Rapport bilan de projet sur la théorie des graphes

**Fête Des Lumières à Lyon**

Ce projet de fin de semestre marque la fin des cours et des TD d’application sur les différents algorithmes de parcours des graphes en informatique. Notre groupe est composé de Thomas PARRAUD, Oscar BURDIN et Samy KNEFATI.

Il s’agit ici de concevoir une application qui aura pour but de guider les visiteurs à travers différents parcours en fonction de leur position à un événement. L’application prendra également en compte les visiteurs déjà présents sur les différentes voies de passage afin de réguler le trafic dans la ville. Cette application sera utilisée lors de la Fête des Lumières pour l’année 2020. L’utilisateur possédant l’application pourra obtenir toutes les informations nécessaires sur les différents événements et sera informé en temps réel de l’affluence des rues afin d’avoir une visite de la ville la plus agréable possible.

De plus cette application sera en lien direct avec les services de secours et la police afin d’avoir un maximum de sécurité sur toute la zone où seront déployées les différentes attractions de la ville.

Elle sera également munie d’un plan de la ville détaillé permettant à l’utilisateur de suivre son chemin sans souci et lui permettre de se déplacer dans la ville avec les différents métro ou bus selon la zone où il se situe.

Table des matières

[Les différentes classes du code 3](#_Toc38572745)

[GitHub 4](#_Toc38572746)

[Frise chronologique du projet 4](#_Toc38572747)

[Le fonctionnement de l’application 5](#_Toc38572748)

[Allegro:  Allegro Low LEvel Game ROutines 6](#_Toc38572749)

[Les différents parcours de la Fête des Lumières 7](#_Toc38572750)

[Les difficultés rencontrées dans le projet 10](#_Toc38572751)

# Les différentes classes du code

Pour la conception de ce projet, nous avons travaillé sur les bases de ce que nous avions vu au préalable lors des nombreux travaux pratiques du début du semestre sur la théorie des graphes. Pour les classes utilisées dans notre code, nous avons décidé de prendre des classes pour les arêtes, les sommets et enfin le graphe qui regroupe les deux autres.

Le schéma des classes est le suivant :

* Les arêtes possèdent :
  + Un départ et une arrivée (arrivant aux sommets)
  + Un type (si c’est une ruelle, une avenue ou autre)
  + Un nom
  + Un flot maximum appelé capacité
  + Un flot à l’instant t
  + Une distance (entre A et D)
  + Un booléen pour savoir si oui ou non telle arête est utilisée dans le graphe
* Les sommets possèdent :
  + Une coordonnée en x et en y pour les situer sur le plan
  + Un nom
  + Un numéro
  + Une liste de successeurs
  + Un booléen pour savoir si le sommet est utilisé
* Le graphe possède :
  + Une capacité totale
  + Une liste de sommets
  + Une liste d’arêtes

Ces trois classes vont être utilisées dans l’intégralité du projet. Les différentes méthodes qui les composent seront présentées plus tard dans le code lors de l’explication des algorithmes utilisés.

