



Projet PPII2 - Rapport final
Groupe 18



Rapport du Projet Pluridisciplinaire d'Informatique Intégrative

23/05/2023

Rédigé par l'ensemble des membres du groupe

Membres de l'équipe :

Nom et mail	Rôles
THUILLIER Killian killian.thuillier@telecomnancy.eu	Chef de projet
SOLDATOV Maxime maxime.soldatov@telecomnancy.eu	Responsable algorithmie
TIGER Maelan maelan.tiger@telcomnancy.eu	Gestion parser
VESSE Léo leo.vesse@telecomnancy.eu	Structure

Table des matières

1	Introduction	5
1.1	Contexte	5
1.2	Objectifs de l'application	5
1.3	Avantages attendus	5
2	Conception et implémentation	6
2.1	Structure de données	6
2.1.1	Structure de Bornes	6
2.1.2	Structure de véhicules	7
2.2	Parser des documents	7
2.2.1	Récupération et traitement des données	8
2.2.2	Génération d'une liste des bornes	8
2.2.3	Génération de la matrice d'adjacence	8
2.3	Algorithmie	9
2.3.1	L'idée première : Dijkstra	9
2.3.2	La solution retenue au final	9
3	Tests et complexités	10
3.1	Tests	10
3.2	Complexité	10
3.2.1	Fonction 'generate_graph_fromCSV'	11
3.2.2	Fonction 'pathFinding'	11
4	Gestion de projet	11
4.1	Composition et organisation du groupe	11
4.2	Réunions	12
4.2.1	Régularité des réunions	12
4.2.2	Investissement durant les réunions	12
4.2.3	Mise en place des plannings	12
4.2.4	Problèmes soulevés	12
4.3	Répartition des tâches	13
4.3.1	WBS	13
4.3.2	Diagramme de Gantt prévisionnel	14
4.3.3	Diagramme de Gantt réel	14
4.3.4	Analyse	15
4.4	Retour sur la matrice SWOT	15
5	Bilan du projet	15
5.1	Améliorations	15
5.1.1	Fonctionnalités abandonnées / en cours de réalisations	15
5.1.2	Fonctionnalités à améliorer	16
5.2	Conclusion générale	16
5.3	Conclusions personnelles	16
5.3.1	THUILLIER Killian	16
5.3.2	VESSE Léo	16
5.3.3	TIGER Maelan	17
5.3.4	SOLDATOV Maxime	17

6	Annexes	17
6.1	Comptes rendus	17

1 Introduction

1.1 Contexte

Nous avons le plaisir de vous présenter le rapport de rendu final pour notre projet portant sur le développement d'une application en langage C dédiée à la localisation des points de recharge pour voitures électriques effectuant un trajet d'un point A à B en France. Ce rapport vise à présenter en détail les objectifs, les étapes de développement, les fonctionnalités implémentées, les résultats obtenus ainsi que les perspectives futures de cette application.

Le développement rapide de la mobilité électrique a suscité un besoin croissant pour des infrastructures de recharge fiables et accessibles. Dans ce contexte, notre équipe s'est attelée à la conception et à la réalisation d'une application intuitive et efficace, capable d'aider les utilisateurs à localiser facilement les stations de recharge les plus proches, de planifier leurs trajets en fonction de ces informations.

1.2 Objectifs de l'application

Dans ce contexte, notre projet vise à contribuer à l'amélioration de l'expérience des utilisateurs de véhicules électriques en France en développant une application en langage C dédiée à la localisation des points de recharge. L'objectif principal de cette application est de fournir aux utilisateurs une solution pratique et conviviale pour trouver rapidement les stations de recharge disponibles à proximité de leur position actuelle ou de leur destination.

1.3 Avantages attendus

En développant cette application, nous cherchons à offrir plusieurs avantages aux utilisateurs de véhicules électriques en France :

- Gain de temps : L'application permettra aux utilisateurs de visualiser les bornes auxquelles s'arrêter sur leur trajet.
- Planification efficace des trajets : Grâce aux informations détaillées fournies par l'application, les utilisateurs pourront planifier leurs trajets en tenant compte de l'autonomie du modèle de leur voiture ou d'éventuels paramètres tels qu'un minimum de batterie à garder, un temps de recharge maximal ou bien un temps d'attente à une station avant de charger maximal.
- Informations pertinentes : L'application fournira des informations utiles telles que le temps de trajet ou le temps passé à chaque borne.
- Contribution à la transition énergétique : En facilitant l'utilisation des véhicules électriques, notre application contribuera à la réduction des émissions de carbone et à la promotion d'un mode de transport plus respectueux de l'environnement.

