

CAHIER DES CHARGES

SAE – GRAMA : Graph MAP Analysis

Nom de l'établissement : IUT Lyon 1

Nom du projet : Graph MAP Analysis

Responsables du projet :

- IDRISSE SALL
- ARO RANDRIAMANANTENA

Adresse établissement : 92 Bd Niels Bohr, 69100 Villeurbanne

Téléphone :

- 04 72 69 20 00 (IUT Lyon 1)
- 07 84 89 28 50 (ARO RANDRIAMANANTENA)
- 07 63 48 56 86 (IDRISSA SALL)

Email :

- aro.randriamanantena@etu.univ-lyon1.fr (ARO RANDRIAMANANTENA)
- idrissa.sall@etu.univ-lyon1.fr (IDRISSA SALL)

SOMMAIRE

1-PRÉSENTATION DE CONTEXTE

1-1-Graphe

1-1-1- Non-orienté

1-1-2- Pondérée

1-1-3- Connexe

1-2- JAVA

2- PRÉSENTATION DU PROJET

2-1- Contexte

2-2- Analyse de l'existant

3- PRESTATIONS ATTENDUES

3-1- Création de contenus

3-1-1- Contraintes :

3-1-2- Structure des données

3-2- Développement d'une interface IHM :

3-2-1- Contraintes

3-2-2- Fonctionnalités

3-2-3- Résultats et sorties IHM

4- RÉSULTAT ATTENDU

1 - PRÉSENTATION DE CONTEXTE

1 - 1 - Graphe

1 - 1 - 1 - Non-orienté

Un graphe non orienté est la donnée d'un ensemble non vide de sommets, d'un ensemble d'arêtes et d'une application.

1 - 1 - 2 - Pondérée

Un graphe est dit pondéré lorsque le poids de chaque arête (non orienté) ou de chaque arc (orienté) du graphe est un nombre positif.

1 - 1 - 3 - Connexe

Un graphe est dit connexe s'il existe un chemin ou une chaîne, c'est-à-dire une suite d'arêtes ou d'arc, reliant toujours deux sommets quelconques du graphe.

1 - 2 - JAVA

JAVA est un langage de programmation créé en 1995 par James Gosling et Patrick Naughton orienté objet. L'application à rendre doit être programmée dans ce langage de programmation.

2 - PRÉSENTATION DU PROJET

Dans le cadre de notre formation, il nous été demandé de réaliser le projet SAE GRAMA, une application programmée dans le langage Java capable d'analyser un graphe donné en fichier CSV ou fichier texte. Ce projet est encadré par les professeurs de GPI, Math, Java, SDD, IHM et Qualité dev de l'IUT Lyon 1.

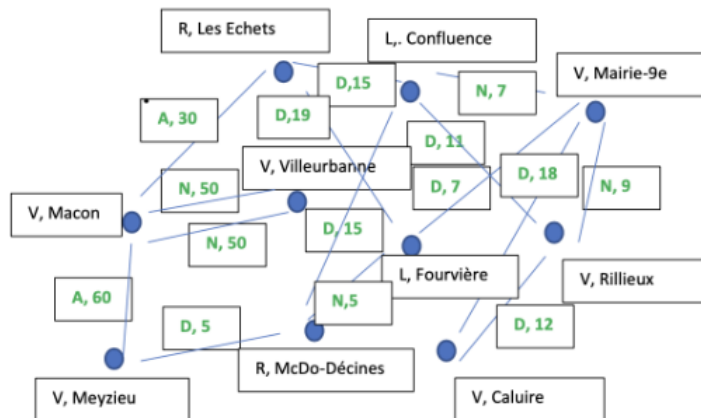
Les diverses fonctionnalités de l'application seront précisées dans la partie fonctionnalité du cahier des charges.

3- PRESTATIONS ATTENDUES

3-1- Création de contenus

Construire un Graph-MAP contenant les deux villes de naissance ou résidences des deux binômes. Ce dernier doit être non orienté en arêtes en kilométrage et contient trois types de nœuds (Villes : V , Loisir: L et Restaurant: R) ainsi que trois types d'arêtes (Autoroute: A , Nationale : N et Départementale: D).

exemple :



3-1-1- Contraintes :

Le résultat doit être connexe et doit contenir au moins 30 nœuds en respectant les contraintes suivantes:

- Nœuds : 3/5 de villes, 1/5 de restaurants et 1/5 de centres de loisirs.
- Liens : 1.5/5 d'autoroutes, 1.5/5 de nationales et 2/5 de départementales.
- Possibilités d'avoir plusieurs liens (A, N et/ou D) entre deux mêmes villes.
- Le Graphe MAP doit être chargé en mémoire dans une structure de données dynamique (listes).
- Le Graphe MAP doit être donné (en dessin) dans le rapport final.

