



Projet PPII2 - Rapport final

Groupe 18



Rapport du Projet Pluridisciplinaire d'Informatique Intégrative

12/06/2023

Rédigé et relu par l'ensemble des membres du groupe.

Membres de l'équipe :

Nom et mail	Rôles
THUILLIER Killian killian.thuillier@telecommancy.eu	Developper Debogger Lead réunions
TIGER Maelan maelan.tiger@telecommancy.eu	Developper Secrétaire Conception d'algorithmie
VESSE Léo leo.vesse@telecommancy.eu	Developper Debogger Responsable Organisationnel

Contents

1. Introduction	4
1.1. Contexte	4
1.2. Objectifs de l'application	4
1.3. Avantages attendus	4
2. Conception et implémentation	5
2.1. Formalisme et convention retenue pour le nom des fonctions et des variables .	5
2.2. Structure de données	5
2.2.1. Structure de Bornes	5
2.2.2. Structure de Véhicules	6
2.2.3. Structure d'Horaires	7
2.2.4. Structure pour le Parser	7
2.3. Parser des documents	8
2.3.1. Lecture et récupération des données sous forme de string	8
2.3.2. Importation vers les structures existantes	8
2.4. Algorithmie	9
2.4.1. L'idée première : Dijkstra	9
2.4.2. La solution retenue à la V1	9
2.4.3. La solution retenue à la V2	10
3. Tests et complexités	11
3.1. Tests	11
3.2. Complexité	12
3.2.1. Fonction 1 : betterWaitTime	12
3.2.2. Fonction 2 : pathFinding	12
4. Gestion de projet	13
4.1. Composition et organisation du groupe	13
4.2. Réunions	13
4.3. Répartition des tâches	14
4.3.1. WBS	14
4.3.2. Diagramme de Gantt	14
4.4. Retour sur la matrice SWOT	15
5. Bilan du projet	16
5.1. Améliorations	16
5.1.1. Fonctionnalités abandonnées	16
5.1.2. Fonctionnalités à améliorer	16
5.2. Conclusion générale	17
5.3. Conclusions personnelles	17
5.3.1. THUILLIER Killian	17
5.3.2. VESSE Léo	18
5.3.3. TIGER Maelan	18
6. Annexes	18
6.1. Comptes rendus	18

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. Formalisme et convention retenue pour le nom des fonctions et des variables

Dans l'optique de garder une cohérence dans l'ensemble des fonctions et des variables ainsi que pour trouver facilement les fonctions des structures que nous implémentons, nous avons choisis de suivre les règles suivantes quant aux noms donnés:

- Nom des fonctions :
 - Noms clairs et précis au format camelCase (exemple : `nomDeMaFonction`)
 - La longueur du nom d'une fonction n'a pas d'importance si elle n'est pas liée à une structure de donnée en particulier
 - Les fonctions liées à une structure de donnée respectent un formattage particulier : `nomDeLaStructure_actionARealiser`
- Nom des variables :
 - Noms clairs et précis au format camelCase (exemple : `nomDeMaVariable`)
 - Noms de variables en anglais au maximum
- Nom des structures :
 - Nom des structures clair et concis
 - Pour des structures faisant appel à d'autre sous-structures, celles-ci seront spécifiées comme suis : `sousStructure_typeDeSurStructure`. Exemple : `car_list` implémente une liste de sous structure `car`

2.2. Structure de données

2.2.1. Structure de Bornes

Relationships - Class Diagram

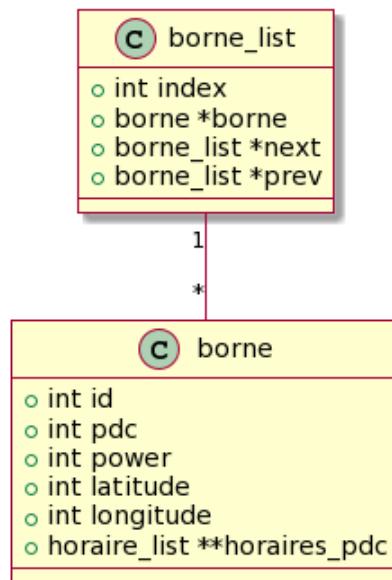


Figure 1: Diagramme UML structure `borne_list`

L'implémentation des structures liées aux bornes s'est faite en 2 parties:

- Une première structure de bornes appelée **borne** nous permettant de stocker les informations importantes liées aux zones de rechargements telles que :
 - L'id de la borne, noté **id**
 - La position (longitude, latitude) , notées **latitude** et **longitude**
 - La puissance de rechargement d'une borne qui nous permet de déterminer le temps de recharge d'une voiture, notée **power**
 - Le nombre de points de recharges dans une zone, noté **pdc**
 - Un tableau d'**Horaires_list** qui contient les horaires d'arrivée et de départ des autres voitures sur les prises de chargement de la borne, noté **horairesPdc**.
- Une deuxième structure englobant la première appelée **borne_list**. Celle-ci a pris la forme d'une liste doublement chaînée cyclique afin de réduire la complexité des algorithmes d'insertion et de taille. Elle contient un attribut d'indice **index** afin de permettre une condition d'arrêt des algorithmes plus claire ainsi qu'un prédecesseur **prev** et un successeur **next** et une borne **borne**. Cette structure est utile pour déterminer la liste des voisins d'une borne ou contenir un chemin.

2.2.2. Structure de Véhicules

Relationships - Class Diagram

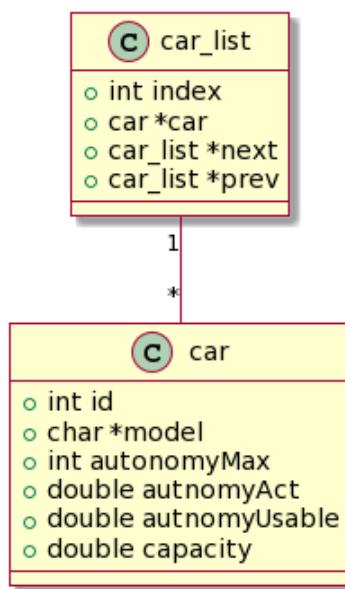


Figure 2: Diagramme UML structure **car_list**

L'implémentation des structures liées aux voitures s'est faite en 2 parties:

- Une première structure de véhicules appelée **car** qui nous permet de stocker les informations importantes sur un modèle de voiture :
 - L'id de la voiture, noté **id**
 - Le nom du modèle de la voiture, noté **model**
 - Son autonomie maximale en km, notée **autonomyMax**
 - L'autonomie utilisable par la voiture (restreinte par l'utilisateur) en km, notée **autonomyUsable**

- L'autonomie actuelle de la voiture, qui représente le nombre de km parcourables par la voiture à l'instant t , notée `autonomyAct`
- La capacité de batterie la voiture, notée `capacity`.
- Une seconde structure de liste de véhicules appelée `car_list` qui est une liste doubllement chaînée. Elle contient donc un antécédent `prev`, un successeur `next`, un index `index` et une voiture `car`.

2.2.3. Structure d'Horaires

Relationships - Class Diagram

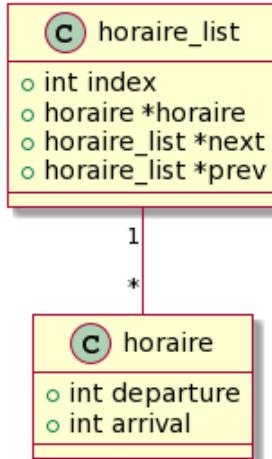


Figure 3: Diagramme WBS du projet

L'implémentation des structures liées aux horaires s'est faite en 2 parties:

- Une première structure d'horaires appelée `horaire` qui nous permet de stocker les informations importantes sur un horaire :
 - L'heure de départ, notée `arrivalTime`
 - L'heure d'arrivée, notée `departureTime`.
- Une seconde structure de liste de véhicules appelée `horaire_list` qui est une liste doublument chaînée. Elle contient donc un antécédent `prev`, un successeur `next`, un index `index` et un horaire `horaire`.

2.2.4. Structure pour le Parser

Relationships - Class Diagram

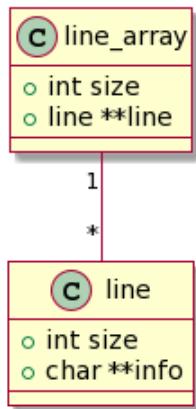


Figure 4: Diagramme WBS du projet

L'implémentation des structures liées au Parser s'est faite en 2 parties:

- Une première structure de ligne appelée `line` qui nous permet de stocker les informations importantes d'une ligne :
 - La taille de la ligne modélisant le nombre d'informations sur une ligne de csv, noté `size`
 - Un tableau de chaînes de caractères contenant chacune une des info dans la ligne du csv, noté `info`
- Une seconde structure de liste de lignes appelée `line_list` qui contient un tableau de `line` se nommant `line` et un attribut `size` modélisant le nombre de lignes du fichier csv.

2.3. Parser des documents

Afin d'accélérer la récupération des données et l'insertion dans nos structures, nous avons décidé d'implémenter notre propre parser de csv très généraliste.

2.3.1. Lecture et récupération des données sous forme de string

A l'aide de la structure `ligne` et `ligne_array` présentées plus haut, nous avons implémenté la fonction `csvParser`, elle prend en entrée le chemin relatif vers le fichier à parser, le nombre de lignes à ne pas lire (comme la première ligne d'entête) et un `line_array`, qui est un tableau préconstruit par l'utilisateur avec le nombre de ligne choisies (qui correspond chez nous au nombre de données intéressantes), au nombre de colonnes du csv puis pour chaque colonne, nous réservons la place suffisante pour le plus grand `string` du csv. Pour déterminer l'indice de la ligne nous l'initialisons à 0 et incrémentons à chaque occurrence du caractère `\n`, nous faisons de même pour la colonne avec l'occurrence du caractère `,`. Nous obtenons ainsi un tableau contenant

2.3.2. Importation vers les structures existantes

Une fois la génération d'une `line_list` contenant l'ensemble des données stockées dans le fichier csv, il nous faut l'importer dans une structure `borne_list` ou `car_list` en fonction du csv donné. Il ne reste alors qu'à créer deux fonctions `borne_list_dataImporter` et `car_list_dataImporter`. Chacune insère dans les structures données en paramètres le

contenu d'une `line_list` en changeant le format de `string` à `int` ou `double` en fonction du type de la donnée dans la structure de destination.