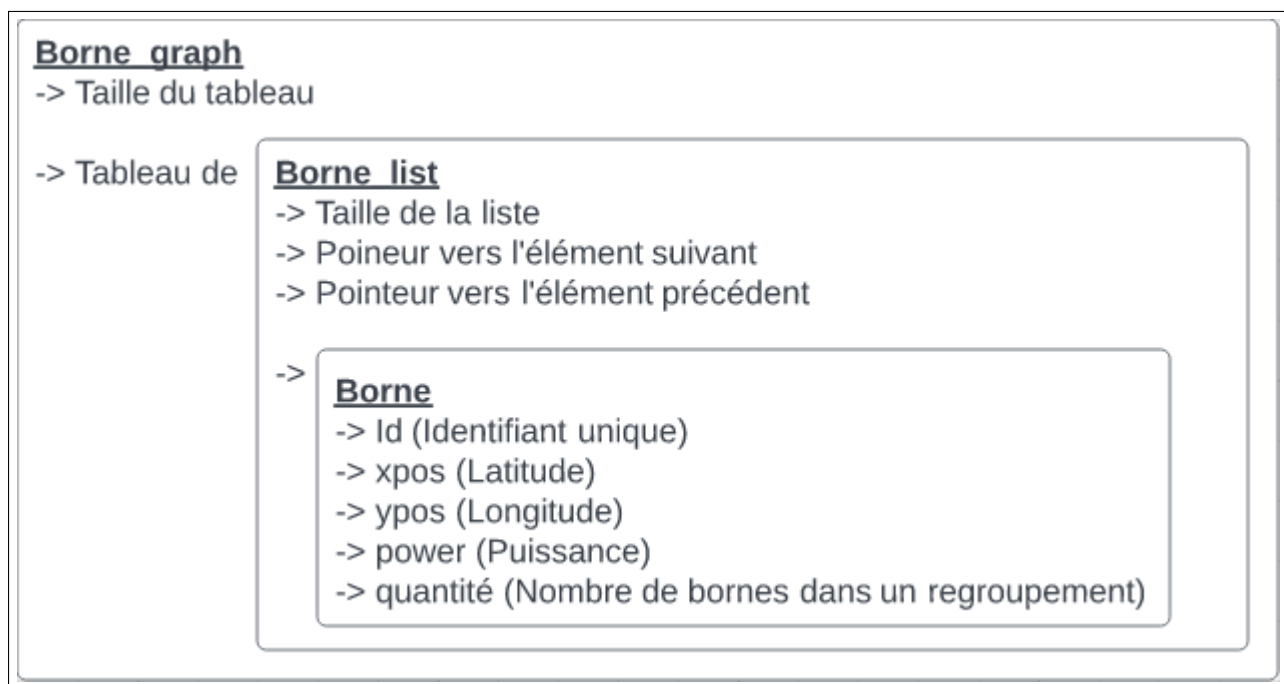
2 Conception et implémentation

2.1 Structure de données

Pour réaliser l'implémentation de l'application, nous avons du créer une structure de données adaptée à l'algorithme que nous souhaitons implémenter et pouvant contenir toutes les données utiles au projet. Nous avons conçu plusieurs structures simples afin de pouvoir les manipuler et les modifier facilement au besoin. L'utilisation de ces structures et particulièrement des méthodes associées a été un point crucial dans une progression rapide du projet qu'elle a permis d'identifier plus rapidement l'origine des problèmes survenu lors de la programmation de l'application.

2.1.1 Structure de Bornes

L'implémentation des structures liées aux bornes s'est faite en 3 parties.



- Une première structure de bornes nous permettant de stocker les informations importantes liées aux zones de rechargements telles que
 - La position (longitude, latitude)
 - La puissance de rechargement d'une borne qui nous permet de déterminer le temps de recharge d'une voiture
 - Le nombre de points de recharges dans une zone qui nous aurait permis à terme de gérer plus facilement l'attente aux bornes.
- Une deuxième structure englobant la première. Celle-ci a pris la forme d'une liste doublement chaînée cyclique afin de réduire la complexité des algorithmes d'insertion et de suppression notamment. Elle contient un attribut de taille afin de permettre un condition d'arrêt des algorithmes plus claire. Cette structure est utile pour déterminer la liste des voisins d'une borne notamment.

- Enfin une troisième structure englobant les deux premières qui servira pour la conception de notre graphe. C'est un array (tableau) de taille fixe dont le i-ème élément contient les voisins de la i-ème borne. Une implémentation en liste d'adjacence classique.

Remarques sur l'implémentation de la structure de données:

- Des fonctions de 'print' sur la sortie standard ont été implémentées afin de faciliter le débogage.
- Des fonctions de créations et de destruction ont bien évidemment été implémentées afin d'éviter les "memory leaks"

2.1.2 Structure de véhicules

Vehicule list

-> Taille de la liste

-> Pointeur vers l'élément suivant

->

Vehicules

-> Id (Identifiant unique)

-> Nom (modèle de la voiture)

-> Autonomie (en km)

-> Capacité (Puissance totale de la batterie)

L'implémentation de la structure des véhicules, moins importante est plus simple.

- Une première structure de véhicules qui nous permet de stocker les informations importantes sur un modèle de voiture.
 - Le nom d modèle de la voiture
 - Son autonomie en km
 - Sa capacité, qui permet d'obtenir son temps de charge
- Une seconde structure de liste de véhicules, qui est une liste simplement chaînée. Les contraintes du projet ne nécessitaient pas une implémentation doublement chaînée.

2.2 Parser des documents

L'ensemble des données du projets nous furent transmises via des fichiers csv ou un fichier txt. Le but était dans un premier temps de récupérer et traiter ces données puis de les utiliser dans l'application C.

