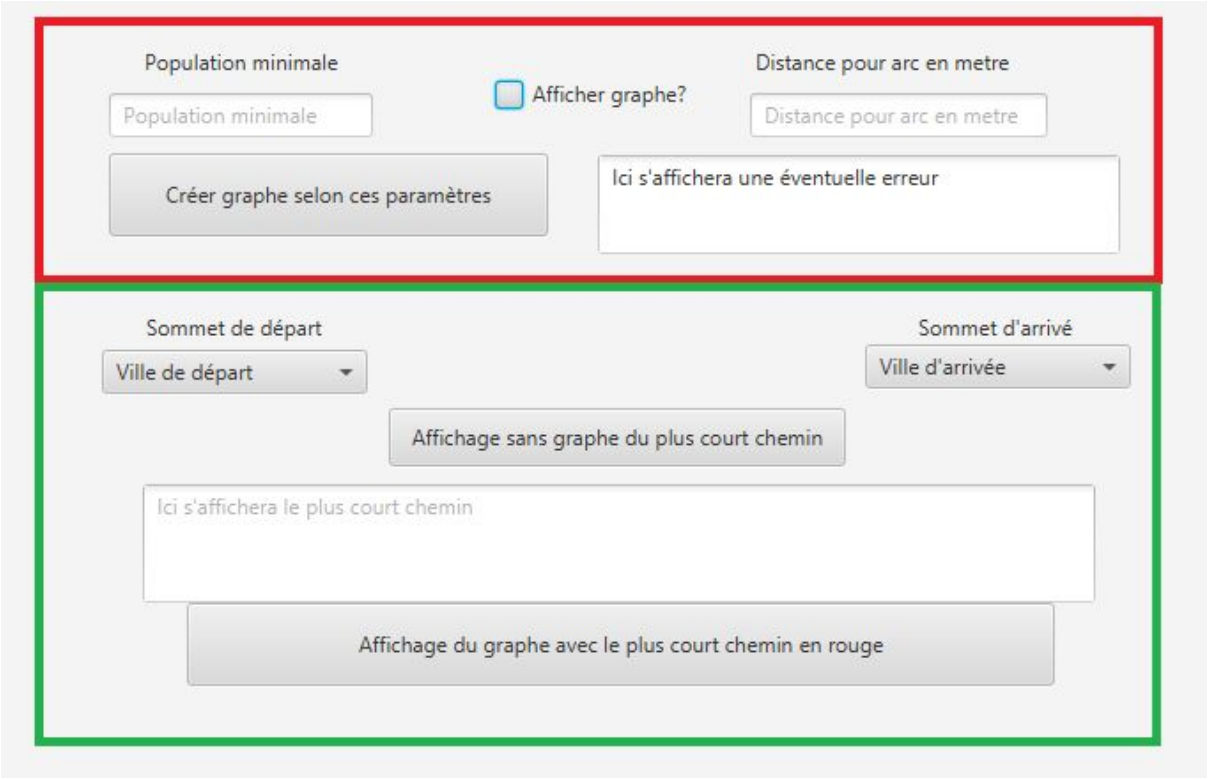


Plus court chemin de lyon à Nantes sur le troisième graphe

```
-
Chemin de Sommet nantes à Sommet lyon
Sommet nantes-> Sommet bordeaux-> Sommet toulouse-> Sommet lyon
De valeur 737648
```

Résultat dans le console :

Utilisation de l’interface graphique :



L’interface graphique se divise en deux parties, la partie verte, et la partie rouge.

La partie rouge sert à créer un graphe, et à l’afficher si on le souhaite. Pour cela, on rentre la population minimale que l’on veut pour prendre en compte la ville, et la distance maximale à partir de laquelle on admet qu’un arc existe entre 2 villes. Puis on choisit d’afficher le graphe ou non. ATTENTION : Afficher le graphe ouvre une nouvelle fenêtre dans laquelle il y a le graphe, mais lorsqu’on ferme celle-ci, l’interface graphique se ferme, et il faut repartir de zero, si l’on voulait par exemple, faire une recherche de plus court chemin.

La partie rouge sert à afficher le résultat d’une recherche de plus court chemin entre 2 villes. Il faut, avant de l’utiliser, avoir auparavant créé un graphe sans l’afficher dans la première partie. Puis, on choisi les deux villes dans les listes déroulantes, puis on choisit le mode d’affichage : - sans graphe, c’est à dire juste des lignes de textes indiquant le chemin et la distance

- avec graphe, cad avec un graphe dans lequel on affiche les points et les arêtes en rouge du plus court chemin.

Résumé d’utilisation du projet :

Vous pouvez soit aller dans le main “graph” du package graph, et modifier les données du début du main pour faire les graphes que vous souhaitez, ou sinon, faire un click droit sur le fichier “NewFXMain” du package Vue, et faire un “Run File” pour lancer l’interface graphique.