L'ensemble lecture + stockage + transformation s'effectue en environ 60ms pour le fichier `bornes_cleaned.csv` contenant 14000 bornes

2.4. 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.

2.4.1. 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

2.4.2. La solution retenue à la V1

Nous avons finalement opté pour un programme certes un peu similaire, mais opérant différemment. Après avoir généré un graphe en respectant la structure que nous avions implémenté, c'est à dire 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 soit vraie:

```
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 :

```
bornes_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 celà :

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.

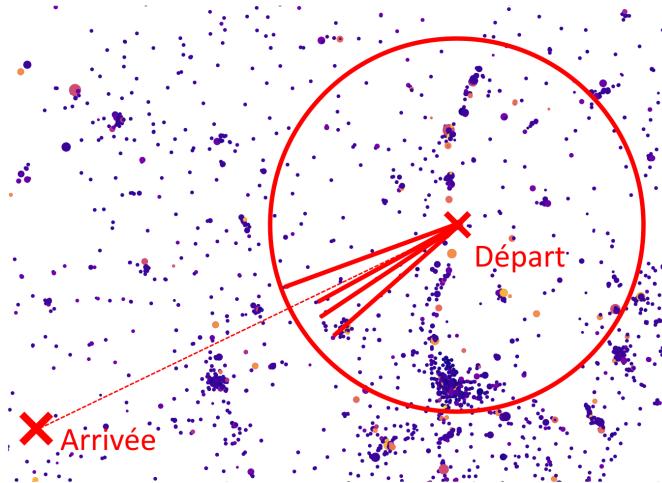


Figure 5: Schéma explicatif de l'algorithme de la partie 1

2.4.3. La solution retenue à la V2

Pour la version 2, nous avons décidé d'implémenter un algorithme plus efficace à petite échelle. Si la borne est accessible, nous l'ajoutons au chemin et terminons l'algorithme de recherche.

```
int pathFinding(car *usedCar, borne *actual, borne *ending, borne_list *path,
                horaire_list *pathTime, borne_list *allListsBorne,
                int maxTimeWaiting, int maxTimeCharging, int *actualTime)

// src/utils/pathfinding.c
```

Sinon nous traçons une ligne droite entre le point de départ et d'arrivée et déterminons un point autour duquel nous traçons un cercle virtuel. Celà nous permet de chercher des zones qui sont dans la bonne direction.

```
int creationZone(borne *actual, borne *ending, double p, car *usedCar,
                  borne_list *allListsBorne, borne_list *Zone,
                  borne_list *path)

// src/utils/pathfinding.c
```

Une fois cette zone établie, si elle est vide alors le chemin est impossible. Sinon nous recherchons la meilleure borne dans cette zone.

```
void updateBestBorne(borne *actual, borne *goal, borne *borneInTest,
                      borne *bestBorne, double *bestTime, int *chargeTime,
                      int *waitingTime, int *travelTimeToGoal,
                      int *travelTimeToBorneInTest, int *bestPdcIndex,
                      car *usedCar, int maxTimeCharging, int maxTimeWaiting,
                      int currentTime)

// src/utils/pathfinding.c
```

Nous l'appelons de nouveau la fonction pathfinding de façon récursive jusqu'à trouver un chemin complet

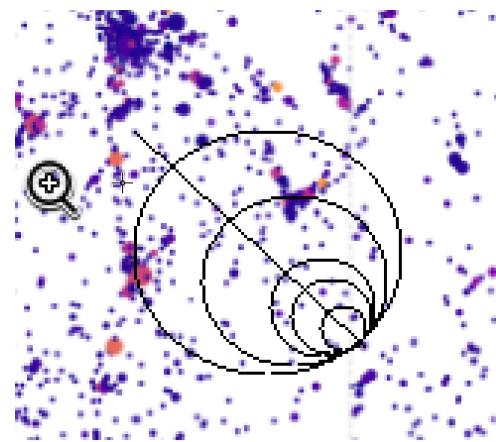


Figure 6: Schéma du fonctionnement de l'algorithme creationZone

3. Tests et complexités

3.1. Tests

Tout au long de l'implémentation des structures, des fonctions associées et des algorithmes, nous avons implémenté et réalisé des tests retrouvable dans le dossier test. L'ensemble des tests implémentés ont tous validés. Ces tests auront été réalisés de façon progressive : dès qu'une structure est implémentée, nous testons toutes les fonctions annexes. De même pour les fonctions de l'algorithme.

3.2. Complexité

3.2.1. Fonction 1 : betterWaitTime

```
bool betterWaitTime(borne *borne, int *watingTimeAct, int *betterPdcIndex,
                    int *currentTime, int *chargeTime) {
    int waitingTime = INT16_MAX;
    for (int i = 0; i < borne->pdc; i++) {
        getWaitingTimeAndPotentialChargeTime(borne->horairePdc[i], &waitingTime,
                                              betterPdcIndex, currentTime,
                                              chargeTime);
        if (waitingTime == 0) {
            *betterPdcIndex = i;
        }
    }
    bool b = waitingTime < *watingTimeAct;
    if (b) {
        *watingTimeAct = waitingTime;
        // printf("AAAAAAAAAAAAAAAAAAAAAAA\n");
    }
    return b;
}
```

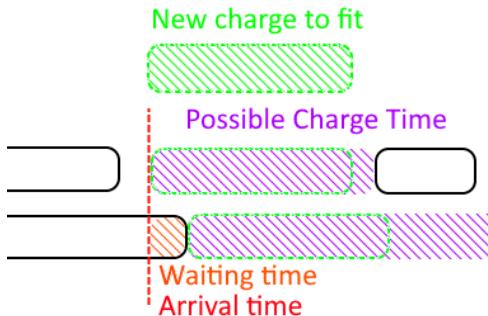


Figure 7: Schéma du fonctionnement de l'algorithme betterWaitTime

Ce schéma représente la manière dont on choisit la meilleure borne de recharge à laquelle peut se brancher l'utilisateur. Cet algorithme est en $O(pdc \times \max(\text{horaire_list_length}(\text{horairePdc})))$ en prenant en compte la complexité de la fonction `getWaitingTimeAndPotentialChargeTime`

3.2.2. Fonction 2 : pathFinding

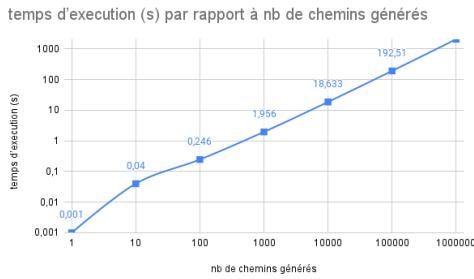


Figure 8: Schéma du fonctionnement de l'algorithme pathFinding (double échelle logarithmique)

L'algorithme est linéaire en nombre de chemin, la génération de 1000 chemin prend en moyenne 2s.

4. Gestion de projet

4.1. Composition et organisation du groupe

Membres de l'équipe :

Nom et mail	Rôles
THUILLIER Killian killian.thuillier@telecommancy.eu	Développer Débugger Lead réunions
TIGER Maelan maelan.tiger@telecommancy.eu	Développer Secrétaire Conception d'algorithmie
VESSE Léo leo.vesse@telecommancy.eu	Développer Débugger Responsable Organisationnel

4.2. Réunions

Contrairement au premier épisode de ce projet, nous avons organisé le projet en sessions de travail journalières de 7h organisées comme suit :

- 13h : réunion de planification des tâches de 30min
- 13h30 - 16h : Session de travail
- 16h : Réunion d'avancement de 30min
- 16h30-17h : Pause
- 17h30 - 19h30 : Session de travail
- 19h30 - 20h : Réunion de fin de journée, bilan de fin de journée

Après avoir établi le Gant du projet, nous avons choisi de se rajouter des temps de travail en distanciel si nous étions en retard sur le Gantt.

Il y a donc un CR de réunion par jour contenant les notes et décisions prises au cours des 3 micro-réunions quotidiennes.

4.3. Répartition des tâches

Tout au long du projet, toute l'équipe a participé à tout les types de tâches.

4.3.1. WBS

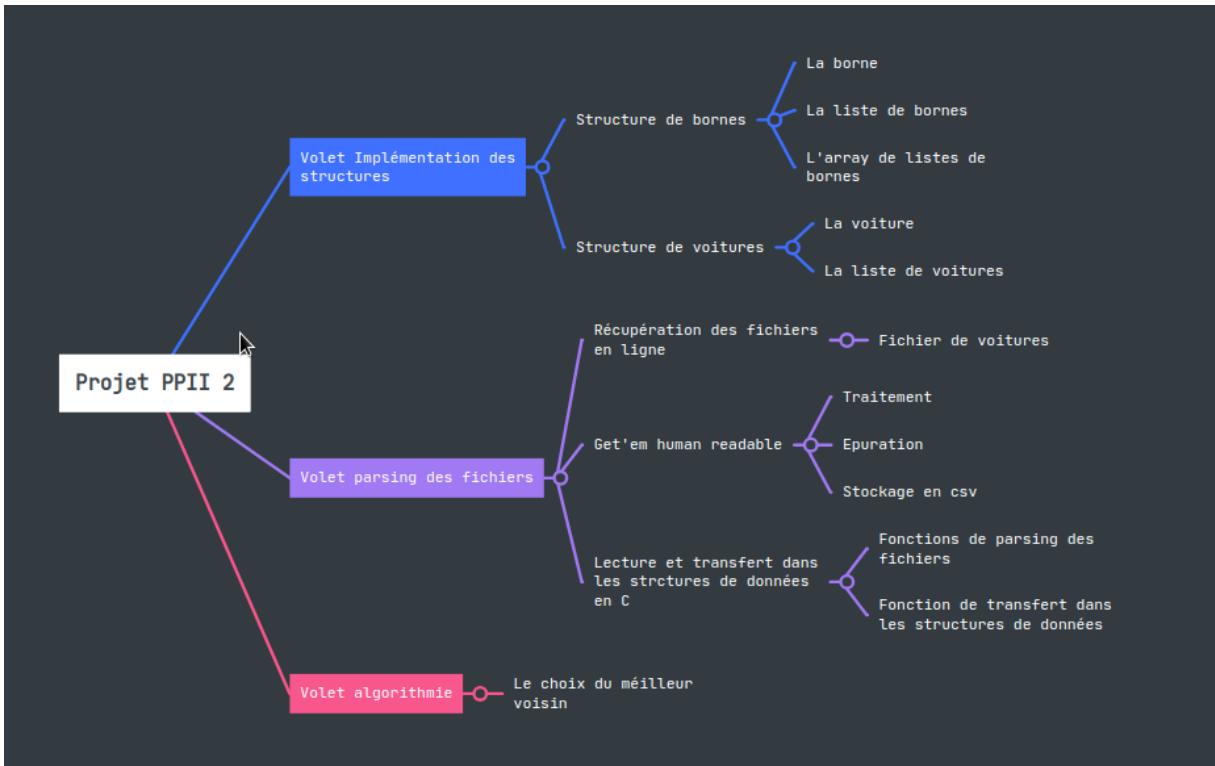


Figure 9: Diagramme WBS du projet

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. 9a 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

