

# Calcul d'itinéraire dans le métro Parisien

---

Projet Génie Logiciel

COMMA Ibrahima  
DIALLO Mamadou Abdoulaye  
MERDRIGNAC Thierry  
MINATCHY Jonathan  
NGATCHA Guy Armand

2010/2011

M1 MIAGE

# Introduction

---

Les années précédentes, nous nous sommes initiés à des langages de programmation tels que Java, C. Ce semestre, dans le cadre du cours intitulé « Génie Logiciel », nous sommes appelés à analyser et implémenter en Java un programme de gestion de transport urbain.

Par le biais des méthodes d'analyse, de modélisation, nous dégagerons les classes nécessaires à la réalisation du logiciel. Les tests unitaires avec l'outil JUNIT de Netbeans, nous permettront de réaliser les tests pour chaque méthode et donc, de tester l'ensemble des fonctionnalités de notre logiciel.

Grace à GIT, un logiciel de gestion de version décentralisée, nous présenterons les différentes versions et les contributions de chaque membre du groupe.

# Analyse et conception

---

La recherche du plus court chemin est un problème assez récurrent dans l'implantation et la mise en place de réseaux de communication, réseau routier, réseaux de télécommunication, réseaux internet. Dans ce projet, nous avons comme objectif de réaliser un système qui gère le réseau du métro parisien. Le système doit assister l'utilisateur à retrouver le chemin qui mène à sa destination en fonction de du mode de recherche qu'il aura choisi. Après une analyse approfondie, nous avons mis en place un diagramme de classes initiales pour l'implémentation du réseau.

## Utilisation de Git dans le projet :

### -Logiciels utilisés :

Nous avons utilisé Git sous Linux et en utilisant le logiciel *msysgit* pour Windows.

### -Création d'un dépôt sous GitHub :

Après avoir créé un compte sous Github, nous avons créé un projet dont l'adresse est :

[https://github.com/john974/M1\\_MIAGE\\_GIT6\\_Metro\\_Project](https://github.com/john974/M1_MIAGE_GIT6_Metro_Project)

### -Création du projet avec Git :

Nous avons créé le dossier M1\_MIAGE\_GIT6\_Metro\_Project dans lequel nous avons placé les sources, puis nous avons fait :

```
git init
```

### -Récupération du projet :

On peut récupérer le projet en ligne en tapant :

```
git clone git@github.com:john974/M1_MIAGE_GIT6_Metro_Project.git
```

**-Les membres qui ont participé au projet ont les pseudos suivants sous Git :**

-comma058@gmail.com (Comma)

-Diallo (Diallo)

-Merdrignac (Merdrignac)

-johh974 (Minatchy)

-KmerSoft (Ngatcha)

**-Utilisation de GIT pour le projet :**

Nous avons utilisé un peu toutes les fonctionnalités de GIT (add, commit, pull, checkout, merge) pour faire évoluer notre projet jusque sa version 1.9.