2.2.1 Récupération et traitement des données

Après avoir téléchargé les données depuis les sources fournies dans le sujet du projet, nous avons choisi d'utiliser python pour traiter les données et les mettre sous un format plus adéquat. En effet les différentes librairies (Pandas, numpy...) sont particulièrement adaptées à l'analyse de tableaux de données. Les opérations de manipulation des données, telles que le filtrage, le tri et l'agrégation, peuvent être réalisées en quelques lignes de code claires et compréhensibles en Python, grâce aux fonctionnalités intégrées de Pandas.

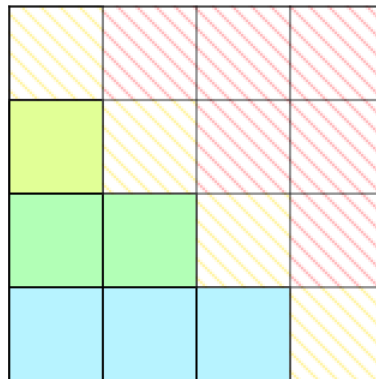
2.2.2 Génération d'une liste des bornes

Une analyse préliminaire des données concernant les bornes électriques a mis en lumière des erreurs d'unités, des bornes n'ayant pas de coordonnées géographiques cohérentes, ainsi qu'une grande quantité d'informations non pertinentes dans notre cas (identifiant de l'entreprise, type d'accès, modèle de prise compatible...). En complément, bien que le nombre de borne par zone de rechargement soit précisée, chaque borne individuelle avait une ligne.

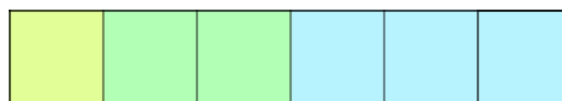
Nous avons donc décidé de nous limiter aux informations nécessaires à l'implémentation de notre algorithme, cela nous a aussi amené à regrouper ensemble toutes les bornes proches les unes des autres. Via la librairie Pandas, l'importation du csv est très rapide, et la création d'un fichier csv contenant les données importantes (position en x, position en y, la puissance nominale d'un ensemble de bornes ainsi que le nombre de points de chargement dans le groupement de bornes) est généré rapidement grâce à la fonction 'to_csv'.

2.2.3 Génération de la matrice d'adjacence

La génération de la matrice d'adjacence était assez simple à implémenter, l'utilisation de Numpy a permis assez facilement de réaliser une multiplication vectorielle des données des bornes pour obtenir la distance entre deux bornes. Le résultat était un fichier de 6 gigaoctets, assez complexe à utiliser tel quel. Nous avons donc procédé à une première optimisation, en effet les chemins étaient à double-sens, ce qui signifie que la matrice était symétrique et redondante.



Matrice d'adjacence non optimisée



Matrice d'adjacence optimisée

2.3 Algorithmie

Au vu de ce que devait réaliser l'application, la partie algorithmie allait être une partie majeure. C'est pourquoi dès la première réunion, nous avons instantanément commencé à discuter sur la façon d'implémenter cela.

2.3.1 L'idée première : Dijkstra

Lors de notre toute première réunion, nous avons réfléchis tous ensemble à comment résoudre efficacement le problème posé. Nous est alors venue une première idée naturelle : l'algorithme de Dijkstra.

L'idée nous est venue pour les raisons suivantes :

- Cet algorithme est très simple à implémenter. Après avoir générer les variables et données nécessaires à l'algo, il suffisait de lancer l'algo.
- Dijkstra est un algorithme assez performant en temps. Nous nous attendions alors à avoir assez rapidement le chemin dans le graphe généré assez rapidement

2.3.2 La solution retenue au final

Nous avons finalement opté pour un programme certes un peu similaire, mais opérant différemment. Après avoir généré le graphe en respectant la structure que nous avons implémenté, c'est à lire un 'Array' contenant une structure 'borne_list', nous parcourons les voisins du point de départ, en omettant les points déjà dans le chemin, et vérifions que la fonction définie comme suit :

```
bool test_condition(borne *current_borne, borne *current_neighbour_i,
                   borne *destination, borne *nearest_neighbour, int charge,
                   int battery_minimum,
                   int max_time_waiting, int actual_time,
                   vehicule *v, int max_time_charging)
```

Cette fonction a pour but de vérifier si le voisin 'current_neighbour_i' est meilleur que l'actuel meilleur 'nearest_neighbour'. Elle effectue les tests suivants :

- Vérifie si la voiture a assez de charge pour arriver au voisin testé avec plus de batterie que la limite fixée par l'utilisateur
- Vérifie si l'attente avant le chargement n'est pas plus long que la limite fixée par l'utilisateur
- Vérifie si le temps de trajet de l'origine au voisin testé puis de ce voisin à la destination finale (à vol d'oiseau, sans chargement) (en comptant le temps de chargement à la borne) est meilleur que celui pour aller à l'actuel meilleur

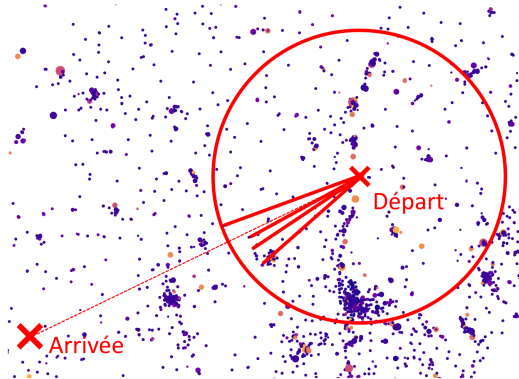
Ne reste alors plus qu'à faire tourner ce test sur toutes les bornes voisines de l'origine puis ajouter le meilleur voisin trouvé au chemin et changer le prochain point d'origine en arrêtant la boucle lorsque la destination est dans le chemin.