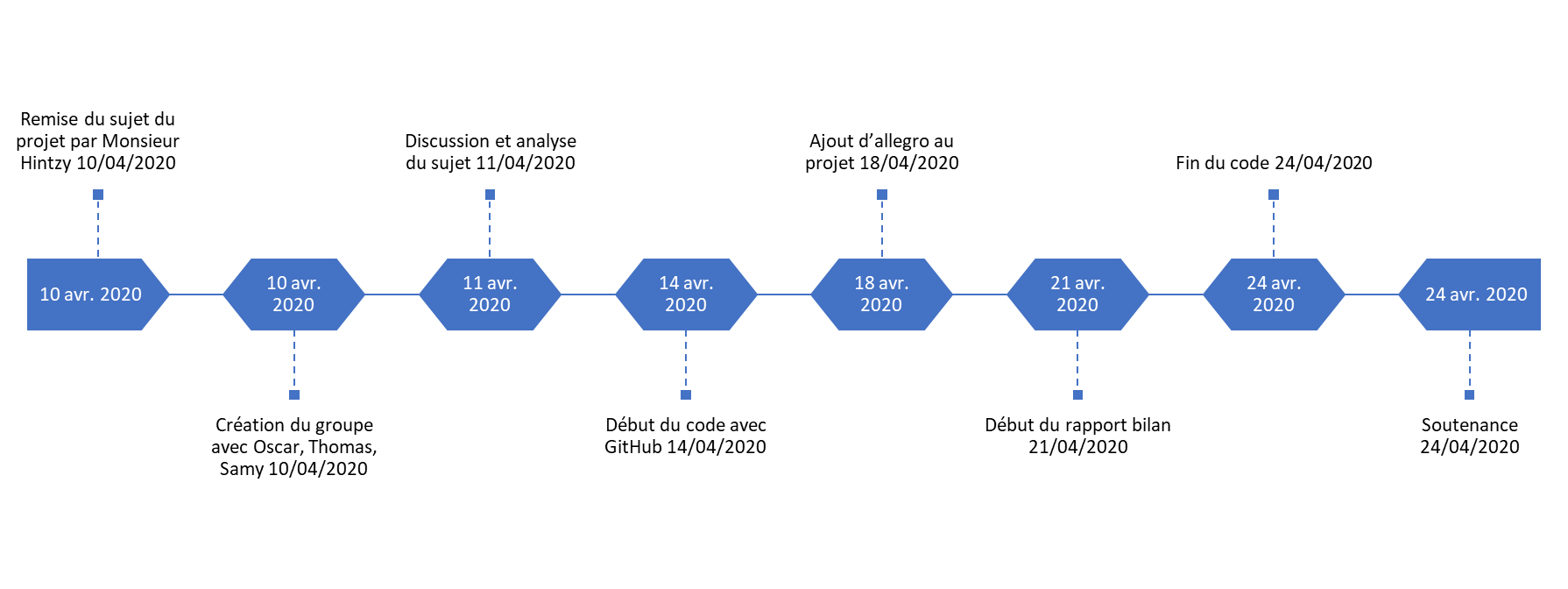
# GitHub

Pour le partage du code et la modification de celui-ci, nous avons créé un repository sur GitHub. Après avoir ajouté l’ensemble du groupe et fait les ajouts du code de base que nous avions, nous avons commencé le code de manière séparée et individuelle.

Github nous a permis de centraliser le code sur un seul serveur et également de voir l’avancement de chacun dans le code. Quelques problèmes sont apparus à la mise en place du système mais tout a été réglé par la suite.

Le lien de notre repository est le suivant : <https://github.com/SamyKnefati/Projet-Info-S2.git>

# Frise chronologique du projet



Cette frise montre l’avancement sur ce mois d’avril du projet. Elle contient les différentes phases principales qui ont fait naître cette application.

