2. MANUEL D'UTILISATION

Une fois le projet décompressé, il suffit d'exécuter le programme en ligne de commande par l'intermédiaire de la classe InterfaceConsole grâce à la commande :

Java InterfaceConsole

Ensuite, un menu intuitif s´affiche, il permet d´accéder aux principales fonctionnalités du programme.

```
-----MENU------
1- Charger le Graphe.
2- Vérification de la connexité.
3- Calcul du plus court chemin simple.
4- calcul du plus court chemin optimisé.
5- calcul du plus court chemin optimisé avec prise en compte de l'autonomie.
0- Quitter.
Votre choix :
```

Figure 1: Interface Console

Une fois le graphe chargé, on peut voir des messages de confirmation ainsi que le nombre de sommets et le nombre de stations pour ce graphe.

```
Votre choix : 1
Nombre de sommets : 52918
Sommets charges
Aretes chargees
Nombre de stations : 93
Stations chargees
```

Figure 2 : Chargement du graphe avec l'interface

Il est à noter que c´est le graphe de la France qui est chargé par défaut, mais on peut modifier ce paramètre directement dans la classe.

```
switch (choix){
case 1 :
    int nbSommet = LecteurSommets.lireNbSommet("./Src/france.sommets");
    graph = new Graphe(nbSommet, null);
    System.out.println("Nombre de sommets : "+ nbSommet);
    LecteurSommets.lireSommet("./Src/france.sommets", graph);
    System.out.println("Sommets charges");
    LecteurAretes.lireAretes("./Src/france.aretes",graph);
    System.out.println("Aretes chargees");
    int nbStations = LecteurStations.nbStations("./Src/france.stations");
    petitgraph = new petitGraphe(nbStations, graph);
    System.out.println("Nombre de stations : "+ nbStations);
    LecteurStations.lireStation("./Src/france.stations");
    break;
```

Figure 3 : InterfaceConsole.java, code de chargement du graphe

On peut ensuite voir si le graphe est connexe ou non grâce à l'option 2 de l'interface ainsi que calculer le plus court chemin avec ou sans contraintes, optimisé ou non (sans ou avec tas).

🗙 xterm 📃 🗖 🔀
~/4annee/graphes/Graphes : java InterfaceConsole
MENUMENU 1- Charger le Graphe. 2- V??rification de la connexit??. 3- Calcul du plus court chemin simple. 4- calcul du plus court chemin optimis??. 5- calcul du plus court chemin optimis?? avec prise en compte de l'autonomie. 0- Quitter. Votre choix : 1 Nombre de sommets : 52918 Sommets charges Aretes chargees Nombre de stations : 93 Stations chargees
MENUMENU 1- Charger le Graphe. 2- V??rification de la connexit??. 3- Calcul du plus court chemin optimis??. 5- calcul du plus court chemin optimis?? avec prise en compte de l'autonomie. 0- Quitter. Votre choix : 2 Le graphe n'est pas connexe (23 composantes) Construction du graphe des stations Graphe termine
MENUMENU 1- Charger le Graphe. 2- V??rification de la connexit??. 3- Calcul du plus court chemin simple. 4- calcul du plus court chemin optimis??. 5- calcul du plus court chemin optimis?? avec prise en compte de l'autonomie. 0- Quitter. Votre choix :

Figure 4 : Vérification de la connexité

🗙 xterm	
La Roche	
Leuville-sur-Orge	
Linas	
Linas	
Ballainvilliers	
Ballainvilliers	
Champlan	
Massy	
Massy	
Le Petit Massy	
Le Petit Massy	
Le Petit Massy	
Le Fetit Massy	
La croix-de-berny	
Hrcuell Oneuril	
Arcuell Contillu	
Contillu	
Centillu	
Pario	
478 Etapes - 645 3251587233359Km	
410 Ecapes 043,02010072000000	
Le calcul a dure 5450ms	
1- Charger le Graphe.	
2- V??rification de la connexit??.	
3- Calcul du plus court chemin simple.	
4- calcul du plus court chemin optimis??.	
5- calcul du plus court chemin optimis?? avec prise en compte de l'au	utonomie.
0- Quitter.	
Votre choix :	





Ville de d??part : Paris Ville d'arriv??e : Toulouse Paris La Corbilliere Sainte-Radegonde-en-Touraine Le Miosson Riviere Brie Saint-Christophe Estillac Toulouse 8 Etapes - 779.5043334960938Km Le calcul a dure Oms --MENU 1- Charger le Graphe. 2- V??rification de la connexit??. 3- Calcul du plus court chemin simple. 4- calcul du plus court chemin optimis??, 5- calcul du plus court chemin optimis?? avec prise en compte de l'autonomie. 0- Quitter. Votre choix :

Figure 7 : Paris->Toulouse avec contrainte

Cette interface est assez intuitive et permet d'exploiter directement le graphe, néanmoins elle ne permet pas de dessiner le graphe. Nous aurions pu faire appel à la fonction dessinerGraphe que nous avons implémenté mais celle-ci génère une image gif que l'utilisateur doit regarder de lui-même. Pour se faire, il suffit qu'il exécute cette fonction puis si il a demandé à utiliser grapheviz, une image « Graphe.gif » est alors automatiquement crée dans le répertoire « Src », sinon avec la méthode utilisant neato, le nom de l'image est « carte.jpg ». On peut faire appel à la méthode dessinerGraphe dans la classe Progmain.java.

```
"Teppe" [pos=" 937, 4213"];
"Casevecchie" [pos=" 936, 4213"];
"Casevecchie" [pos=" 935, 4214"];
"Pointe Capizzali" [pos=" 932, 4214"];
"Pointe Chigliani" [pos=" 932, 4215"];
"Pointe Chigliani" [pos=" 931, 4216"];
"Ruisseau Saparelle" [pos=" 929, 4215"];
"Pointe Muracinto" [pos=" 927, 4217"];
"Pointe Muracinto" [pos=" 926, 4217"];
"Pointe Scandola" [pos=" 925, 4218"];
"Rospigliani" [pos=" 922, 4219"];
"Col de Morello" [pos=" 920, 4219"];
"Col de Morello" [pos=" 920, 4219"];
"Noceta" [pos=" 920, 4220"];
"Ruisseau de Quercio Grosso" [pos=" 919, 4221"];
"Ruisseau de Quercio Grosso" [pos=" 919, 4221"];
"Guiselle" [pos=" 919, 4222"];
"Guiselle" [pos=" 919, 4222"];
"Guiselle" [pos=" 919, 4222"];
"Guiselle" [pos=" 918, 4222"];
"Guiselle" [pos=" 918, 4223"];
"Guiselle" [pos=" 917, 4222"];
"Punta Chiova" [pos=" 918, 4214"];
"Punta Chiova" [pos=" 918, 4213"];
"Punta Chiova" [pos=" 918, 4212"];
"Punta Chiova" [pos=" 917, 4211"];
"Punta Chiova" [pos=" 918, 4211"];
"Punta Chiova" [pos=" 919, 4211"];
"Mont Calvi" [pos=" 920, 4210"];
"Mont Calvi" [pos=" 920, 4210"];
```

Figure 8 : exemple d´une partie de la génération automatique du fichier dot pour le dessin du graphe