C'est là l'essence de la fonction pathFinding définie comme suit :

```
ornes_list *pathFinding(bornes_graph *bg, borne *origin, borne *destination,
                        int battery_minimum, int max_time_charging,
                        int max_time_waiting,
                        int actual_time, vehicule *v)
```

Nous pouvons donc schématiser l'implémentation choisie comme cela :

1. On génère le graphe qui contient les voisins de chaque borne. On trace donc d'un cercle de rayon `autonomie_max` et on stocke toutes les bornes dans le cercle dans la liste à l'indice 'i' du 'borne_graph'
2. On utilise le test expliqué ci-dessus sur les voisins (les bornes dans le cercle)
3. Le point final est le point duquel repartir pour l'étape suivant (boucler sur 2)
4. Si la destination est dans le cercle, on a fini notre chemin.



3 Tests et complexités

3.1 Tests

Dans ce projet, nous avons effectué deux types de tests distincts pour évaluer les différentes implémentations de fonctions.

- D'une part, nous avons testé les fonctions qui gèrent le graphe, notamment celles qui ajoutent, suppriment ou manipulent les bornes. Ces tests nous ont permis de vérifier le bon fonctionnement de ces fonctions et de nous assurer que le graphe était correctement construit et modifié.
- D'un autre coté, nous avons vérifié que les fonctions de parsing étaient capables de lire et de convertir les données CSV correctement. Ces tests ont été essentiels pour s'assurer que les données étaient correctement interprétées et utilisées dans la création du graphe.
- Nous n'avons pas réalisé de tests spécifiques pour l'algorithme de recherche de chemin que nous avons mis en place. Il aurait été bénéfique de tester cet algorithme pour confirmer son bon fonctionnement et vérifier s'il fournissait effectivement le meilleur chemin possible.

3.2 Complexité

Cette partie est réservée au calcul de complexité de l'ensemble du projet. Nous nous attarderons uniquement sur deux parties algorithmique dont la complexité ne saute pas aux yeux : la fonction 'generate_graph_fromCSV' et la fonction 'pathFinding'. On notera ici s le nombre de sommets et e celui d'arcs.

3.2.1 Fonction 'generate_graph_fromCSV'

Voyons déjà quelles fonctions sont appelées par 'generate_graph_fromCSV' et calculons leur complexités respectives:

- 'generateListFromCSV' : En posant n le nombre d'éléments données, (ici c'est le nombre d'éléments stockés dans la matrice d'adjacences soit $n=e$ (nombre d'arcs)). On trouve que 'generateListFromCSV' a une complexité $C_1 = O(e)$
- 'create_bornes_graph' : Complexité en $C_2 = O(s)$ puisque on crée s listes contenant un seul élément 'null' chacune.
- 'getBornesNumberFromIndex' : Complexité en $C_3 = \sqrt{e}$ puisque l'argument est de longueur au maximum e
- Les autres fonctions appelées seront de complexité linéaire.

De plus, 'generate_graph_fromCSV' fait appel à une boucle 'while' de longueur e et fait appel à l'intérieur à e fois le fonction 'getBornesNumberFromIndex'. Cette boucle est donc de complexité $C_4 = O(e^2\sqrt{e})$ ce qui donne une complexité $C = O(e^{3/2})$.

3.2.2 Fonction 'pathFinding'

Voyons déjà quelles fonctions sont appelées par 'pathFinding' et calculons leur complexités respectives:

- test_condition: $C_5 = O(1)$

L'appel à test_condition s'effectue dans une boucle for d'au pire e itérations elle même dans un 'while', 'while' d'au pire s itérations. Ainsi, la complexité au pire est de $C = e * s$. En réalité, la boucle 'while' s'exécute peut car au vu de la distance parcourue à chaque itération, on arrive très vite à destination.

4 Gestion de projet

4.1 Composition et organisation du groupe

Nom et mail	Rôles
THUILLIER Killian killian.thuillier@telecomnancy.eu	Chef de projet
SOLDATOV Maxime maxime.soldatov@telecomnancy.eu	Responsable algorithmie
TIGER Maelan maelan.tiger@telcomnancy.eu	Gestion parser
VESSE Léo leo.vesse@telecomnancy.eu	Structure

4.2 Réunions

4.2.1 Régularité des réunions

Au cours de notre projet, nous avons initialement opté pour des réunions à une fréquence relativement élevées, environ une par semaine. Au début, cela semblait suffisant pour maintenir une communication adéquate entre les membres de l'équipe.