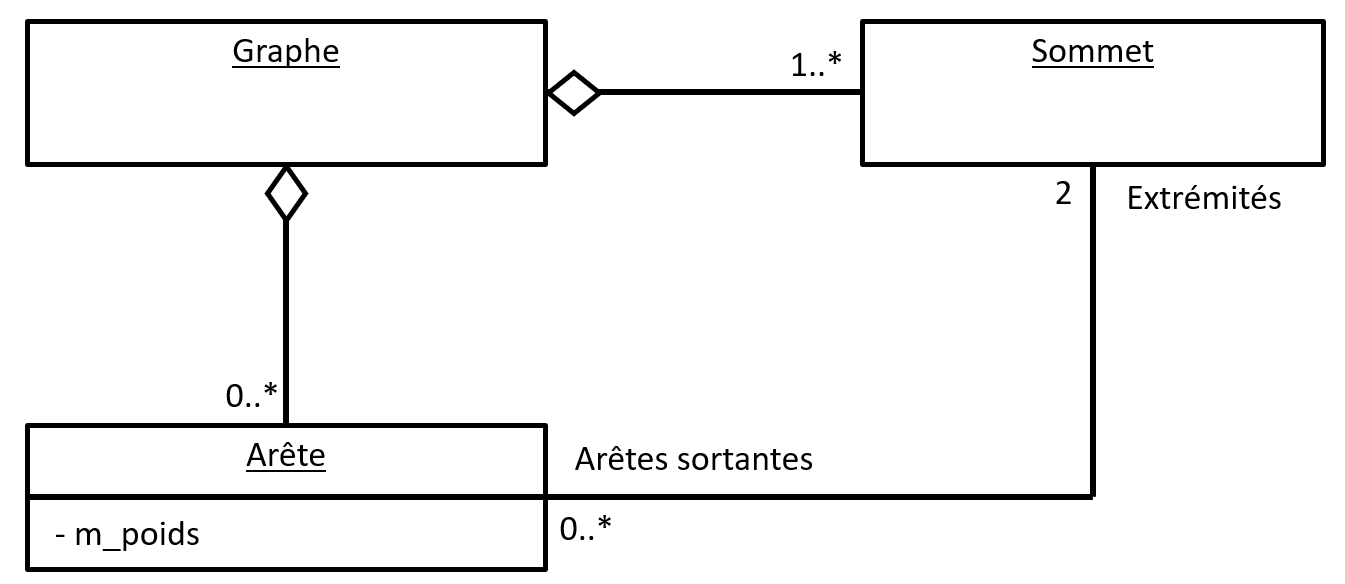
# Le fonctionnement de l’application

Nous avons cherché à développer une application qui pourra être utilisée par un visiteur lambda mais également par le personnel de sécurité et le personnel qui sera sur le terrain pour guider les touristes.

Cette application sera composée d’une carte de la zone avec l’affichage de tous les parcours possibles pour l’utilisateur. Elle comportera aussi des indications dans les différentes zones de Lyon avec les affluences en temps réel. La sécurité et les personnes sur le terrain seront capables de traiter des problèmes d’affluences directement sur l’application. La police pourra si besoin fermer une route, qui apparaitra sur la carte et l’organisation de la fête sera modifiée en fonction.

La zone de la fête des lumières se déroulant sur plusieurs rives et quartiers de la ville, nous avons choisi une carte du centre de la presqu’île et les alentours adéquates.

En ce qui concerne les différents algorithmes que nous utilisons dans l’application, une partie spécialement dédiée arrivera plus tard dans le rapport.



Ce diagramme montre le fonctionnement d’un graphe avec des arêtes pondérées, comme celui que nous utilisons dans le code pour l’application.

# Allegro:  Allegro Low LEvel Game ROutines

Pour la partie affichage de notre projet de Fête des Lumières, nous avions le choix entre deux manières différentes de générer un rendu.

La première proposition faite par les professeurs était d’utiliser un affichage SVG avec un rendu simple montrant le graphe avec les parcours choisis par l’utilisateur.

Ensuite, nous avions Allegro, qui permet un affichage plus complexe avec de meilleurs résultats en termes de beauté d’image. Cependant la mise en place de cette bibliothèque est plus compliquée.

Nous avons donc choisi la manière Allegro pour donner vie à notre application. Les différentes fonctions disponibles dans la bibliothèque nous a permis de développer la complexité de l’affichage afin de rendre cette application utilisable pour le grand public. Pour remettre les choses en perspective, Allegro est une bibliothèque logicielle libre dédiée à la création de jeux vidéo. Dans notre cas, il s’agit d’un affichage plus simple avec des animations rapides montrant les différents parcours que nous avons défini.

Une image contenant carte

Description générée automatiquement

Cette capture d’écran montre les nombreuses variables affichées sur l’application lorsque l’utilisateur cherche à se repérer dans la ville.

# Les différents parcours de la Fête des Lumières

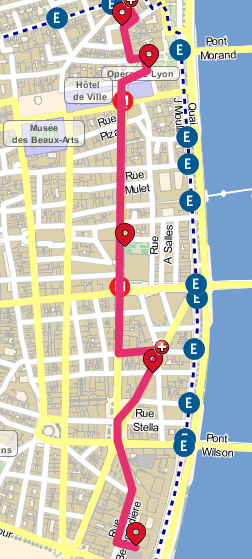
Lors des recherches sur la Fête des Lumières nous avons relevé quatre parcours à travers la ville. Nous avons donc choisi de nous inspirer des parcours existants afin d’avoir une meilleure idée de ce qui pourrait intéresser les futurs visiteurs de Lyon.

