14 Mai 2018

UNIVERSITÉ DE LORRAINE

Projet Génie Logiciel

-

Application de calcul d'itinéraire dans le métro parisien

Sprint n°5

8 au 14 Mai 2018

Equipe :
Rodolphe Aubry, Laurene Cladt, Charlotte Fernandes, Benjamin Rath

Enseignant:
Olivier Perrin

Sommaire

Sprint

Liste des tâches

Tâches	Estimation	Développeurs	Terminée	Temps réel
Gestion des stations en incident	0.5	Charlotte	✔	0.5
Indiquer le type d'itinéraire (rapide,points de passages...)	0.5	Benjamin	✔	0.5
Laisser l'utilisateur choisir des points de passages	0.5	Benjamin	✔	0.5
Gérer les incidents sur les lignes	0.25	Rodolphe	✔	0.1
Correction d'un problème au niveau des abus sur le temps de parcours	0.125	Rodolphe	✔	0.125
Afficher les lignes parcourues par le trajet choisi	0.25	Laurene	✔	0.125
Afficher l'itinéraire le plus rapide en temps	2	Laurene	✔	1.5
Afficher l'itinéraire avec le moins de changements	1		X	

Afficher l’itinéraire en fonction des points de passage de l’utilisateur	1	Laurene	✔	0.5
Afficher la direction du métro	0.25	Laurene	✔	0.125
Afficher les stations de changement	0.25	Laurene	✔	0.125

Gestion des stations en incident

Lorsqu'un utilisateur désire trouver un itinéraire, nous devons gérer si une station ou une ligne est en incident. Concernant les stations, nous avons décidé que si elle était en incident, l'itinéraire ne prendra pas en compte cette station.
Il est donc impossible pour l'utilisateur de sélectionner en départ, en arrivée, ou en point de passage une station en incident.
Si un incident est apparu sur une station de changement (où l'utilisateur devra changer de ligne pour atteindre sa destination), notre algorithme calculera un autre itinéraire.

Laisser l'utilisateur choisir des points de passages

Lorsqu'un utilisateur souhaite aller d'un point A à un point B, il se peut qu'il veuille passer par certains endroits plutôt que d'autre. Pour cela, il a la possibilité de choisir entre son départ et son arrivée, des endroits où passer.

Indiquer le type d'itinéraire (rapide,points de passages...)

Lorsque l'utilisateur souhaite effectuer un itinéraire, celui-ci peut avoir envie d'avoir un itinéraire rapide vers sa destination ou alors de choisir certains endroits où passer. Pour cela, après avoir choisi son départ et son arrivée, il devra ensuite spécifier quel type d'itinéraire il souhaite. Il peut soit ajouter des points de passages soit effectuer un itinéraire rapide vers son arrivée.

Gérer les incidents sur les lignes

Pour les lignes nous avons décidé que lorsqu'une d'entre elles est en incident, elle ne devait pas faire partie des lignes possibles lors d'un trajet.
Si une ligne est sous l'état incident alors l'algorithme d'itinéraire ne la prendra pas en compte.

Correction d'une possibilité d'abus sur les temps de parcours

Les temps de parcours lorsque créés via la méthode setTempsParcours pouvaient être négatifs.

Afficher l’itinéraire le plus rapide en temps

L’implémentation de l’itinéraire étant au coeur du projet, elle a demandé un certain temps de réflexion et d’implémentation. Nous avons décidé d’utiliser l’algorithme de Dijkstra pour trouver le plus court chemin entre deux stations et avons de ce fait implémenté la modélisation du réseau en temps que graphe. Nous avons ajouté les classes Arc, modélisant le bout de ligne reliant deux stations entre elles, Graphe représentant l’ensemble du réseau comme stations liées par des arcs, et Itinéraire permettant de calculer l’itinéraire entre deux stations grâce à l’algorithme de Dijkstra. Le contrôleur associé à la classe itinéraire permet de générer le type d’itinéraire voulu par l’utilisateur et de formater son affichage.

