

Brooklyn

Titouan
LOCATELLI

Guillaume
MERCIER

Hamza
OUHMANI

2022



Cahier des charges du projet S4

Table des matières

1	Introduction	3
2	Présentation	3
3	Etat de l'art	5
4	Détails du projet	6
4.1	Etapes du projet	6
4.2	Distribution des tâches	7
4.3	Planning	7
4.3.1	Première soutenance, 14 fevrier	7
4.3.2	Deuxième soutenance, 4 avril	8
4.3.3	Soutenance Finale, 16 mai	8
4.4	Ressources	8
4.4.1	GTK	8
4.4.2	GitHub	8
4.4.3	Language de programmation :	8
5	Conclusion	9

1 Introduction

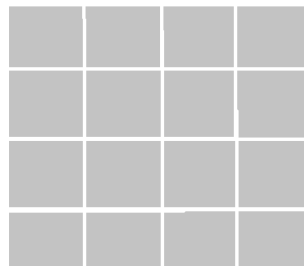
Nous sommes Guillaume Mercier, Hamza Ouhmani et Titouan Locatelli, le seul groupe à 3 personnes d'EPITA LYON pour ce projet de S4. La moitié de gens pour un projet et cependant nous allons tout de même faire un projet complet et fonctionnel ! Nous formons le groupe Brooklyn. Ce nom n'a rien à voir avec nos personnes ou goûts, mais avec notre projet, vous allez voir cela dans la suite.

De nos jours les gens ne savent plus lire une carte. Ils sont tellement dépendants de leurs téléphones que naviguer dans une ville avec un bout de papier à la main est une tâche périlleuse. Plusieurs y laisseraient leurs vies dans cette quête de retrouver son chez soi chaleureux, voir serais peut-être même être amené à sociabiliser en pleine rue et demander des directions, cet acte répugnant que nous n'avons pas vu depuis une vingtaine d'années. Ce serais de la régression technologique, quelle atrocité.

C'est pour cela que le jour où Google aura un crash mondial, que Google maps sera donc aussi offline (et Waze, et city mapper, et Maps.me et toutes les autres applications de cartes en ligne...), Nous sommes là à la rescousse avec notre version simplifiée de Google maps.

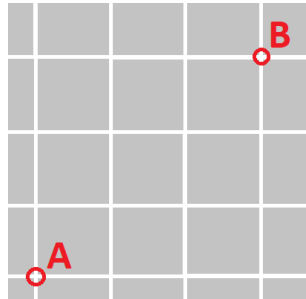
2 Présentation

Nous avons en tête d'utiliser les graphes pour naviguer sur une carte. Ce projet serait une version simplifiée de Google maps. Lorsque nous cherchons à nous rendre d'un point A à un point B, notre projet nous donne le chemin le plus rapide entre les 2. Nous allons simplifier Google maps (parce que tout d'abord des centaines d'ingénieurs ont travaillé des dizaines d'années sur la version de Google maps que nous connaissons de nos jours, et nous sommes que 3 sur 4 mois) de quelques manières :



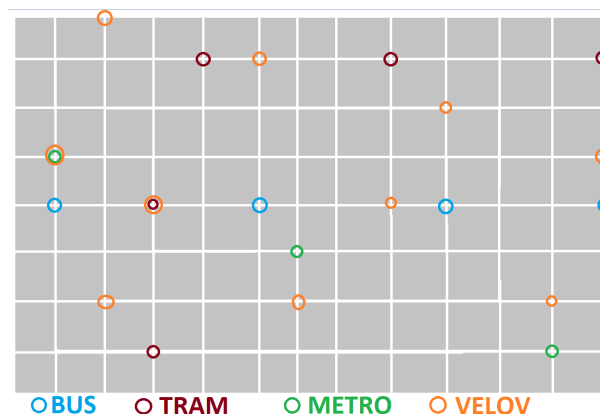
- La carte va être plus petites, eh oui quelle surprise, nous n'allons pas faire notre projet sur toute la carte du monde, mais sur celle de la taille d'une petite ville, ou un centre-ville

- La carte ne sera pas une représentation exacte d'une ville, mais une simplification de style Brooklyn, d'où notre nom de groupe. C'est-à-dire que tous les pâtés de maison seront des carrés, et les intersections à équidistance entre elles. Ainsi la "carte" sera en fait une grille, ce qui facilite beaucoup d'aspect aussi bien pour l'affichage que le calcul des chemins.