Les quatre parcours sont les suivants :

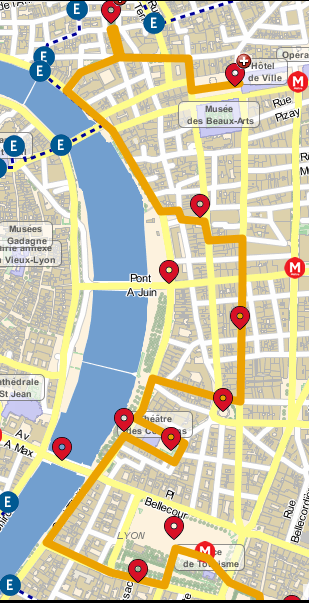
* Le parcours bleu de 2.8km :



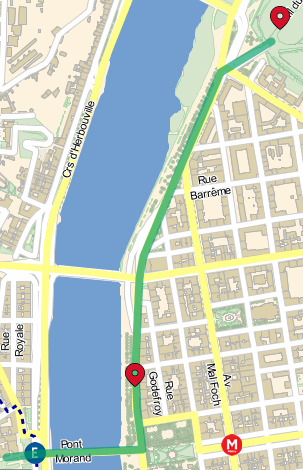
* Le parcours rose de 1.6km :



* Le parcours orange de 3.4km :

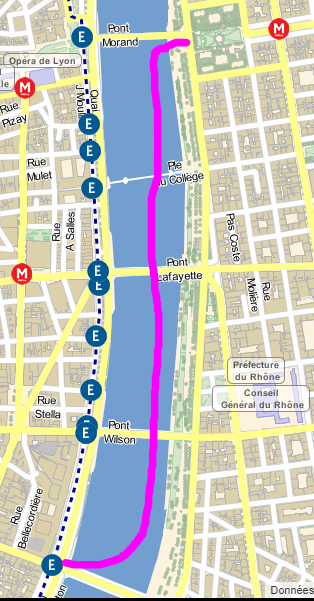


* Le parcours vert de 1.6km



Afin d’ajouter une nouvelle expérience pour les visiteurs de la ville, nous avons pensé à ajouter un programme qui pourrait être appréciable.

Il s’agit du parcours fuchsia de 1.4km sur le Rhône :



Cet ajout a également un intérêt dans le code car son trajet peut être représenté par une arête où la capacité maximale sera la capacité du bateau. Ce trajet sera possible seulement avec le flot maximal, donc quand l’embarcation sera remplie. Nous rajoutons ici une attraction à la Fête des Lumières pour la moderniser et apporter un plus au code.

# Les difficultés rencontrées dans le projet

Lors de ce projet nous avons rencontré trois soucis majeurs qui nous ont suivi sur l’ensemble de ces deux semaines. Tout d’abord le sujet nous a plu et nous avons été trop enthousiaste. Nous avions énormément d’idées concernant le sujet proposé, peut-être même trop. Qu’il s’agisse des différents parcours, des endroits à ajouter et de la complexité de l’application, nous avions vu trop gros. En effet, les ajouts des idées dans le cahier des charges que nous nous étions imposés avaient rendu celui-ci irréalisable dans les délais donnés. Nous avons donc dû faire preuve de lucidité pour alléger la charge de travail et ainsi arriver au bout de nos objectifs.

Le deuxième souci qui nous a retardé au début de ce projet a été les différents problèmes liés à GitHub et aux premiers commit/push et après lorsqu’il fallait push les codes ajoutés. Ces problèmes ont été résolus assez rapidement mais ils ont tout de même fait perdre du temps aux membres du groupe. Le cours donné par Monsieur Hintzy a ce sujet a aussi permis d’éclaircir les zones d’ombre de Git et ainsi de plus perdre du temps à réparer les erreurs des membres du groupe. L’ajout du gitignore nous par exemple permis de ne plus se soucier de certains fichiers sensibles propres à chaque PC comme le Cmakelist par exemple.

Le troisième souci que le groupe a rencontré se révèle être allegro. Cette bibliothèque dédiée à la création de jeux vidéo que nous avons utilisé dans ce projet s’est avérée difficile à mettre en place. En effet, nous nous sommes confrontés à de nombreux bugs avec les libraires ainsi que des conflits entre MinGW et Allegro.

# Bilan collectif et individuel