4.2.2 Investissement durant les réunions

Nous avons constaté que tous les membres de l'équipe étaient fortement investis et participaient activement en proposant des idées et en donnant leur point de vue. Cela a permis d'enrichir les discussions et de générer des solutions novatrices. Cela a été particulièrement important pour la compréhension et l'implémentation de nos structures de données.

4.2.3 Mise en place des plannings

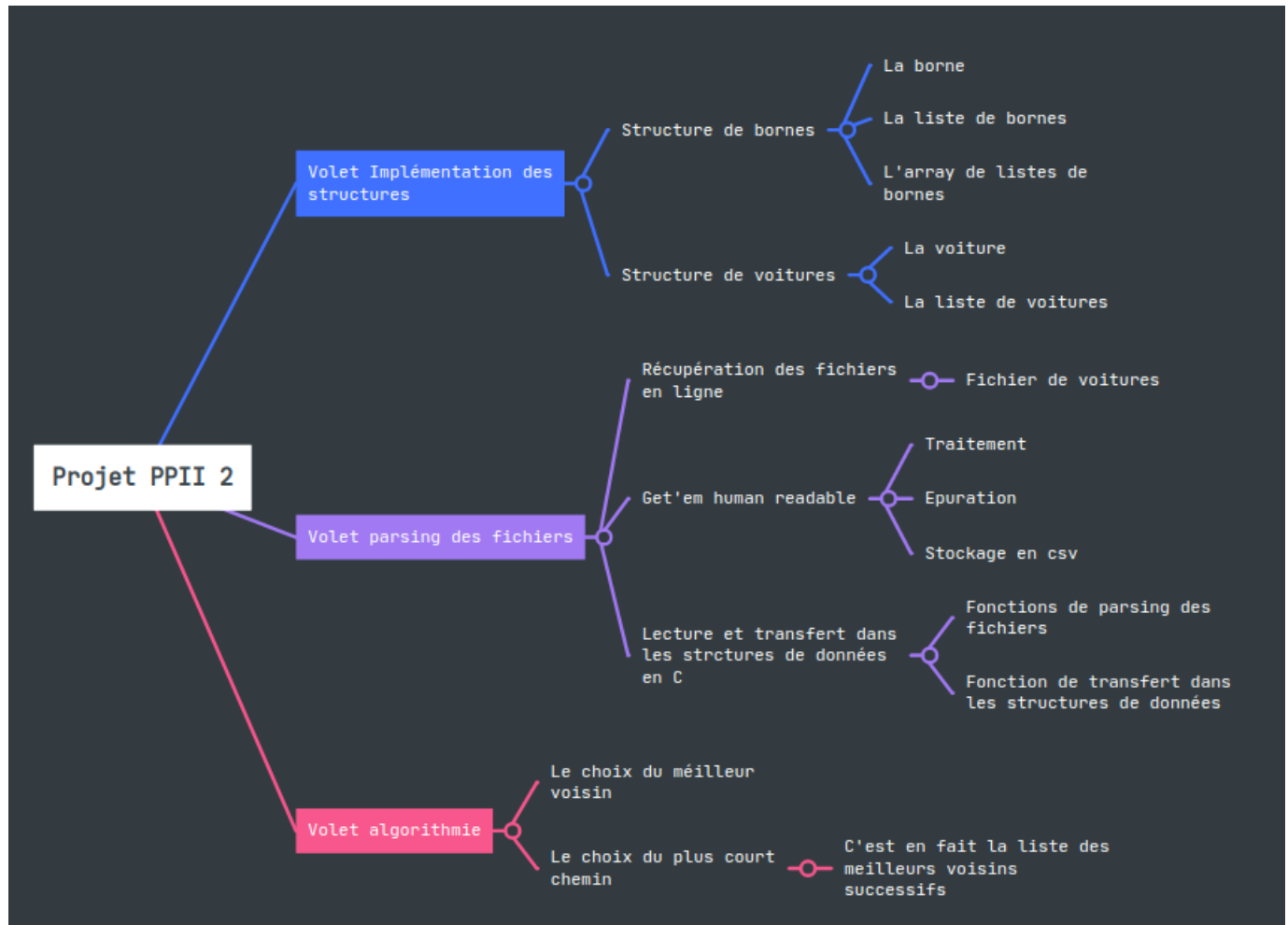
- L'idée d'utiliser des sites web et des applications dédiées tels que Trello et Notion pour faciliter la gestion du projet semblait prometteuse, mais cela s'est avéré un peu intimidant.
- D'autre part, l'utilisation du calendrier Google pour fixer les échéances et les horaires s'est révélée être une très bonne idée. D'autant plus qu'il permettait de prendre en compte les autres activités et obligations de chaque membre du groupe.

4.2.4 Problèmes soulevés

Malheureusement, nous avons eu des difficultés croissantes à s'organiser pour définir des dates de réunions. Les retards dans la communication ont entraîné des problèmes de coordination et une certaine confusion au sein de l'équipe. Nous aurions dû organiser des réunions plus fréquentes pour discuter des progrès, des obstacles rencontrés et des ajustements nécessaires.