Toutes les actions effectuées sont visibles en tapant :

```
gitk --all
```

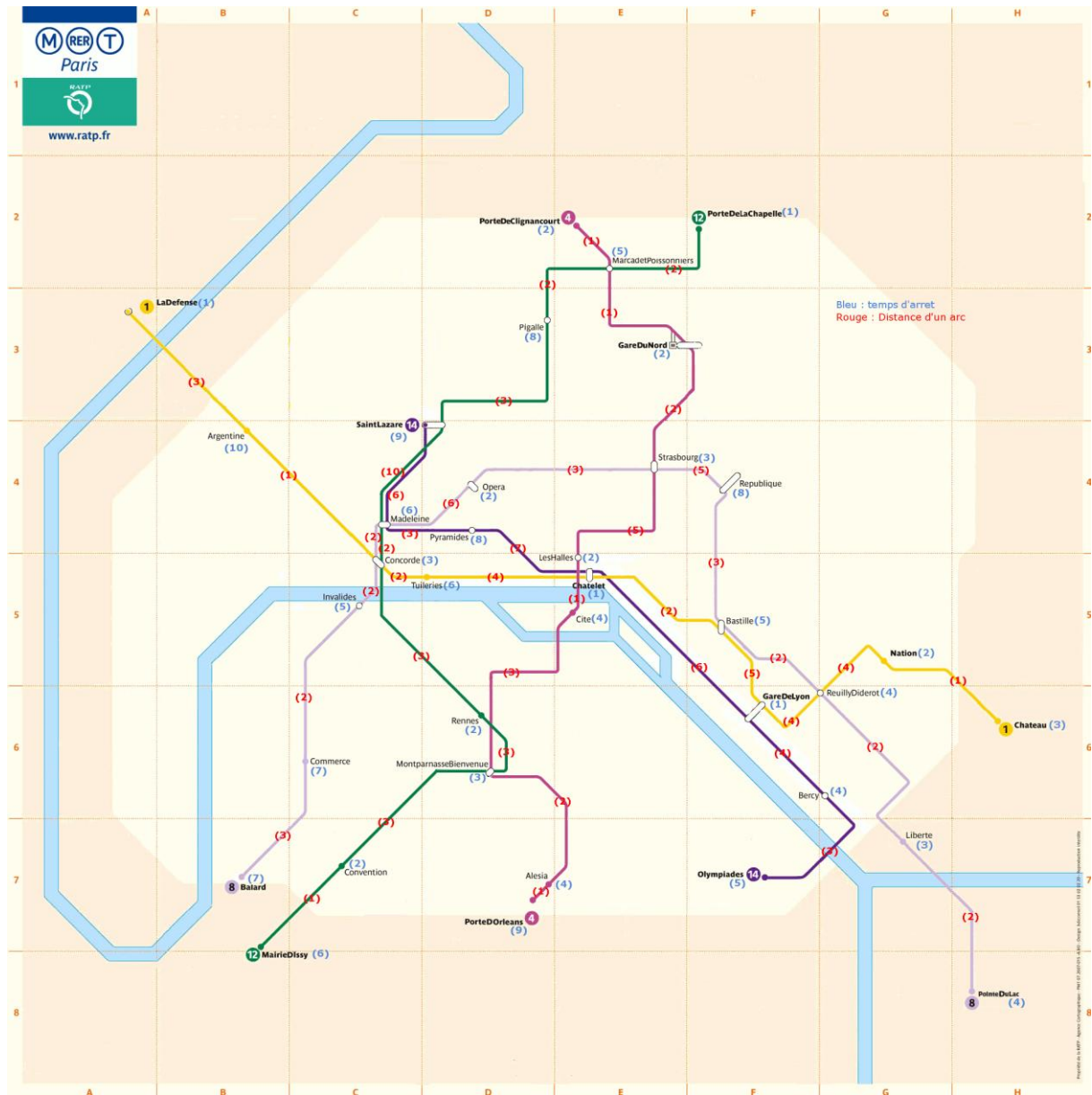
## Réseau test

Pour simplifier le projet, nous avons décidé de réduire le réseau. Pour cela nous avons supprimés certaines lignes ainsi que certaines stations afin de pouvoir tester notre projet sur un réseau moins conséquent. Le nouveau réseau n'est donc composé maintenant que de 5 lignes au lieu de 16 et de 36 stations au lieu de 300 :

- Ligne 1: La Défense - Château de Vincennes  
Composé plus que de 10 stations
- Ligne 4 : Porte de Clignancourt - Porte d'Orléans  
Composé plus que de 9 stations
- Ligne 8 : Balard - Créteil Pointe du Lac  
Composé plus que de 12 stations
- Ligne 12: Porte de la Chapelle - Mairie d'Issy  
Composé plus que de 10 stations
- Ligne 14 : Saint Lazarre - Olympiades  
Composé plus que de 7 stations.

Les nombres entre parenthèses sur le schéma ci-dessous nous renseigne sur 2 choses :

- **En bleu** : Temps d'arrêt : C'est le temps d'arrêt d'une rame à une station
- **En rouge** : Distance d'un arc : C'est le temps que met une rame pour aller d'une station à la station suivante.



## Réalisation et tests

---

Nous avons effectué le travail par petit groupe. L'équipe a été divisée par binôme. Chaque binôme avait une tâche bien précise. Après chaque grande modification, le binôme effectue un commit et met le dépôt distant à jour afin que les membres puissent intégrer les modifications à leur dépôt local.

1)- le Parcours le chemin le plus Rapide (choix 1) dans la console

Dans ce parcours nous avons implanté des algorithmes qui permettent à l'utilisateur de saisir la station de départ, la station d'arrivée et affiche le Trajet en indiquant la liste des station par lesquelles il doit passer puis les lignes

Exemple :

Pour un trajet de "Argentine" à "Concorde" le programme affiche :

Depart: Argentine

Arrivée: Balard

Chemin: Argentine-Concorde-Invalides-Commerce-Balard

Lignes à emprunter:

Ligne 8: Concorde-->Invalides-->Commerce-->Balard-->

Cout: 23.0

2)- Le parcours avec le moins de changement (choix 2)

Dans ce choix l'utilisateur fournit la station de départ et celle d'arrivée

Le programme lui retournera le trajet (cette fois-ci le programme affichera si c'est un trajet direct la/les ligne(s) permettant d'y aller ou bien les lignes et la station où il faudra changer de ligne