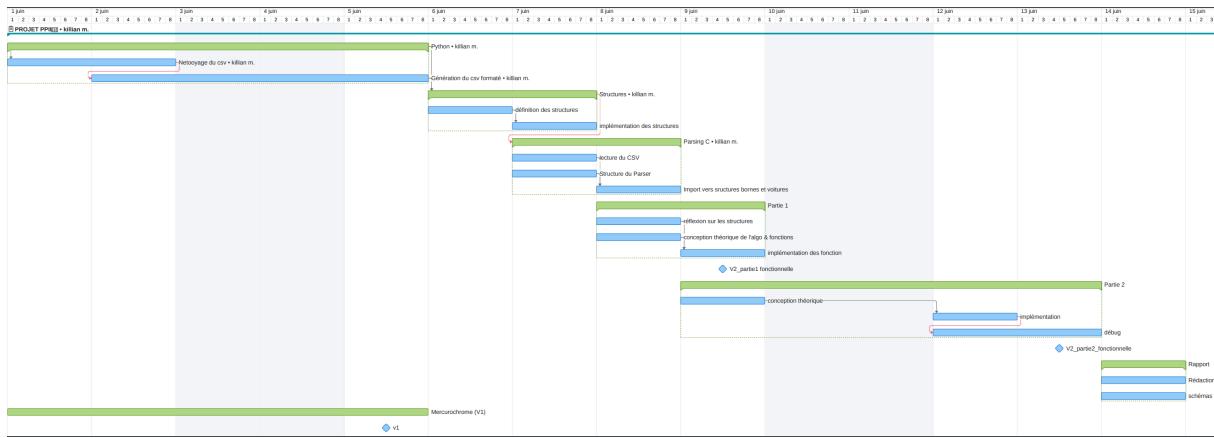


Figure 10: Diagramme de Gantt prévisionnel

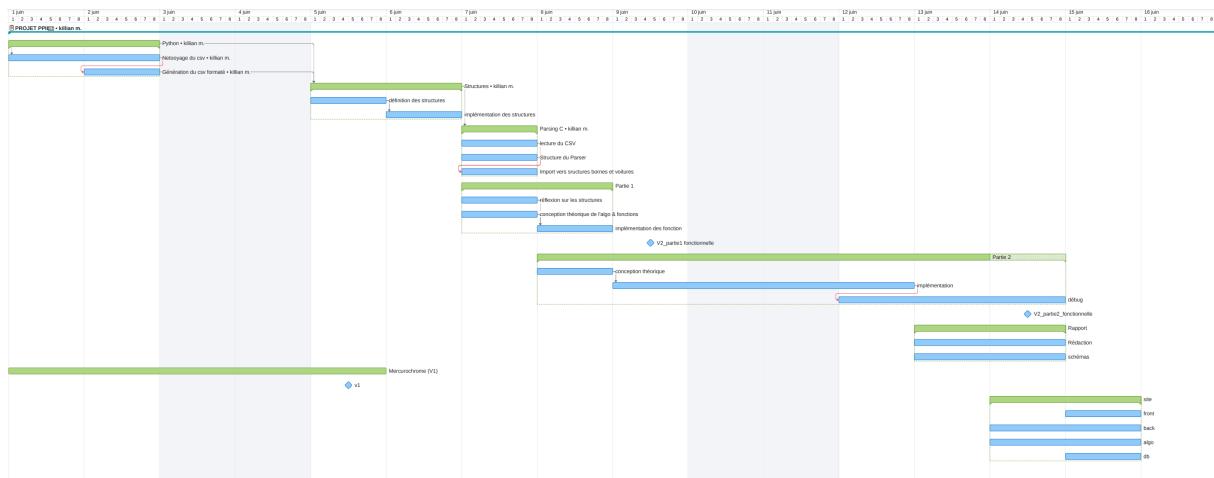


Figure 11: Diagramme de Gantt réel

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 W2 - Ne pas suivre le Gantt
Source externe	O1 - Travailler en groupe O2 - Développer nos compétences informatiques O3 - Pas de cours	T1 - Perte de motivation T2 - Autres occupation prévues

Table 1: Matrice SWOT du projet

ID	Description
S1	Nous avons utilisé la licence Typst 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. De plus, nous connaissons le sujet sur le bout des doigts, ce qui permet un démarrage rapide.
W1	L'implémentation pourrait s'avérer compliquée dans le langage rigoureux qu'est le C.
W2	Recommencer le projet de zéro peut être une source de démotivation et les membres peinent à se détacher du projet.
O1	Le groupe était soudé et très communiquant.
O2	Recommencer le projet est utile pour approfondir encore plus nos compétences informatiques.
O3	Le projet a été développé pendant la phase post examens et pré stage. Nous n'avions donc aucun cours, ce qui a permis d'organiser des sessions de travail en présentiel tous les jours de la semaine.
T1	Avec la fin des cours et l'été, nous n'avons pas envie d'être enfermés à coder.
T2	L'ensemble des membres de l'équipe a aussi pas mal de responsabilités ou d'occupations déjà prévues. Ce projet et ses dates nous sont imposées.

Table 2: Description et analyse des éléments de la SWOT

5. Bilan du projet

5.1. Améliorations

5.1.1. Fonctionnalités abandonnées

La création de la zone devait dans un premier temps avoir un point de central proche de la distance que la voiture peut parcourir et non un point central proche de l'arrivée. Pour des raisons de temps et de complexité à implémenter pour un gain de temps faible, cette option a été abandonnée.

La représentation graphique de la charge des zones a été abandonnée pour des raisons de temps

L'algorithme force une recharge à 100% ou bien un temps de charge maximisé au `maxTimeChrging` donné en paramètre par l'utilisateur. La raison pour laquelle nous avons décidé de ne pas optimiser la charge pour le trajet est que cela aurait imposé un backtracking et un temps de calcul bien trop élevé

5.1.2. Fonctionnalités à améliorer

Eviter de jouer entre les différentes unités pourrait nous permettre dans des cas très particuliers d'optimiser des trajets et réduire le temps de réflexion sur les conversions. Il est actuellement nécessaire de définir en tant que variable globale le nombre de lignes, de colonnes, et le nombre de caractères pour l'import. Il serait intéressant pour avoir

un parser le plus général possible d'avoir une fonction qui parcourt le fichier une fois et qui détermine toutes ces constantes du fichier.

5.2. Conclusion générale

En conclusion, ce projet de création d'une application pour trouver une suite de bornes de recharge de voitures électriques en langage C a été une entreprise passionnante et réussie. L'application développée offre une solution pratique et conviviale pour les propriétaires de voitures électriques, leur permettant de localiser facilement les bornes de recharge les plus proches, d'optimiser leurs trajets et de planifier efficacement leurs déplacements.

L'utilisation du langage C a permis de développer une application performante et robuste, avec des fonctionnalités avancées telles que la cartographie, la recherche de bornes par critères spécifiques (vitesse de charge, disponibilité, etc.) et des indications détaillées pour atteindre chaque borne sélectionnée. Les algorithmes de recherche ont été soigneusement optimisés pour garantir des résultats précis et rapides, offrant ainsi une expérience utilisateur fluide.

En plus de fournir une solution pratique aux utilisateurs, cette application contribue également à promouvoir l'utilisation de voitures électriques en facilitant l'accès à un réseau de bornes de recharge. Cela peut encourager davantage de personnes à adopter des véhicules électriques, contribuant ainsi à réduire les émissions de carbone et à protéger notre environnement.

Bien que ce projet ait été couronné de succès, il existe encore des possibilités d'amélioration. Des fonctionnalités supplémentaires telles que l'intégration de la navigation en temps réel, l'affichage de la disponibilité en temps réel des bornes de recharge et l'intégration avec d'autres applications de covoiturage ou de planification de trajets pourraient être envisagées. De plus, l'extension de la base de données des bornes de recharge et la mise à jour régulière des informations garantiront une expérience utilisateur toujours plus enrichissante.

En conclusion, ce projet a permis de développer une application en langage C offrant une solution pratique, performante et conviviale pour trouver des bornes de recharge de voitures électriques. Il constitue une contribution significative à la promotion de la mobilité électrique et démontre le potentiel du langage C dans le développement d'applications innovantes.

5.3. Conclusions personnelles

5.3.1. THUILLIER Killian

Ce deuxième essai sur le projet a été bien plus agréable à vivre pour moi que le précédent. En effet, tout mon groupe était motivé tous les jours même si un quatrième membre aurait été précieux. Au rythme de 7h par jour pendant 10 jours, c'est beaucoup, surtout dans un laps de temps si court. J'estime vraiment que ce deuxième projet est la version qui aurait du voir le jour dès le début, en tout cas c'était notre ambition dès le départ. Aujourd'hui, je trouve enfin le plaisir de rendre un travail dont je n'ai pas honte. De

plus, je trouve vraiment que cette version du projet s'est déroulé de façon parfaite, et la gestion de projet était optimale.