4.3 Répartition des tâches

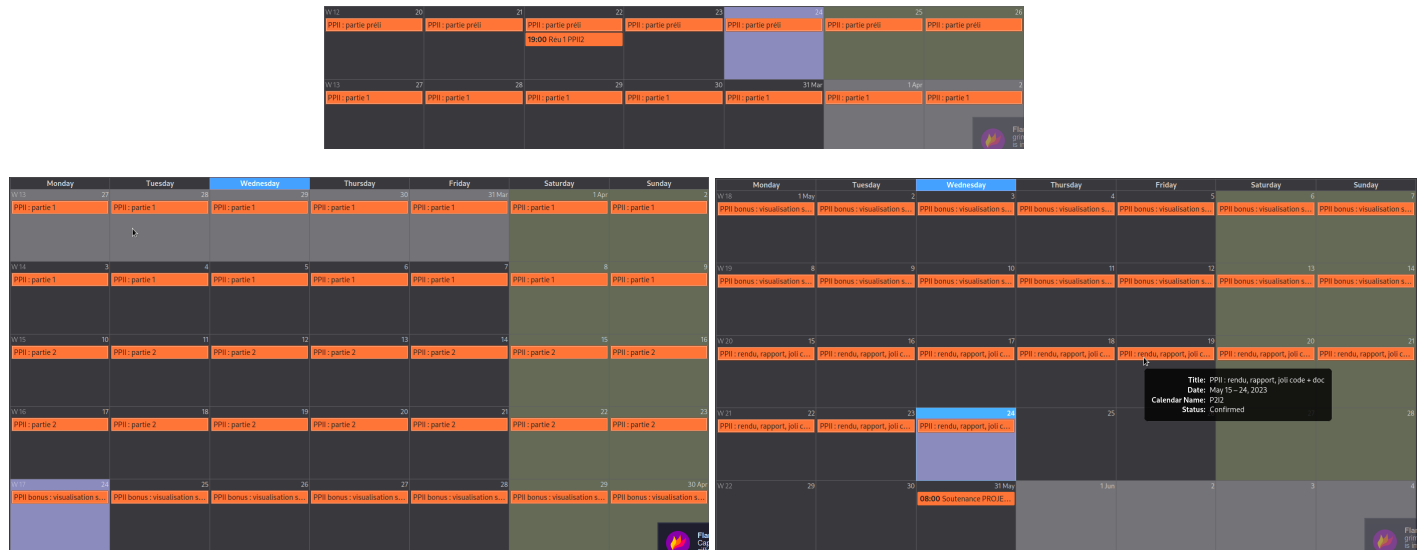
4.3.1 WBS



Nous avons divisé la charge de travail en 3 grand volets.

- En premier, l'implémentation des structures de graphes et de voitures. Celles-ci permettent à la totalité du groupe d'utiliser les mêmes structures de données et de manipuler les mêmes objets théoriques. Ça a permis de faciliter aussi la réécriture du code, la manipulation des objets et les tests de ces fonctions centrales dans la partie algorithmie notamment
- Ensuite un deuxième volet qui a posé les fondements du projet, le parsing des fichiers. Il fallait aller les récupérer sur les ressources données dans le sujet, les lire et les mettre en forme pour l'application. Une deuxième sous partie consistait à lire les fichiers produits par le python afin de les intégrer au C. Il fallait donc réaliser un parser de fichier en C.
- Enfin la dernière partie consistait à relier les deux premières et construire un algorithme efficace pour la recherche de chemin.

4.3.2 Diagramme de Gantt prévisionnel



La conception de ce projet a été répartie en 5 grands axes lors de notre première réunion.

1. 20-26 Mars : Une partie préliminaire qui consistait à récupérer les informations nécessaires au projet, la liste des véhicules ainsi que les bornes. Cette partie comprenait aussi la mise en formes de ces documents afin de faciliter la lecture et le parsing.
2. 27-9 Avril : La gestion de la partie doit se faire durant cette période. L'objectif était de réaliser l'implémentations des structures, du parser et terminer l'algorithmie du programme. Nous avons déjà réalisé que la deuxième partie du projet était bien plus simple si la première partie était suffisamment générale.
3. 10-23 Avril : Cette date de fin de projet ne tenait pas compte des retards possibles du groupe mais prévoyais une marge suffisante pour s'en permettre.
4. 24-14 Mai : 3 membres sur 4 participant à la création du nouveau PayMyBar (OpenBar), nous souhaitons nous attarder la création d'une visualisation sur web avec un framework svelte.
5. 15-24 Mai : Cette période était réservée à la rédaction du compte rendu. Elle permettait aussi de se permettre du retard et de prendre de l'avance sur le rendu compte tenu des partiels approchants.

4.3.3 Diagramme de Gantt réel

En réalité comme dans tout projet, le Gantt a été largement décalé vers la date e rendu de projet et s'est effectué en très grande partie dans la dernière semaine. La partie 1, à l'heure où ont été écrites ces lignes n'a toujours pas été terminée. De multiples erreur d'exécutions et d'algorithmie ont été commises. Le manque de communication dans le groupe a été un problème de taille. Les codes étant faiblement commentés et répondant à une logique très spécifique pour répondre au problème posé, il était compliqué de maîtriser la totalité du sujet et du code produit. Ainsi, en l'absence de

deux membre durant la conception de la plus grande partie du projet, certaines questions se sont posées, il été difficile de comprendre et notre groupe a perdu beaucoup de temps à comprendre ce que d'autres avaient fait.