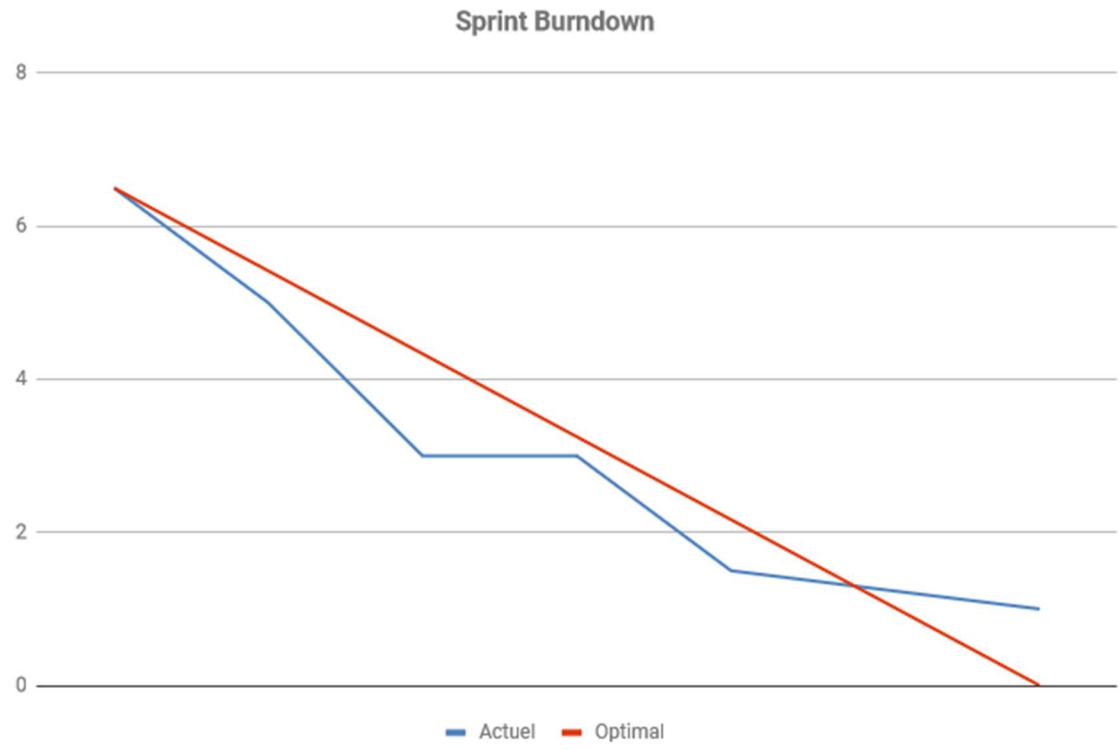
Afficher les lignes parcourues, les stations de changement et la direction du métro

De par l’implémentation de l’itinéraire, l’affichage des lignes parcourues, des stations de métro et de la direction s’est faite lors de la création des classes mentionnées ci-dessus. Il a fallu faire quelques ajustements, notamment au niveau du type de données renvoyées par les méthodes, afin de pouvoir formater l’itinéraire correctement et avec les informations nécessaires pour l’utilisateur.

Afficher l’itinéraire en fonction des points de passage définis par l’utilisateur

Afin de vérifier que la méthode d’implémentation de l’itinéraire était viable pour la suite, nous avons effectué quelques tests pour savoir si il était possible d’ajouter des points de passages. La méthode étant concluante, nous avons décidé d’implémenter cette fonctionnalité lors de ce sprint, quitte à devoir la peaufiner par la suite.

Burndown chart



Réunion post-sprint

Cette itération a été consacrée entièrement à la réalisation des itinéraires pour l'utilisateur. Toutes les tâches planifiées pour cette itération ont été remplies, cependant nous avons rencontré des difficultés pour une fonctionnalité : créer un itinéraire avec le moins de changement de lignes. L'algorithme que nous utilisons actuellement, ne nous permet pas de planifier ce type d'itinéraire. Nous essaierons de trouver une solution à la prochaine et dernière itération.

Cette itération est notre dernière ligne droite, elle nous permettra d'améliorer le code (refactoring), de revoir les tests réalisées, de trouver des bugs s'ils en restent et de tester notre application pour éviter toute surprise lors de notre présentation finale.