5.3.2. VESSE Léo

10 jours à 7h par jours c'est très long, surtout lorsqu'on est censé être en vacances. Mais bon on a fait avec. Une quatrième personne n'aurait pas été de trop je pense. Malgré tout on est arrivé au bout de ce PPII, potentiellement avec un site fonctionnel, une simulation etc. Concevoir à plusieurs est une réelle épreuve, il faut s'accorder sur la logique à suivre, et donc s'imposer d'écrire les algorithmes sur du papier avant de les implémenter. Celà nous a permis de débugger facilement. Celà m'a permis de me replonger dans l'algorithmie, j'aurais voulu expérimenter neovim sur ce projet mais je n'ai pas osé faire perdre du temps à l'équipe, ce sera pour prochaine fois. PS: Pouvoir au `typst`, il est plus modulable que le `latex`, il est encore en développement et pourtant il pourtant ses possibilités sont plus grandes que celles du `latex`

5.3.3. TIGER Maelan

Ce remake du projet de programmation en C s'est déroulé vraiment bien compte tenu du contexte. Je ne me suis pas vraiment rendu compte à quel point il était épuisant de travailler tous les jours de façon soutenue sur de longues plages de travail. Cependant, le fait que nous travaillions tous dans la même pièce, échangeant des idées les uns avec les autres, rendait l'expérience vraiment motivante. Je suis un peu triste que notre programme ne soit pas parfaitement propre, je pense qu'avec plus d'expérience ou de temps il aurait pu l'être, mais je pense que nous sommes en train de livrer quelque chose de tout à fait correct pour notre niveau. PS : Je suis convaincu que Typst a la capacité de remodeler le visage des documents compilés

6. Annexes

6.1. Comptes rendus

Compte Rendu n°1

2 Juin 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 02/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Etape 1 : Python, nettoyage CSV	Léo & Maelan	<input checked="" type="checkbox"/>
Patch « Mercurochrome »	Killian	<input type="checkbox"/>

Ordre du jour

1. Etape 1 : Python & CSV parsing
2. Patch « Mercurochrome »
3. Structures de données

Informations échangées

Partie 1 du Gantt terminée avec 2 jours d'avance

Etape 1 : Python & CSV parsing

1. Nettoyage BD fournie
2. Choix formattage des données
3. Regroupement des bornes
4. Ecriture en CSV

Patch « Mercurochrome »

Gestion des listes d'horaires

- Problème sur l'index des éléments : Mauvaise initialisation → **Résolue**
- Problème d'horaire de départ : Probablement mauvaise initialisation de la première valeur → **Résolue**
- Ajout d'un nouvel horaire : Horaires de départ et d'arrivée identiques → **Résolue**

Calcul de distance

- Problème sur récupération distance depuis le CSV des distances avec les voisins
- Problème avec la fonction de calcul de distance en C

↳ Debug d'au moins une des deux fonctions et utilisation dans la totalité du code V1.

Structures de données

Organisation pour l'étape 2 du gantt : Structure de données

- Design des structures
- Choix des fonctions à implémenter
- Algorithmie
- Implémentation

Avancement à la fin de la journée

- Design de toutes les structures
- Signatures des méthodes
- Méthodes pour les bornes

Décisions

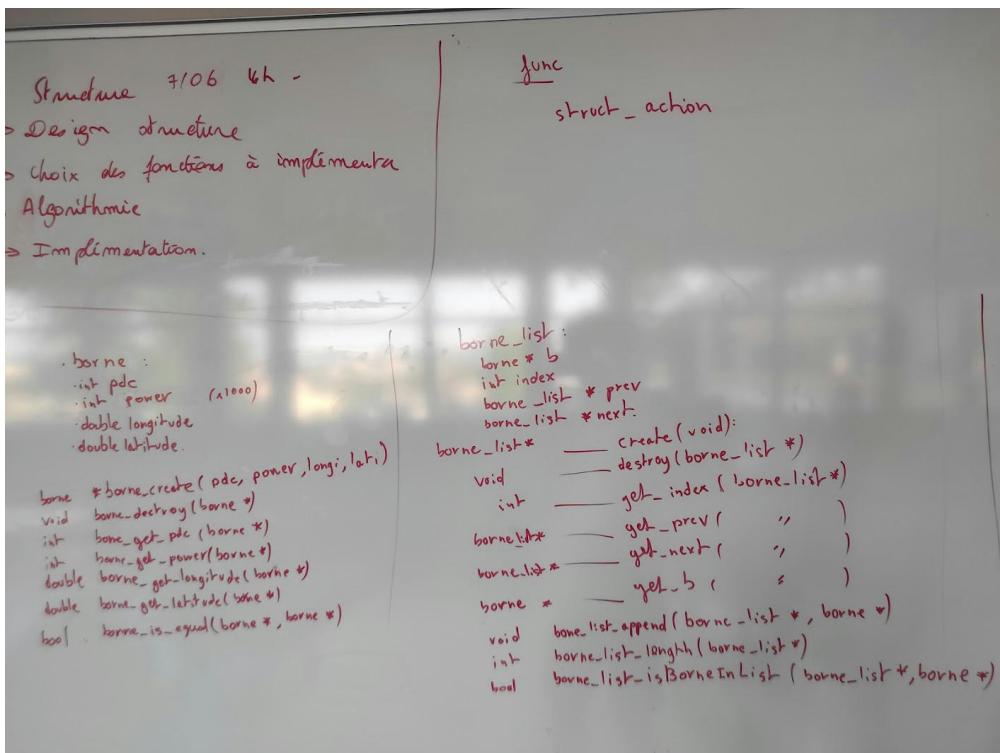
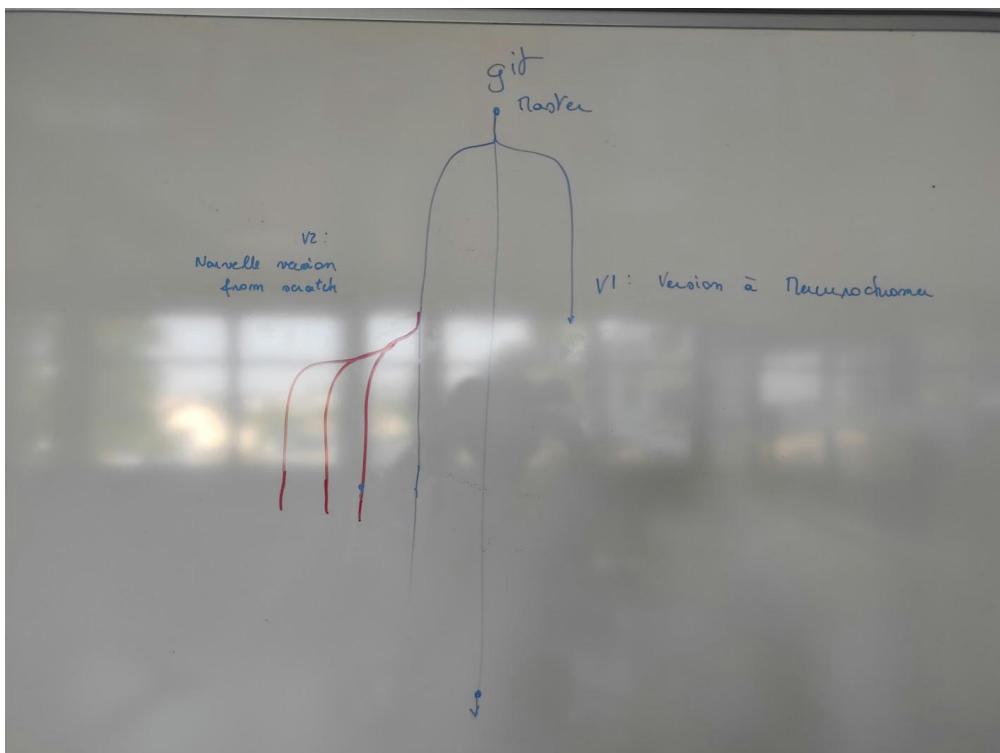
- Séparation des différentes structures de données et fonctions associées en modules indépendants → parallélisation possible

TODO List

Description	Responsable	Délai	Livrable	Validé par
Réflexion sur les structures de données	Léo & Maellan	02/06/23 au soir	Schéma structurel Header fonctions Algorithmie générale	Tout le monde
Implémentation des structures de données	Léo & Maellan	07/06/23 (pause)	Structures Fonctions de manipulation associées	Tout le monde
Patch « Mercuro-chrome » : Calcul de distance Double free	Killian	05/06/2023 au soir	V1 fonctionnelle	Tout le monde

Date de la prochaine réunion : 05/06/2023

Documents additionnels



```
car :  
    char * name;  
    int autonomy;  
    double powerCap;  
    int id;  
  
nature * car_create (char *, int, double);  
void car_destroy (car *);  
int car_get_Autonomy (car *);  
double car_get_PowerCap (car *);  
char * car_get_Name (car *);  
bool car_is_Equal (car *, car *);
```

```
car_list :  
    car * C;  
    car_list * next;  
    car_list * prev;  
    int index;
```

cf: borne-list.h.

Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
• complet	<input checked="" type="checkbox"/>
• motivant	<input checked="" type="checkbox"/>
• envoyé à l'avance	<input checked="" type="checkbox"/>
Introduction :	
• tous les objectifs rappelés	<input checked="" type="checkbox"/>
• gestion du temps annoncée	<input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	~
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
• gestion du temps	<input checked="" type="checkbox"/>
• sujets importants traités	<input checked="" type="checkbox"/>
• pas de perte de temps	~
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte Rendu n°2

5 Juin 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 5/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Réflexion sur les structures de données	Léo & Maelan	<input checked="" type="checkbox"/>
Implémentation des structures de données	Léo & Maelan	<input checked="" type="checkbox"/>
Patch « Mercurochrome » : Calcul de distance Double free	Killian	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

Ordre du jour

1. Avancement V1
2. Avancement V2

Informations échangées

Avancement V1

- Tous les paths sont bons
- ⚠ le CSV distances n'est plus du tout utilisé → sera supprimé du projet
- Patch de l'erreur double free → en cours
- Code refactor → à venir
- Code commenting → à venir
- Code cleaning → à venir
- Création d'une branche V1 debug avec verbose en parallèle d'une branche V1 release
- app.c contiendra les 2 modes : Choix d'une voiture + trajet & création aléatoires de trajets de voitures

Avancement V2

- Réflexions sur les fonctions *_get et *_set quant à leur utilité par rapport au -> du C

- Rédaction des corps des fonctions
- Ecriture de tests pour vérifier l'intégrité des structures de données réalisées
- L'implémentation de `borne.c` et `borne_list.c` est terminée, une première vague de tests basiques ont permis de nettoyer quelques erreurs d'initialisation