4.3.4 Analyse

4.4 Retour sur la matrice SWOT

	Points positifs	Points négatifs
Source interne	S1 - De nombreuses licences de sites et logiciels accessibles S2 - Compétences acquises en algorithmie et structure de données au cours de l'année	W1 - Difficulté de réalisation de certaines fonctionnalités
Source externe	O1 - Travailler en groupe O2 - Développer nos compétences informatiques	T1 - Ne pas suivre le Gantt T2 - Perte de motivation

S1- Nous avons utilisé la licence Overleaf ainsi que le logiciel VSCode pleinement. La documentation sur internet nous a fortement aidé.

S2- Ces compétences nous ont permis de constituer une structure de données très rigide et un algorithme fonctionnel malgré quelques bugs qui subsistent

W1- L'implémentation pourrait s'avérer compliquée dans ce langage rigoureux qu'est le C

O1- La communication a été catastrophique. Il a été impossible de joindre des membres du groupe.

O2- Évolution exponentielle, nous sommes partis de presque rien pour arriver à de nombreux automatismes aussi bien en base de données qu'en HTML et Flask.

T1- La Gantt n'a pas été suivi à cause du manque de motivation et l'impossibilité de joindre certains membres

T2- La période de rush se superposant avec une période de révisions de partiels, la motivation a été plus faible sur intervalle de temps central dans le projet

5 Bilan du projet

5.1 Améliorations

Vous trouverez ici l'ensemble des amélioration qui seront ou pourraient être implémentés sur le projet ainsi qu'un descriptif des fonctionnalités abandonnées

5.1.1 Fonctionnalités abandonnées / en cours de réalisations

- L'idée d'un site affichant le résultat est devenue irréalisable par manque de temps.

- L'implémentation d'un Dijkstra n'est pas réalisable en raison de la forte connexité du graphe et du non sens de la recherche de plus court chemin en distance au but des besoins de l'utilisateur.
- L'idée de segmenter la carte (en fonction de la direction vers laquelle on part pour réduire le temps de calcul) a été abandonnée car la rapidité d'exécution de la recherche de chemin était assez rapide.
- L'implémentation de la simulation sur N voitures a déjà été réfléchie et pré-implémentée. Elle pourrait être finie d'ici à la présentation devant le jury.

5.1.2 Fonctionnalités à améliorer

Vous trouverez ici l'ensemble des améliorations qui seront ou pourraient être implémentées sur le projet ainsi qu'un descriptif des fonctionnalités à améliorer

- La correction de quelques bugs devrait être un point central dans l'avancée du projet. Pour le moment la plupart des fonctions sont indépendamment correctes mais l'association entre elles ne l'est pas. On peut penser à une mauvaise compréhension du travail de l'autre, de l'utilisation des structures ou des bases du projet peu solides.
- L'implémentation en matrice d'adjacence pourrait apporter en efficacité dans l'exécution de l'algorithme. Cette forme permettrait de mettre de côté certaines fonctions annexes dont le temps d'exécution est long. Il faudrait l'implémenter et comparer avec l'implémentation pour pouvoir observer les différences d'efficacité.
- Enfin, une interface utilisateur et l'ajout de quelques facilités de lecture et de compréhension des informations renvoyées par l'application pourraient être implémentées

5.2 Conclusion générale

5.3 Conclusions personnelles

5.3.1 THUILLIER Killian

Ce projet a pour moi eu été très lourd à porter. En effet, certains(s) membres de mon groupe ayant manqué de motivation et n'ayant que peu ou pas participé ont miné à moi aussi mon moral. Ce projet, qui selon moi devait être bouclé très vite car les membres de l'équipe sont en mon sens très doués et efficaces, a été une immense déception et une source de frustration et de colère assez grande pour moi. Cependant, je me dois de reconnaître ma part de responsabilités : en ma qualité temporaire de chef du projet, mon devoir aurait dû être de relancer plus mes camarades et non attendre que ceux-ci reviennent vers le reste du groupe.

De plus, mon groupe et moi avons convenu d'un planning "à minima" qui permettait d'avoir le champ libre pour les périodes de partiels, mais le retard accumulé a été si gros qu'il m'a fait prendre beaucoup de retard dans leurs révisions.

Toutefois, j'estime que ce projet m'a été bénéfique dans deux domaines : ma rapidité à coder et ce en symbiose avec UN camarade, et il m'a permis de beaucoup discuter avec Léo, un ami que j'estimais beaucoup déjà avant ce projet (♥).