Exemple :

Pour un trajet de "Concorde" à "Balard" le programme retourne :

"TRAJET DIRECT Prenez la Ligne 8"

Pour un Choix de trajet entre "Argentine" et "Opera" le programme affiche:

*"TRAJET Prenez la Ligne 1, descendre à la Station Concorde  
puis prendre le Ligne 8"*

3)- Le parcours en passant par un endroit (station)

Dans ce choix l'utilisateur il fournit : la station de départ, les points par lesquelles il souhaite passer (donc il peut saisir une liste de station) et la station d'arrivée

Exemple d'exécution : avec pour Station intermédiaire "Opera"

Depart: Argentine

Arrivée:Nation

Chemin:Argentine-Concorde-Madeleine-Opera

Lignes à emprunter:

Ligne 8: Concorde-->Madeleine-->Opera-->

Cout: 18.0

Chemin:Opera-Strasbourg-Republique-Bastille-ReuillyDiderot-Nation

Lignes à emprunter:

Ligne8:Opera-->Strasbourg-->Republique-->Bastille-->ReuillyDiderot-->

Ligne 1: ReuillyDiderot-->

Cout: 37.0

En plus de ces 3 options l'utilisateur à d'autre choix notamment :

Le choix 4 qui lui permet de visualiser le reseau

## Projet Génie Logiciel

Exemple d'exécution : pour le choix 4 qui permet d'afficher le réseau entier

\*\*\*\* L'ENSEMBLE DES STATIONS DU RESEAU REGROUPEES PAR LIGNE \*\*\*\*

Ligne : Ligne 1

Stations : [ReuillyDiderot][Bastille][Concorde]

Ligne : Ligne 4

Stations : [PorteDeClignancourt][MarcadetPoissonniers][GareDuNord]  
[Strasbourg][LesHalles][Cite][MontparnasseBienvenue][Alesia]  
[PorteDOrleans]

Ligne : Ligne 8

Stations : [Balard][Commerce][Invalides][Concorde][Madeleine][Opera][Strasbourg]

[Republique][Bastille][ReuillyDiderot][Liberte][PointeDuLac]

Ligne : Ligne 12

Stations :  
[PorteDeLaChapelle][MarcadetPoissonniers][Pigalle][SaintLazare][Madeleine][Concorde][Rennes]  
[MontparnasseBienvenue][Convention][MairieDIssy]

Ligne : Ligne 14

Stations :  
[SaintLazare][Madeleine][Pyramides][Chatelet][GareDeLyon][Bercy][Olympiades]

Il existe également un cinquième choix qui permet une éventuelle modification des lignes et des stations.

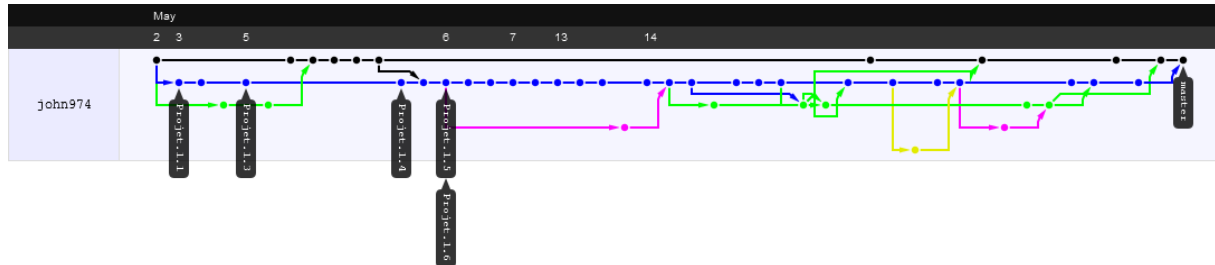
Ce programme n'est pas fonctionnel.

Un dernier choix permet de quitter l'exécution de l'application.



# Historique des Commit

---



## Conclusion

---

Partant de l'analyse, suivi d'une modélisation des différentes classes à implémenter, nous sommes rentres petit à petit dans l'implémentation du logiciel.

Il a donc été nécessaire d'utiliser les concepts et méthodes vues en cours et TD.

Ainsi de l'application de la méthode Agile, de l'utilisation des tests unitaires avec JUnit, de l'échange de version de fichiers avec GIT, nous nous sommes initiés aux techniques et méthodes de travail de futurs développeurs de logiciel.

Nous en tirons chacun, dans le groupe, un satisfecit personnel, tant sur le plan humain, car nous aurons développé notre esprit de travail en équipe que sur le plan professionnel.