Décisions

- L'erreur de `double-free` du patch de la V1 n'est pas résolu à ce jour, il ne sera donc plus travaillé durant les heures de travail définies, mais Killian souhaite continuer à travailler en autonomie dessus → possibilité d'implémenter 2 méthodes de calcul dont la complexité temporelle diffère et ainsi privilégier la plus optimale selon la demande de l'utilisateur
- Un document d'aide (en markdown) contenant les instructions des différentes fonctions va être réalisé tout au long des sessions de travail
- L'implémentation de branches de développement individuelles pour la V2 sont une bonne idée pour éviter la suppression éventuelle de commits

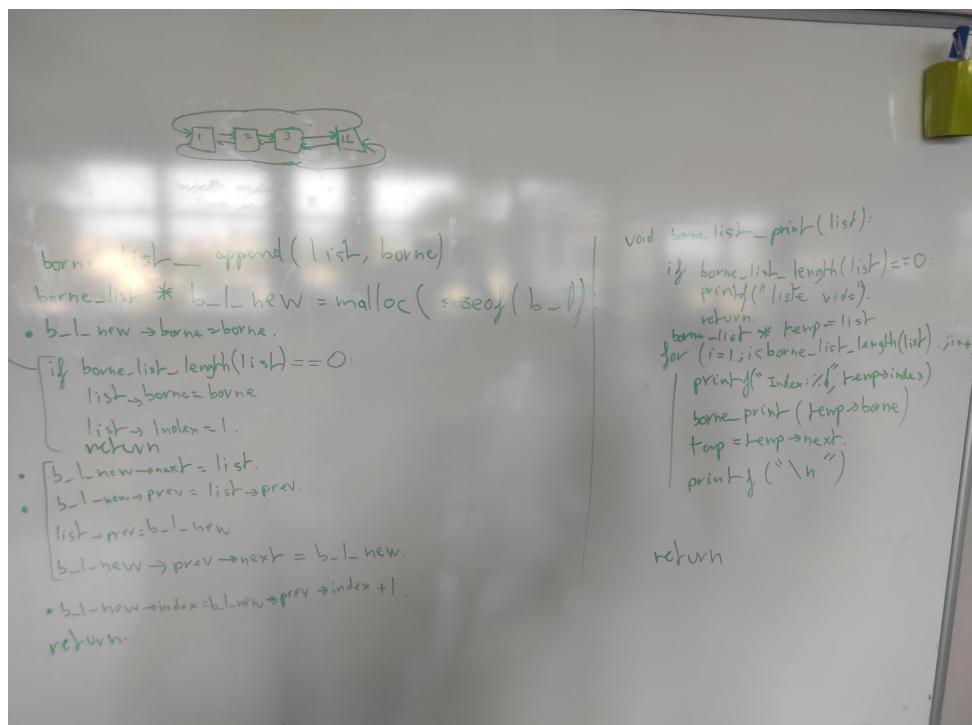
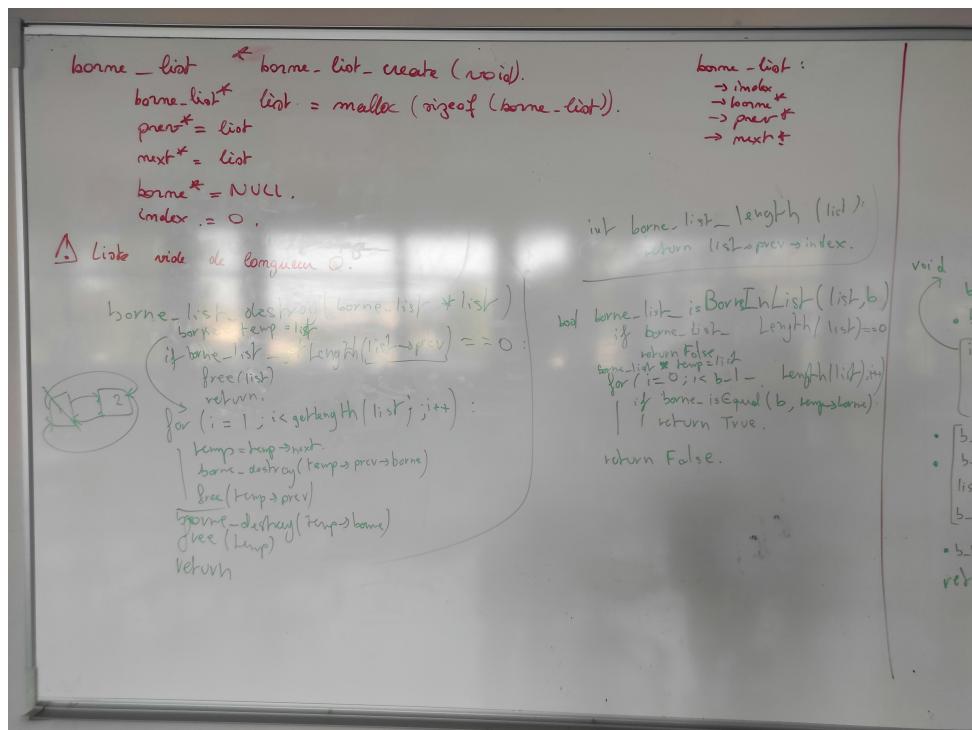
TODO List

Description	Responsable	Délai	Livrable	Validé par
Implémentation des fonctions de SD	Léo & Maelan	06/06/23 (pause)	Totalité des corps des fonctions	Tous
Correction des fonctions de SD	Léo & Maelan	06/06/23 (pause)	Validation des tests des fonctions	Tous
Code refactor	Killian	06/06/23 (pause)	Code V1 refactor	Tous
Code commenting	Killian	06/06/23 (pause)	V1 commentée	Tous
Code cleaning	Killian	06/06/23 (pause)	V1 nettoyé	Tous

Formattage de CSV optimisé: Bornes Voitures Tests de parsing	Tous	06/06/23 (pause)	CSV et fonctions de parsing	
---	------	------------------	-----------------------------	--

Date de la prochaine réunion : 06/06/23

Documents additionnels



Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
<ul style="list-style-type: none"> • complet • motivant • envoyé à l'avance 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Introduction :	
<ul style="list-style-type: none"> • tous les objectifs rappelés • gestion du temps annoncée 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	<input checked="" type="checkbox"/>
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
<ul style="list-style-type: none"> • gestion du temps • sujets importants traités • pas de perte de temps 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte Rendu n°3

6 Juin 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 5/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Implémentation des fonctions de car Correction des fonctions de borne Validation des tests des fonctions associées	Maelan	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Formattage de CSV optimisé: Parser Bornes Voitures Tests de parsing	Killian & Léo	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

Ordre du jour

1. Retour sur la V1
2. Retour sur les tests de structures
3. Rédaction des algorithmes `parser`

Informations échangées

Retour sur la V1

- La date limite fixée étant dépassée, la V1 est fonctionnelle, mais le `double-free` n'est pas encore réglé.
- Il manque encore le commenting et le refactoring

Retour sur les tests de structures

- Les fonctions `*_destroy` avaient un problème d'indice limite → réglé
- `car` et `car_list` implémentée
- `car_list_test` implanté
- Création d'un `timer.c` pour faciliter les tests de complexité temporelle des fonctions

Rédaction des algorithmes `parser`

- Implémentation des structures de bases `line` et `csv_import`
- Création des fonctions pour l'importation des données depuis les CSV vers les structures définies ci-dessus

Décisions

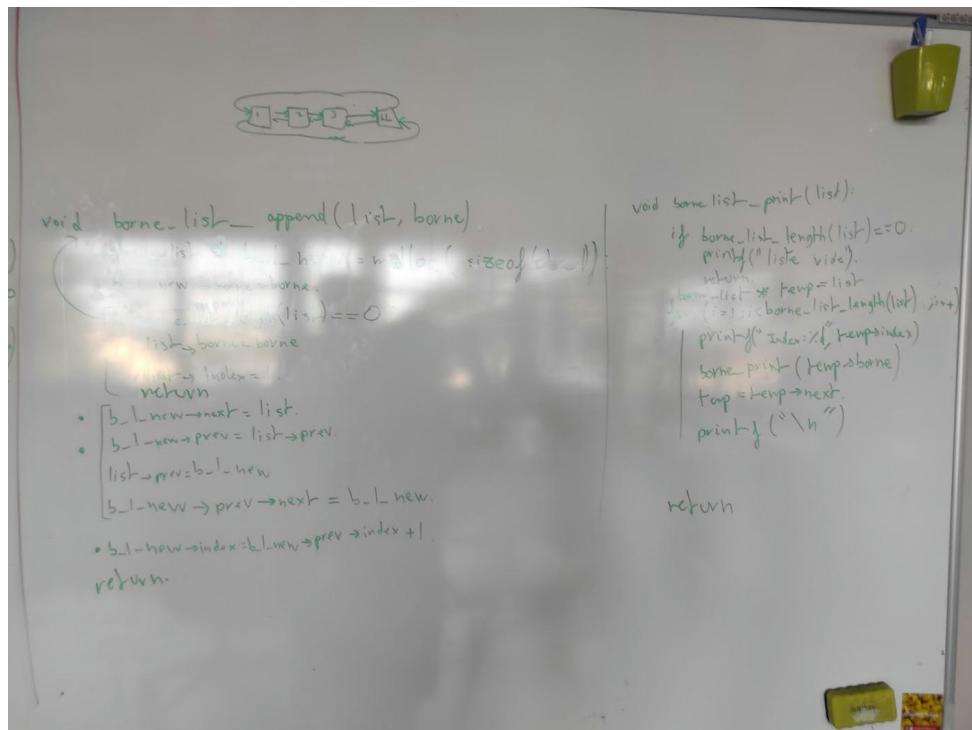
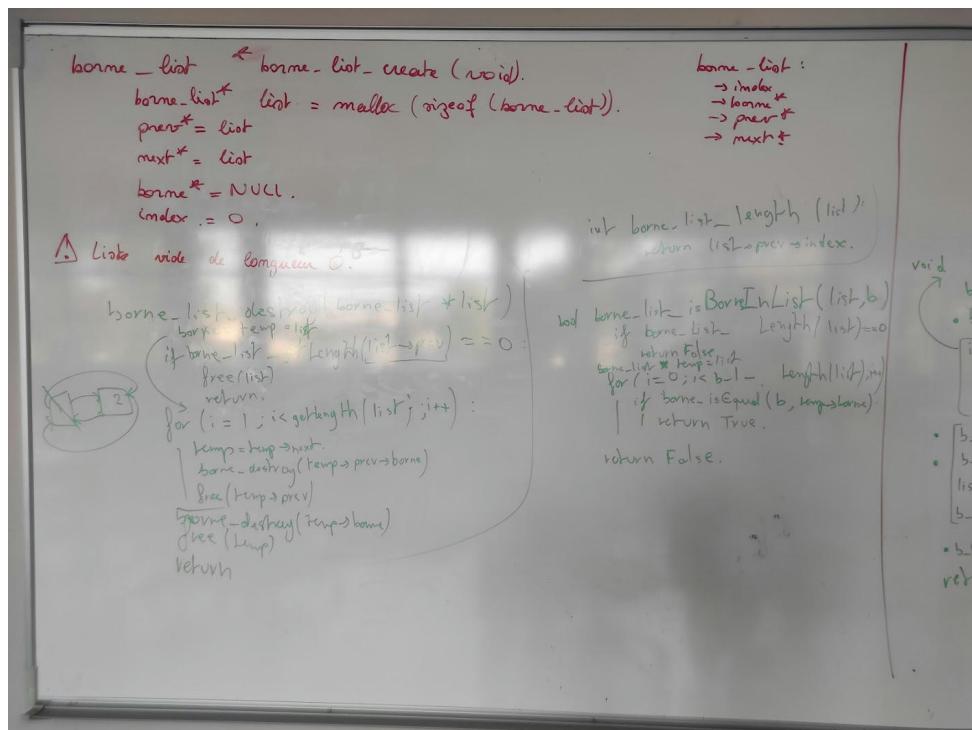
- Aucun travail sur la V1 ne sera effectué durant les sessions de travail définies, elle pourra cependant être poursuivies en travail supplémentaires (ou si la V2 est terminée d'ici là)