La distance Brooklyn est donc bien trop simple à calculer. Le temps pour se rendre d'un point A à un point B est un simple calcul mathématique. Donc nous allons rajouter quelques éléments pour rendre le projet plus intéressant, déjà des transports en commun : ceux que nous trouvons dans notre chère ville de Lyon. A savoir :

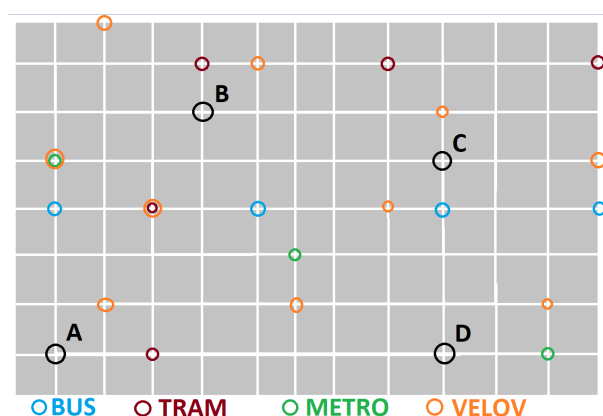
- Des trams : vitesse moyenne, mais stations placées à des endroits assez stratégiques.
- Des Métros : vitesse très rapide mais peu de stations.
- Des velov : vitesse moyenne et bornes de velov presque partout, tous les 3 – 4 pâtés de maisons.
- Des Bus : vitesse rapide, et dessert les endroits peu desservis par les trams et métros.



Cette complexité introduite va nous amener au concept des graphes pondérés. C'est-à-dire que nous allons avoir des nœuds pour chaque intersection de rues

sur la carte et des liaisons qui relient ces nœuds. Ces liaisons auront un poids selon le mode de transport que notre application va emprunter pour se rendre de A à B. Ce poids représentera le temps qu'il faut pour aller d'un nœud à un autre. Ainsi nos algorithmes devront prendre en compte ce poids pour déterminer les chemins les plus rapides.

Nous sommes que 3 dans ce groupe, donc nous n'avons pas un projet avec la même quantité de travail qu'un groupe de 4. Donc la description ci-dessus serait l'objectif principal de notre projet. Cependant pour ajouter de la complexité à notre projet, et pour avoir une partie d'algorithmie plus dominante dans notre projet, nous rajouterions l'option de choisir plusieurs points sur la carte, et que notre application ne se charge pas juste de trouver le plus court chemin en termes de temps entre A et B, mais le temps minimal pour parcourir une plus grande quantité de points. Cela rajouterait de la complexité algorithmique.

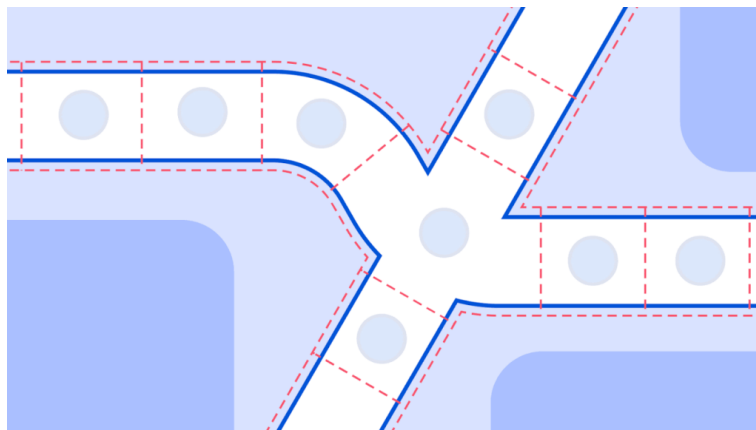


Si nous avons de l'avance sur le projet, nous pourrions activer des options où nous voulons avoir un trajet non payant, c'est à dire désactiver le metro, tram et bus. Ou même incorporer un temps d'attente pour le métro, tram et bus, ce qui ressemblerait plus à la vraie vie... Ce sont juste quelques idées pour rajouter un peu de complexité à notre projet vers la fin.

3 Etat de l'art

Nous essayons de simplifier Google maps. Bien que plus complexe que notre projet, Google maps utilise aussi des graphes comme structure de base. Nous voulons mettre un noeud de notre graph à toutes les intersections de routes, alors que google découpe les routes en plusieurs petits segments et donc place bien plus de noeuds que nous. Google fait cela pour mieux gérer les données reçues

par les conducteurs de voitures et donc mieux afficher des zones où il y aurait potentiellement des embouteillages. Si la vitesse de déplacement de voitures sur une route est bien inférieure à la vitesse moyenne prédite (par machine learning) alors il affiche la route rouge pour signaler un embouteillage.



Ces bouts de route que Google a mis en rouge sont des liaisons entre noeuds plus lourdes (avec un poids plus fort). En matière de temps du coup, aller d'un noeud à l'autre met plus de temps. Nous allons aussi utiliser ces poids sur les liaisons (arcs) qui représentent des transports en commun. Pour le métro par exemple, comme il avance très vite, ces arcs auront des petits poids et l'arc qui représente marcher d'une intersection à une autre aura un plus gros poids car beaucoup plus lent.