3-1-2- Structure des données :

Le Graphe-MAP doit être donné en fichier CSV ou texte où chaque ligne est décrite ainsi :

Nœud:lien1::voisin1; lien2::voisin2;lien3::voisin3;.....;lienk::voisink;;

Exemple :

V, Macon: A,30::R,Les Echets;N, 50::V, Villeurbanne;N,50::V, Villeurbanne;A,60::V,Meyzieu;;

V,Meyzieu:A,60::V,Macon;D,5::McDo-Decines;;

R,McDo-Decines:D,5::V,Meyzieu....

3-2- Développement d'une interface IHM :

3-2-1- Contraintes

- L'exploration et l'analyse du Graph-MAP se fera via l'interface IHM.
- Seul le langage JAVA doit être utilisé pour développement de l'interface IHM

3-2-2- Fonctionnalités

Depuis l'interface IHM, le livrable doit être capable de répondre à certaines interrogations que sont:

- Charger les fichiers
- Visualisation du graphe
- Analyser les éléments du graphe (0-distance) : Donner la possibilité d'afficher ces éléments à la demande sur l'interface IHM :
 - Villes, centres de loisirs et restaurants (NŒUDS),
 - Autoroutes, nationales et départementales (LIENS)
 - Liste par catégories NŒUDS et LIENS.
 - Nombres : villes, de restaurants, de centres de loisir, d'autoroutes, de nationales, de départementales dans le Graphe - MAP
 - Pour un nœud donné, possibilité de connaître les voisins à 1-saut (voisin direct) :

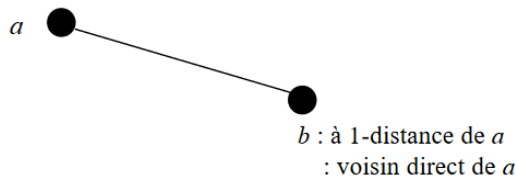
exemples :

Quels sont les restaurants voisins à Décines (voisins directs) ?

Quels sont les centres de loisirs voisins aux Echets (voisins directs) ?

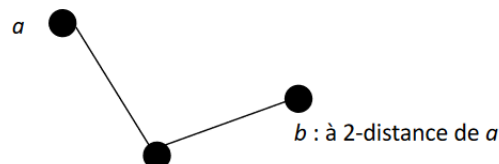
- Pour un lien/arête donné, possibilité de connaître les villes qu'il relie

exemple :



- Connaître les distances entre deux sites (2-distances) : On choisit 2 sites quelconque du graphe sur l'IHM, l'application doit pouvoir dire s'ils sont à 2-distance ou pas.

exemple :



- Comparer deux villes (≥ 2 distance) : Pour 2 villes quelconques A et B sur l'IHM, l'application doit pouvoir dire si :
 - A est plus/moins OUVERTE que B : Si le nombre de villes à 2-distance de A est sup/inf au nombre de villes à 2-distance de B

- A est plus/moins GASTRONOMIQUE que B : Si le nombre de restaurants à 2-distance de A est supérieur/inférieur au nombre de restaurants à 2-distance de B
- A est plus/moins CULTURELLE que B : Si le nombre de centres de loisir à 2-distance de A est sup/inf au nombre de restaurants à 2-distance de B
- Connaître les distances entre deux sites (p-distance) : Étant donnés 2 sites quelconques du graphe via l'IHM, l'application doit être en mesure de :
 - calculer la distance la plus courte entre eux
 - trouver une route traversant, une ville, un lieu de loisir, un restaurant voire plus, passant par 2 villes, un restaurant et un centre de loisir, 2 centres de loisir,...

3-2-3- Résultats et sorties IHM

Écran principal	Chargement du Graphe-MAP et visualisation du graphe
Écran 0	Comprendre les contextes des éléments du graphe
Écran 1	Connaître les distances entre deux sites (1-distance)
Écran 2	Connaître les distances entre deux sites (2-distance)
Écran 3	Comparer deux villes
Écran 4	Connaître les distances entre deux sites

4- RÉSULTAT ATTENDU

- Une interface IHM avec six écrans. Un écran principal pour le chargement du Graph-MAP, l'écran 0, l'écran 1 et l'écran 2 représentant respectivement affichage 0-distance, voisinage direct (1-distance) et voisinage à deux sauts (2-distance). A cela s'ajoutent deux autres écrans, l'écran 3 pour comparaison de sites à deux sauts ou plus (≥ 2 -distance) et l'écran 4 qui indiquera les voisinages à p sauts et distances (p-distance).
- Le cahier des charges du projet
- Le dossier de gestion projet
- Le dossier de spécification fonctionnelle de l'application

- Le dossier de spécification technique de l'application