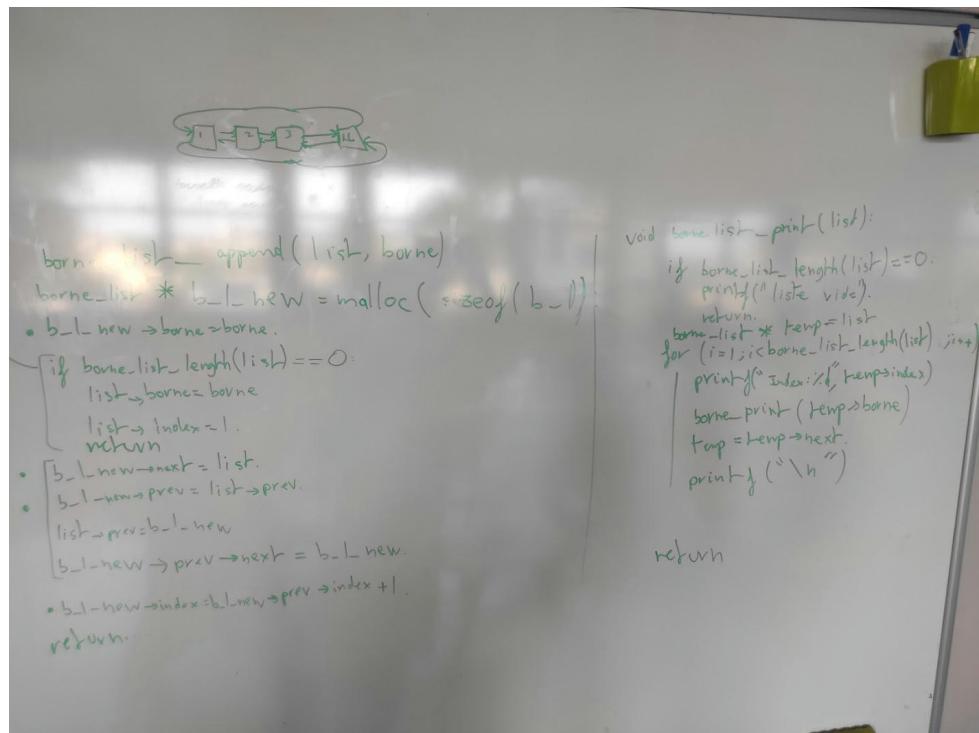
TODO List

Description	Responsable	Délai	Livrable	Validé par
Fonctions du parser	Léo	07/06/23 (soir)	Fonctions de parsing	Tous
Tests du parser (lorsque les fonctions sont finies)	Killian	07/06/23 (soir)	Tests de parsing	Tous
Documentation des nouvelles fonctions	Maelan	07/06/23 (pause)	Fichier markdown mis à jour	Tous
Conception de l'algorithme Partie 1	Maelan & Killian	07/06/23 (soir)	Algorithmes théoriques Fonctions utilitaires Header des fonctions principales	Tous

Date de la prochaine réunion : 07/06/23

Documents additionnels

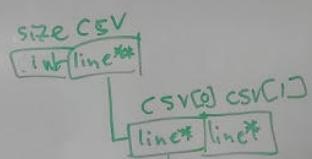




```
Struct csv-import {
```

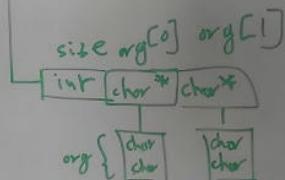
```
    int size  
    line** CSV[100]
```

```
}
```



```
Struct line {  
    int size  
    char** arr[ ],
```

```
}
```



Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
• complet	<input checked="" type="checkbox"/>
• motivant	<input checked="" type="checkbox"/>
• envoyé à l'avance	<input type="checkbox"/>
Introduction :	
• tous les objectifs rappelés	<input checked="" type="checkbox"/>
• gestion du temps annoncée	<input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	<input checked="" type="checkbox"/>
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
• gestion du temps	<input checked="" type="checkbox"/>
• sujets importants traités	<input checked="" type="checkbox"/>
• pas de perte de temps	<input checked="" type="checkbox"/>
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte rendu réunion n°4

7 Juillet 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 07/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Fonctions du parser	Léo	<input checked="" type="checkbox"/>
Tests du parser (lorsque les fonctions sont finies)	Killian	<input type="checkbox"/>
Documentation des nouvelles fonctions	Maelan	<input checked="" type="checkbox"/>
Conception de l'algorithme Partie 1	Maelan & Killian	<input checked="" type="checkbox"/>

Ordre du jour

1. Retour sur le parser
2. Retour sur la conception des algorithmes
3. Point sur l'état mental et physique des membres du groupe

Informations échangées

Retour sur le parser

- Fonctions de base sur l'ouverture des fichiers et du traitement de chaque ligne → **implémentées**
- Modification de la fonction d'insertion du fichier CSV vers nos structures de données **CSV_import implémentée**
- Tests envisagés : Importation de **borne_clean**, mesures de complexités temporelles des fonctions et résultats dans les en-tête des fonctions
- Une fois les tests implémentés, si une phase de debug est nécessaire elle sera assurée par Léo.
- Les tests sont implémentés

Retour sur la conception des algorithmes

- Du retard à été pris sur le démarrage de cette partie (1h de retard)
- La quasi-totalité des algorithmes ont été théorisé en 2h

- L'implémentation des fonctions sera réalisée par Maelan jusqu'à ce soir, Léo et Killian alterneront entre cette implémentation et les tests du parsing
- A l'exception de la fonction `pathfinding`, toutes les fonctions ont été implémentées, mais pas encore testées

Point sur l'état mental et physique des membres du groupe

- D'un point de vue physique, Killian et Maelan sont fatigués
- D'un point de vue mental, Léo est fatigué

Décisions

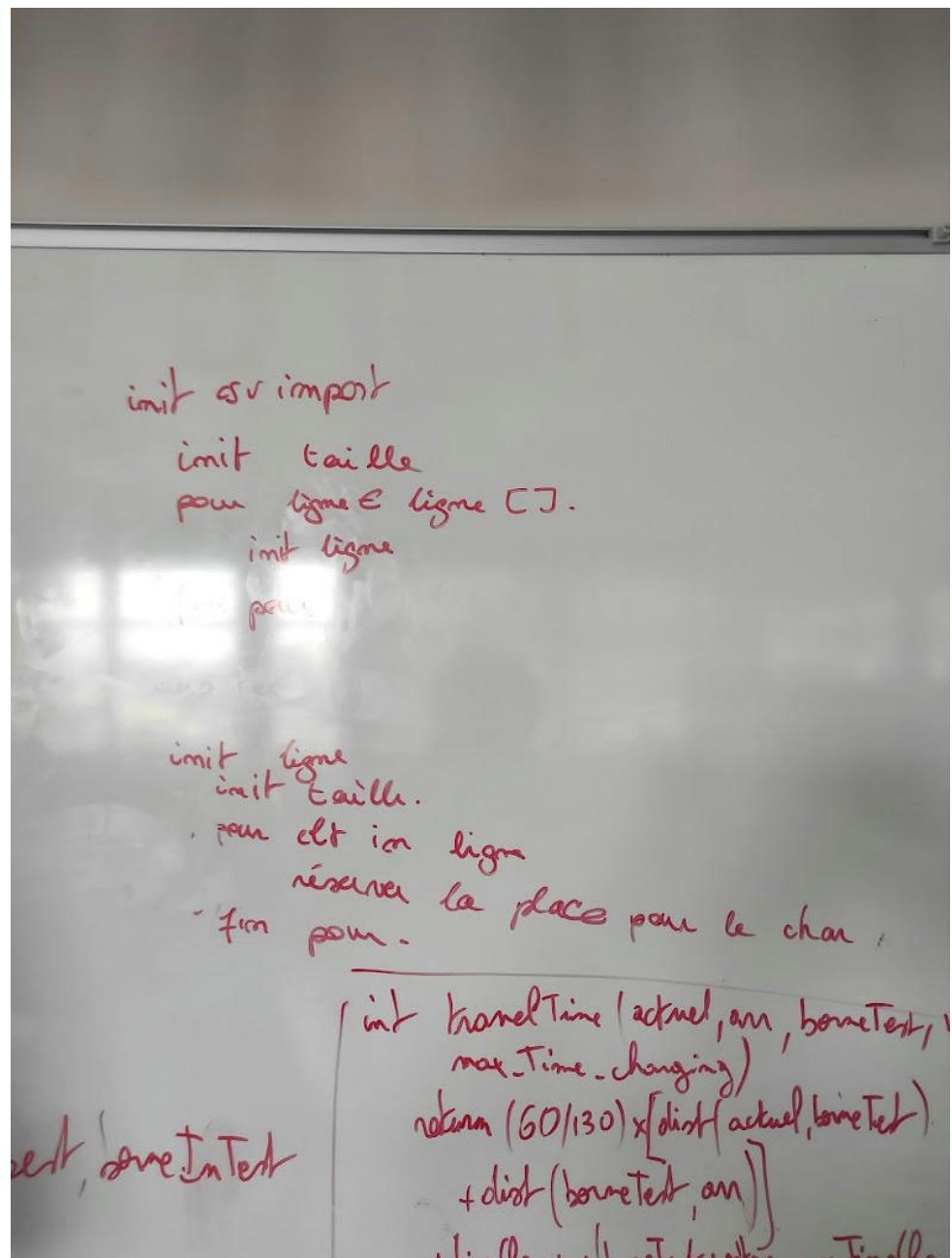
- La création de la structure `vecteur` ne sera pas implémentée, en effet la mesure des distances entre coordonées se fait à l'aide de la norme de Haversine qui est non-planaire, l'équivalent de la norme en coordonnées sphériques.