Assez sur la structure, nous allons parler du parcours. Google cherche le plus court chemin entre deux noeuds grâce à l'algorithme de Dijkstra. Celui-ci est assez fort surtout pour les graphes pondérés. Ce nom était assez intimidant mais il paraît qu'il est assez simple à comprendre. Donc nous envisageons d'utiliser le même.

4 Détails du projet

4.1 Etapes du projet

Comme dans tout projet nous avons decoupe ce que nous avons prévu de faire en plusieurs etapes :

- Faire les classes et fichier
- Créer un graph de base simplist
- Créer le premier algorithme ne prenant en compte que la marche
- Implémenter le bus comme premier transport

- Améliorer l'algorithme pour qu'il prenne en compte les bus
- Implémenter le vélo, le tram et le metro
- Améliorer une nouvelle fois l'algorithme pour choisir le plus rapide entre chaque
- Avoir un rendu graphique pour pouvoir visualiser le chemin rendu
- Créer une interface pour pouvoir changer les points de departs et d'arriver
- Agrandir la carte et rajouter des obstacles
- Laisser la possibilité de choisir entre plusieurs cartes
- Pouvoir rajouter des étapes intermédiaires
- Créer les cartes aléatoirement
- Faire le site web

4.2 Distribution des tâches

Tâches	Guillaume	Titouan	Hamza
Classes et fichiers	x		
Graph carte de base		x	
Premier algorithme	x	x	
Première méthode de transport	x		x
Améliorer l'algorithme en conséquence	x	x	
Implementer autres modes de transports	x		x
Sélection de point de départ et arrivée			x
Affichage graphique			x
Interface graphique			x
Modifier la carte		x	x
Pouvoir choisir entre plusieurs cartes		x	x
Site web	x	x	x
Rajouter des étapes de déplacement	x	x	x
Cartes aleatoires	x	x	x

4.3 Planning

4.3.1 Première soutenance, 14 fevrier

Le but pour la première soutenance sera d'avoir une base technique viable pour pouvoir construire par-dessus. Nous voudrions avoir implementer les structures de graph évidemment, ainsi qu'une première version de la carte. Cette première version ne sera qu'une grille carrée d'une petite taille comme 10 par 10 pour commencer a expérimenté nos algorithmes. Niveau algorithmie justement, il nous faudra minimum pouvoir donner un chemin comprenant un type de transport. Afin de pouvoir visualiser convenablement le projet, l'interface graphique devra être au moins au stade de prototype

4.3.2 Deuxième soutenance, 4 avril

Pour celle-ci l'avancement dépendra des difficultés rencontrées mais à priori nous aimerions avoir fait la majorité de la partie technique que nous nous sommes fixée au début. Même si le fonctionnement ne sera sûrement pas encore parfait ni optimisé nous aimerions que tous les transports en commun fonctionnent. Pour le reste la carte sûrement encore simpliste mais l'interface graphique sera presque finalisée.

4.3.3 Soutenance Finale, 16 mai

Evidemment tous les aspects de base du projet seront implémentés pour cette date, cela inclus : les transports en commun, plusieurs cartes et l'interface graphique. Si le temps le permet nous pourrons rajouter les bonus envisagés précédemment, c'est-à-dire avoir plusieurs points de destination et la génération de cartes procedural

4.4 Ressources

4.4.1 GTK

Une interface graphique permettra de rassembler tous les algorithmes de notre Google Maps made in Epita pour tenter d'offrir une expérience simple et élégante à l'utilisateur. Cette dernière sera réalisée par l'intermédiaire de la librairie GTK qui a été vue au dernier semestre ce qui simplifie la tâche ne serait-ce qu'au début.

4.4.2 GitHub

Nous préférons utiliser github car sur le site de github nous avons beaucoup plus d'informations sur les versions et auteurs que sur un git classique. De plus c'est quand même sympa avoir une fenêtre graphique avec plein de couleurs et des petits boutons interactifs. C'est zoli.

4.4.3 Language de programmation :

On a pas le choix : le C. Nous comptons coder les class graph et les algorithmes de parcours de Dijkstra.

Nous utiliserons le compilateur gcc.

5 Conclusion

Ce projet sera pour nous une grande étape en tant que développeur, il nous permettra de nous approcher de problème concret de la vraie vie de la même façon que le projet OCR. De plus le travail de groupe est une compétence essentielle dans la vie professionnelle, faire ce projet nous permettra donc de nous améliorer dans ce domaine. Nous avons hâte de nous investir dans ce projet afin de recréer une version un tout petit peu simplifiée d'un service que nous connaissons et dépendons tous.