5.3.2 VESSE Léo

Ce projet n'était pas compliqué. Pourtant le temps qu'il a pris à être réalisé a été extrêmement frustrant. Malgré ma motivation à travailler, et à vouloir assister, cela n'a pas été suffisant à cause des multiples absences et des incompréhensions au sein du groupe. Le projet a été complètement

bouclé en 3 jours et demi (week-end de la Toussaint) avec près de 50h de travail par personne (ce qui représente environ 100h de travail). La fragilité psychologique dont j'avais pu faire part en début 2023 ne s'est pas améliorée lors de ce projet. J'aurais bien aimé travailler mes partiels pendant ce week-end, j'avais plus ou moins oublié le projet et de réapprendre son existence le mardi, j'espérais avoir le soutien de tous les membres de projet pour le terminer rapidement quitte à ce que ça ne soit par parfait afin de retourner à mes révisions. Malheureusement durant ces 4 jours je n'en ai trouvé qu'un seul, Killian (♡) qui a certainement été plus actif que moi, et qui a sauvé ce projet de bout en bout en me tenant par la main. Ça a été très éprouvant de travailler autant, et en étant dans l'impossibilité de joindre deux membres. Ce que m'a inspiré le déroulé de ce projet, une pure haine, une grande frustration malgré l'intérêt que je peux porter au C de manière générale. Pour finir, on verra bien comment les partiels se déroulent, j'avais prévu un jour par matière pour pouvoir me reposer le soir au final, cela fait plusieurs jours que je dors très peu, je n'ai pas eu l'occasion de faire face à mes problèmes personnels qui me sont arrivés pendant la semaine. J'ai dû mettre de côté tout ça pour terminer ce projet, c'est déjà peu recommandable de manière générale, alors dans mon état encore moins.

Je vous laisse je vais fêter la fin de cet enfer.

5.3.3 TIGER Maelan

J'avoue avoir ressenti une certaine appréhension à me lancer dans un nouveau projet de groupe, car lors de mon précédent projet, j'ai rencontré des problèmes personnels qui m'ont rendu incapable de travailler activement. Cependant, dès la première réunion de notre nouveau projet, j'ai été agréablement surpris par l'ambiance et l'engagement de l'équipe. Nous avons même commencé avant la plupart des autres groupes, ce qui était encourageant.

Cependant, un manque de communication est apparu sans que je puisse vraiment en expliquer la source. Malgré ces difficultés, j'ai continué à travailler sur le projet. Pendant les vacances, j'ai consacré beaucoup de temps (au moins une vingtaine d'heure) à effectuer des recherches approfondies pour apporter de nouvelles idées et contributions.

Durant la dernière semaine, j'ai dû me recentrer sur mes examens, ce qui a coïncidé avec une période intense où deux autres membres de l'équipe ont travaillé intensivement sur le projet pendant quatre jours consécutifs.

Au moins, j'ai pu identifier la racine principale de mon problème de communication, ce qui rend encore plus difficile le travail en groupe. Cela m'a permis de prendre conscience de l'importance de la communication et de l'effort nécessaire pour établir une communication claire et régulière au sein de l'équipe. J'ai l'intention de travailler sur ce point afin de mieux m'intégrer et de participer activement au succès de mes futurs projets.

5.3.4 SOLDATOV Maxime

(Léo) Je rédige sa partie même si c'est un ressenti qui doit être personnel. Cette partie est une retranscription partielle et interprétée du message qu'il nous a envoyé à tous le 24/05 à 17h39

Il nous a finalement fait part de son ressenti et de sa difficulté à communiquer dans le groupe. C'est dû à des problèmes personnels dont il ne nous a pas fait part e

6 Annexes

6.1 Comptes rendus

Projet PPII2 - Compte rendu n°01

Rédigé par Maelan TIGER

20/03/2023

Motif / type de réunion: Réunion préliminaire	Lieu: Salle de TN
Présent(s) (retard/excusés/non excusés): Tous	Date / heure de début / durée: 20/03/2023 1h de réunion

Suivi des objectifs précédents

Première réunion, pas de précédents

Ordre du jour

1. Etude des documents disponibles
2. Ebauche du planning

Informations échangées

Etudes des documents disponibles

- Première lectures des caractéristiques des voitures
- Analyse de la base de donnée fournie pour la gestion de l'emplacement des bornes
- Réflexions sur le système de coordonnées GPS utilisées

Ebauche du planning

- Partie préliminaire du sujet pour le 26 Mars
- Partie 1 pour le 9 avril
- Partie 2 pour le 23 avril
- Partie supplémentaire à terminer d'ici le 13 Mai

Actions à suivre / Todo list

Description	Responsable	Délai	Livrable	Validé par
Etude de cas				

Prochaine réunion : 27/03/2023

Documents additionnels:

(en l'absence de remarques, ce CR sera considéré comme approuvé le 23/03)

Check-list efficacité réunion

Points	Oui/Non
Pre-work fait	Null
Ordre du jour: - complet - motivant - envoyés à l'avance.	Oui Oui Non
Introduction: -tous les objectifs rappelés -gestion du temps annoncée	Oui Oui
Rebouclage avec CR précédent	Null
Ambiance sérieuse et constructive	Oui
Supports présentés de qualité	Null
Animateur efficace: -gestion du temps -sujets importants traités -pas de perte de temps -circulation de la parole	Oui Oui Oui Oui
Conclusion réalisée	Oui
Prochains objectifs définis et clairs	Oui
Date prochaine réunion fixée	Oui
Compte-rendu propre en 48h	Non