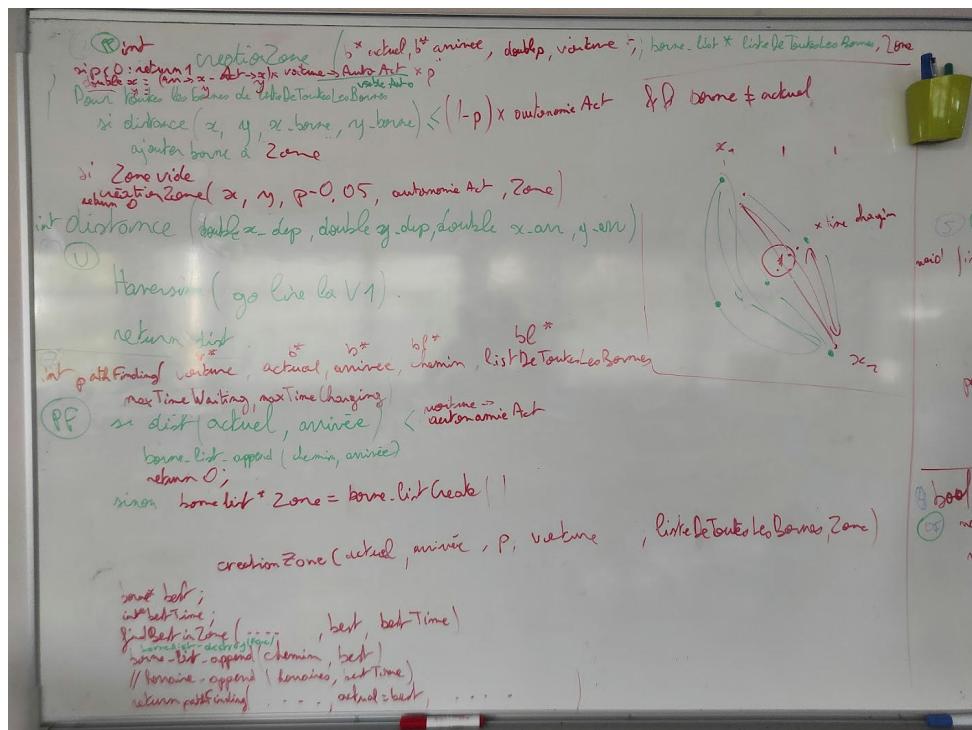
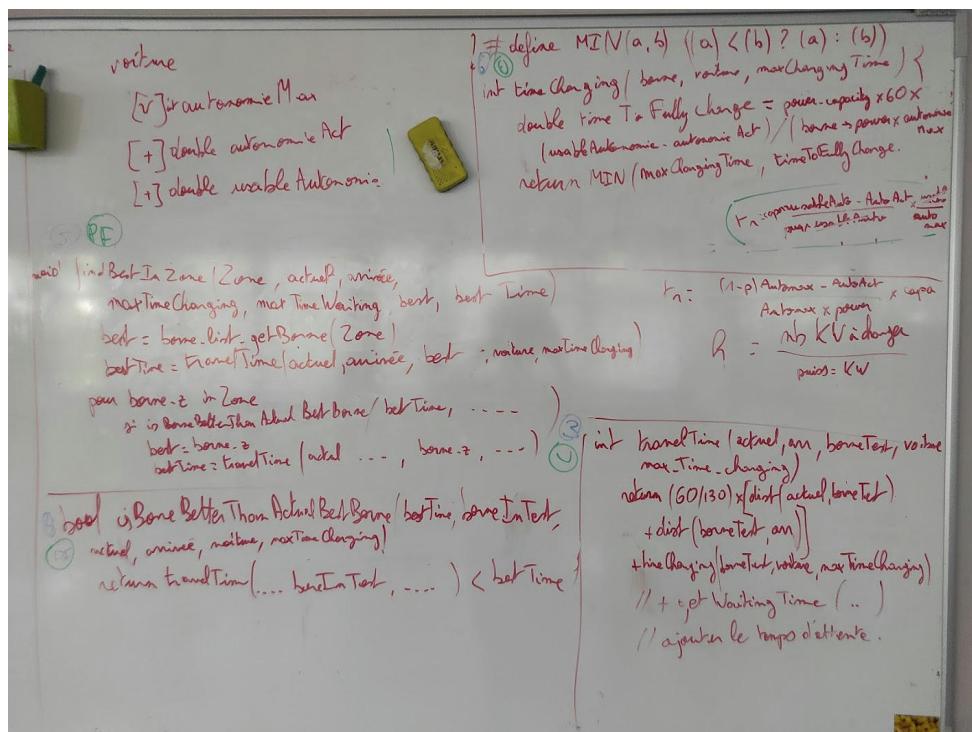
TODO List

Description	Responsable	Délai	Livrable	Validé par
Debug de parser	Léo	08/06/23 (pause)	Parser fonctionnel	Tous
Implémentation <code>pathfinding</code>	Maelan	08/06/23 (pause)	<code>pathfinding.c</code> terminé	Tous
Implémentation tests <code>pathfinding</code>	Killian	08/06/23 (soir)	Tests <code>pathfinding</code> écrits	Tous
Debug <code>pathfinding</code>	Léo	09/06/23 (soir)	Tests et fonctions sans bug	Tous
Conception des modifications de la partie 2	Maelan & Killian	09/06/23 (soir)	Tableau des modifications à apporter	Tous

Date de la prochaine réunion 08/06/23

Documents additionnels





Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
<ul style="list-style-type: none"> • complet • motivant • envoyé à l'avance 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Introduction :	
<ul style="list-style-type: none"> • tous les objectifs rappelés • gestion du temps annoncée 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	~
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
<ul style="list-style-type: none"> • gestion du temps • sujets importants traités • pas de perte de temps 	~ <input checked="" type="checkbox"/> ~
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte rendu réunion n°5

8 Juillet 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 07/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Debug de parser	Léo	<input checked="" type="checkbox"/>
Implémentation pathfinding	Maelan	<input checked="" type="checkbox"/>
Imlémentation tests pathfinding	Killian	<input type="checkbox"/>
Debug pathfinding	Léo	<input type="checkbox"/>
Conception des modifications de la partie 2	Maelan & Killian & Léo	<input checked="" type="checkbox"/>

Ordre du jour

1. Retour sur le parser
2. Retour sur le pathfinding
3. Avancement sur l'implémentation des tests
4. Avancement sur la théorie de la partie 2

Informations échangées

Retour sur le parser

- Les tests passent sans problème.
- Le temps d'importation et de parsing sont de l'ordre de 30 millisecondes pour les bornes, et $\lesssim 1\text{ms}$ pour les voitures
- Le parser a été réalisé de façon générale, et peut s'adapter à n'importe quel fichier CSV et sera probablement réutilisé ultérieurement, même en dehors de ce projet

Retour sur le pathfinding

- Toutes les fonctions annexes sont implémentées, il ne reste que la fonction pathfinding à implémenter, ainsi que les tests et le debug éventuel.
- L'implémentation, le merge et la correction des fichiers a été faite

Avancement sur l'implémentation des tests

- Les tests pour `travelTime`, `timeToCharge`, `distance` sont implémentés
- L'implémentation de multiples tests pour les différents comportements attendus de `creationZone` est en cours
- Le test de `findBestInZone` est plus complexe à appréhender, mais une modification temporaire des caractéristiques des bornes est envisagée : L'utilisation de la région parisienne est privilégiée pour les tests
- Les tests de `pathfinding` vont générer des `string` utilisable dans Google Maps afin de vérifier la faisabilité du chemin généré

Avancement sur la théorie de la partie 2

- Réflexion sur la façon d'implémenter les contraintes d'horaires et de temps d'attente entre les véhicules :
 - L'algorithme va prioriser les trajets dans l'ordre de traitement
 - Il étudie la possibilité d'insérer la phase de chargement requise entre deux utilisateurs
 - Il limite le temps d'attente des utilisateurs
 - Il maximise l'utilisation des différents points de charge des bornes, ainsi optimisant l'utilisation du réseau

Décisions

-

TODO List

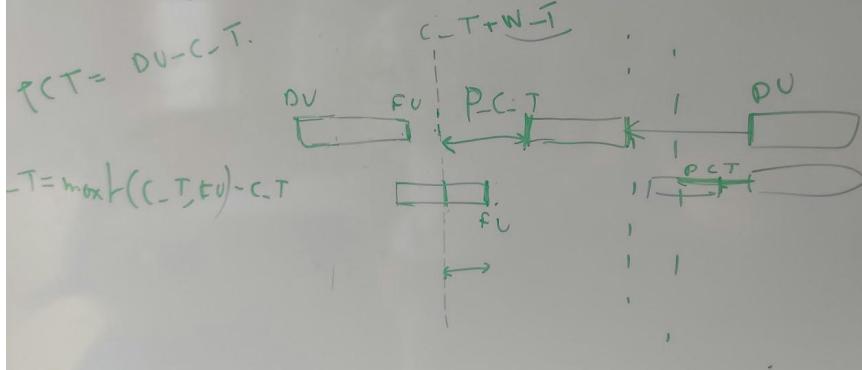
Description	Responsable	Délai	Livrable	Validé par
Debug <code>pathfinding</code>	Léo	09/06/23 (soir)	Tests et fonctions sans bug	Tous
Implémentation des modifications de la partie 2	Maelan & Killian	09/06/23 (soir)	Implémentation des modifications terminée	Tous

Date de la prochaine réunion 09/06/23

Documents additionnels

```
struct horaire:  
    int time_dep  
    int time_arr  
  
hori=horaire-create()  
hori=&h = malloc(sizeof(horaire));  
h->time-dep=htime-curr = 0, 0.  
return h.  
  
void horaire-destroy(horaire *h)  
    free(h)  
    return  
int horaire-get Time Dep (h*)  
int horaire-get Tim-Arr(h*)  
void horaire-set Time Dep(h*, int arr)  
..  
  
h_l* horaire-list:  
    h_l  
    h_l-> next  
    h_l-> prev  
  
Next. f
```

- 2) Check when $pdc[i]$ is free again.
- 3) Check if $pdc[i]$ next usage is sooner than arrival-time + charge time.
 false: check next pdc
 true: add travel; return.



charge time.

charge time.

