|  |
| --- |
| SAE - GRAMA  GRAph Map Analysis  **Cahier des charges**  *Rédigé le 08/02/2022 par BOUCQUEZ Maxence & VOILLOT Antoine*  *Dernière mise à jour : 15/06/2022* |

## Sommaire

[**Sommaire**](#_x4itn8859lhp)2

[**I- Descriptif du projet**](#_b1vbiyjltsep)3

[**II- Equipe du projet**](#_be5j6h948i4s)3

[**III- Contexte du projet**](#_idkfxeuskdtz)3

[1 - Exposé de la situation](#_ytilimso4sce) 3

[**IV- Description fonctionnelle des besoins**](#_e6xpxjgnm3n2)5

1 – Stocker les données dans un graphe **Erreur ! Signet non défini.**

[Objet](#_46v9jp38maxh) Nœud 5

Objet Lien 5

Objet Graphe 5

2 - Chargement d’un fichier csv en graphe 5

3 - Affichage graphique du graphe **Erreur ! Signet non défini.**

[Affichage](#_46v9jp38maxh) Général 5

Affichage personnalisé 5

Affichage des 1-voisins d'un nœud 5

Affichage des 2-voisins d'un nœud 5

4 – Analyse 6

[Comparer](#_46v9jp38maxh) 2 nœuds 6

Vérifier si 2 nœuds sont à 1 de distance 6

Vérifier si 2 nœuds sont à 2 de distance 6

Calcul du plus court chemin 6

[**V- Sécurité**](#_o8xne7sr0pyi) **Erreur ! Signet non défini.**

[**VI- Budget**](#_ievy8j9uaz6q) **Erreur ! Signet non défini.**

[**VII - Calendrier**](#_xtnphyicac2x)6

## 

## 

## 

## 

## I- Descriptif du projet

Le projet à pour but de représenter un fichier csv contenant des nœuds et des liens sous la forme d’un graphe au sein d’une IHM programmer en JAVA. Il contiendra également des fonctionnalités propres au graphe tel que l’affichage des voisins ou la comparaison de nœud.

## II- Equipe du projet

### 

|  |  |  |
| --- | --- | --- |
| Nom – Prénom | Rôle projet | E-mail de contact |
| BOUCQUEZ Maxence | Développeur | maxence.boucquez@etu.univ-lyon1.fr |
| VOILLOT Antoine | Développeur | antoine.voillot@etu.univ-lyon1.fr |

## 

## III- Contexte du projet

Dans le cadre de nos activités, au sein des modules de graphe, de programmation (IHM, Java, SDD), de qualité développement et de gestion de projet nous avons été amenés à réaliser le projet *GRAph Map Analysis*. Ce dernier ayant pour but de digitaliser la représentation d’un graphe entre des lieux avec pour seul base de données un fichier csv fourni par l’utilisateur.

Pour atteindre cet objectif, nous envisageons de mettre à disposition une application java disposant d’une interface graphique pour simplifier les actions de l’utilisateur.

#### 1 - Exposé de la situation

Les principaux axes de développements pour répondre à ces problématiques sont :

* Répondre au problème sur l’aspect backend
* Répondre au problème sur l’aspect frontend
* Optimiser notre application

## IV- Description fonctionnelle des besoins

#### 1 – Stocker les données dans un graphe

##### **Objet Noeud :** Afin de stocker les détails d’un nœud (nom et type)

##### **Objet Lien :** Afin de stocker les détails d’un lien (nœuds qu’il relie, distance, type)

##### **Objet Graphe :** Afin de représenter un graphe composé d’objets lien et nœuds.

#### 2 - Chargement d’un fichier csv en graphe

Le GRAph Map doit être donné sous forme de CSV ou txt pour être manipulé. Il faut donc créer une fonction capable de lire le contenu du fichier et de le charger en mémoire dans un objet de type graphe. Les lignes du fichier seront alors formaté selon la norme suivante : **Noeud:lien1::voisin1;lien2::voisin2;...;lienk::voisink**

#### 3 - Affichage graphique du graphe

##### **Affichage Général :** Afficher le graphe sous forme visuel avec des ronds pour les nœuds (Différencié par des couleurs) et des trais pour les liens.

##### **Affichage personnalisé :** Possibilité d’afficher ou cacher certains nœuds ou liens.

##### **Affichage des 1-voisins d’un nœud :** Permettre de n’afficher que les 1 voisins du nœud.

##### **Affichage des 2-voisins d’un nœud :** Permettre de n’afficher que les 1 voisins du nœud.

##### **Affichage des 2 nœuds que relie un lien :** Permettre de n’afficher que les 2 nœuds que relie un lien.

#### 4 – Analyse

##### **Comparer 2 nœuds : Comparer si un nœud est plus ouvert/gastronomique/fun qu’un autre nœud.**

##### **Vérifier si 2 nœuds sont à 1 de distance**

##### **Vérifier si 2 nœuds sont à 2 de distance**

##### **Calculer le plus court chemin entre 2 nœuds**

## VII - Calendrier

### 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Semaine 1 | Semaine 2 | Semaine 3 | Semaine 4 | Semaine 5 | Semaine 6 |
| [Phase] | **[Phase]** | **[Phase]** | **[Phase]** | **[Phase]** | **[Phase]** |
| *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* |
| *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* |
| *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* | *[Tâche]* |

## 