$pdc[i]$:
 $DU = \text{next from } C_T$
 $FU = FU \text{ qui précède } DU.$

$$W_T = \max(C_T, FU) - C_T$$

$$P_C-T = DU - C_T$$

$$W_T \leq n \cdot W_T \text{ or } P_C-T \geq \text{charge}$$

```

void get_WT_PCT(h_l*, WT, C_T, chargeTime)
if length(h_l) == 0
WT = 0
for i=0 to length(h_l)-1, i++
    if (temp->H_avr) > C_T
        F_U = temp->prev->H_dep
        if F_U > DU
            F_U = C_T
        if DU - C_T > chargeTime && WT > max(C_T, DU) - C_T
            WT = max(C_T, DU) - C_T
    // optimisation possible over loop.

```

charge time and arrival time.
in pdc[i] is free after arrived to
pdc[i] next usage is sooner than arr.
check next pdc F_U = min
add travel return.

T $C_T + W - T$

$W - T > \max(C_T, DU) - C_T$

```

void h_l_insert(h_l*, h*)
    h_l*.current = h_l; bool beenInserted.
    for(i=0, i< h_l.length(h_l*), i++)
        if current->H_Depth < h->H_Depth & not('beenInserted')
            h_l_append(current, h)
            beenInserted = true
        if beenInserted
            current->index++
            current = current.next.

```

```
bool betterWaitTime( borne, waitingTimeAct, int *betterPdcIndex)
    int T = MAX_INT
    for (int i = 0; i < pole; i++)
        | get WT_PCT ( borne - h[i] ) * W - T, int *c - T, charge )
        if waitingTimeAct > W - T
            * (betterPdcIndex) = * (possiblePdcIndex)
    return waitingTimeAct > W - T
```

Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
<ul style="list-style-type: none"> • complet • motivant • envoyé à l'avance 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Introduction :	
<ul style="list-style-type: none"> • tous les objectifs rappelés • gestion du temps annoncée 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	~
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
<ul style="list-style-type: none"> • gestion du temps • sujets importants traités • pas de perte de temps 	~ <input checked="" type="checkbox"/> ~
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte rendu réunion n°6

9 Juillet 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 09/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Debug pathfinding	Léo	<input checked="" type="checkbox"/>
Implémentation des modifications de la partie 2	Maelan & Killian	<input checked="" type="checkbox"/>

Ordre du jour

1. Retour sur les modifications de la partie 2
2. Retour sur le pathfinding

Informations échangées

Retour sur les modifications de la partie 2

- La partie théorique est complète
- L'implémentation est quasiment terminée, toutes les fonctions auxiliaires ont été rédigées
- Seule l'ajout des variables temporelles dans la fonction `pathfinding` reste à faire
- Les tests et le debugging seront réalisé dans la foulée
- Les fonctions annexes et les structures sont implémentées
- Il subsiste quelques erreurs d'intégration des nouvelles fonctions

Retour sur le pathfinding

- Les trajets peuvent être encodés dans une string d'un format particulier afin d'être compatible avec google maps. Cela permet de vérifier facilement la cohérence des chemins, de plus cela sera utile dans l'optique d'une intégration dans une interface web
- 90% des tests de `pathfinding` passent avec succès avant l'ajout des fonctions d'horaires.

Décisions

- Un site internet en Flask sera réalisé pour fournir une interface utilisateur plus limpide et facile d'utilisation
- Les tests de **pathfinding** n'étant pas complètement validés, du travail supplémentaire ce soir et ce week-end est prévu pour terminer la validation de la partie 1

TODO List

Description	Responsable	Délai	Livrable	Validé par
Implémentation des modifications de la partie 2	Maelan & Killian	13/06/23 (soir)	Implémentation des modifications terminée	Tous
Implémentation des tests de la partie 2	Maelan	12/06/23 (soir)	Tests fonctionnels	Tous
Debug pathfinding avec partie 2	Léo	13/06/23 (soir)	Tests et fonctions sans bug	Tous

Date de la prochaine réunion 12/06/23

Documents additionnels

Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
• complet	<input checked="" type="checkbox"/>
• motivant	<input checked="" type="checkbox"/>
• envoyé à l'avance	<input type="checkbox"/>
Introduction :	
• tous les objectifs rappelés	<input checked="" type="checkbox"/>
• gestion du temps annoncée	<input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	<input checked="" type="checkbox"/>
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
• gestion du temps	<input checked="" type="checkbox"/>
• sujets importants traités	<input checked="" type="checkbox"/>
• pas de perte de temps	<input checked="" type="checkbox"/>
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte rendu réunion n°7

9 Juin 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Ce compte rendu est un condensé des 2 réunions réalisés ce jour :

- La réunion de 15h30 a permis de définir les objectifs de la journée et répartir les tâches
- Celle de 19h40 fait le bilan des tâches effectivement réalisées ce jour et préparer la journée de travail suivante

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 12/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Debug pathfinding	Léo	<input checked="" type="checkbox"/>
Implémentation des modifications de la partie 2	Maelan & Killian	<input checked="" type="checkbox"/>
Implémentation des tests de la partie 2	Maelan	<input checked="" type="checkbox"/>
Debug pathfinding avec partie 2	Léo	<input checked="" type="checkbox"/>

Ordre du jour

1. Debug partie 1
2. Debug et tests partie 2
3. Avancement interface web
4. Rapport et comptes rendus

Informations échangées

Debug partie 1

- 90% des tests passent toujours, même suite à l'implémentation des nouvelles fonctions pour la partie 2
- Il faut encore debug les derniers tests
- Après réflexion, l'insertion de la partie 2 dans la partie 1 est erronée, les fonctionnalités n'étaient pas correctement imbriquées dans le code

Debug et tests partie 2

- Les tests et le debug sont encore à faire, ils ne seront implémentable suite à l'ajout effectif de la partie 2 dans la partie 1

Avancement interface web

- L'UI se basera en partie sur l'interface web des autres applications de trajets déjà existantes (google maps, île-de-France mobilités...). L'affichage d'un trajet se fera donc de façon verticale



Rapport et comptes rendus

- La rédaction du rapport est entamée, les parties sur la complexité et l'interface web ne sont pas réalisés car le contenu actuel du programme ne le permet pas encore
- La moitié du rapport est déjà rédigée

Décisions

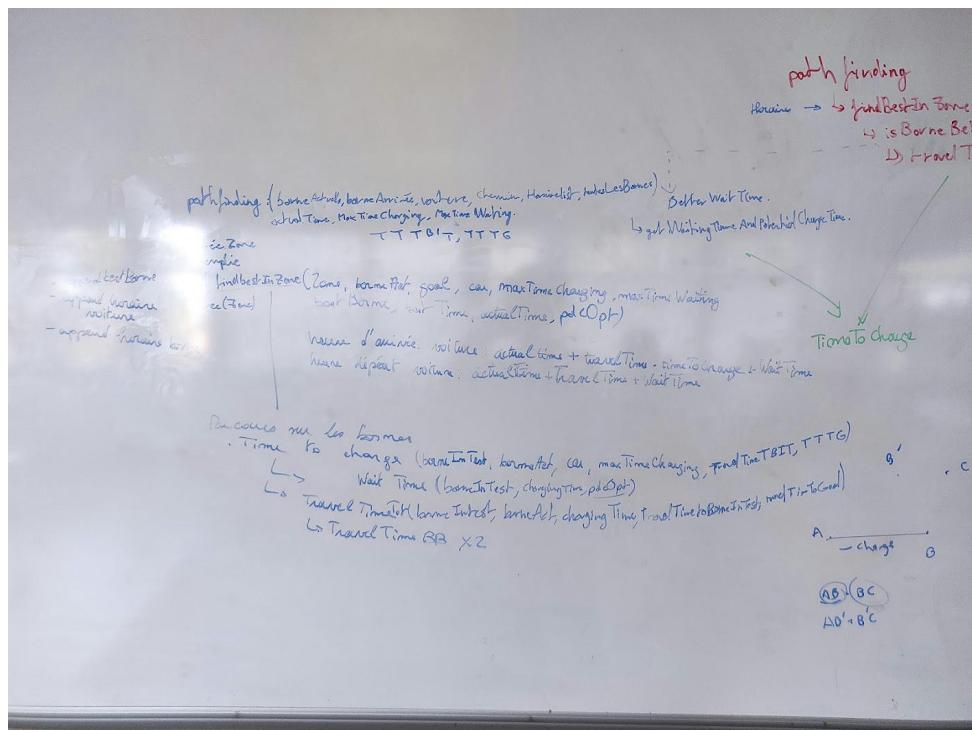
- L'ensemble des membres du projet réalisera du travail supplémentaire ce soir en distanciel en raison d'un retard au début de la session de travail du jour (discussion sur le jury et la soutenance avec M. Oster). Seront réalisé ce soir:
 - Léo & Maelan : suite du debug & implémentation partie 2
 - Killian : fin de la rédaction de la partie du rapport réalisable pour le moment
- Un retour sera obligatoire en début de séSSION prochaine.

TODO List

Description	Responsable	Délai	Livrable	Validé par
Implémentation partie 2 dans la partie 1	Maelan & Léo	13/06/23 (mardi)	Partie 2 correctement implémentée	Tous
Rapport (partie réalisable)	Killian	13/06/23 (mardi)	Rapport écrit	Tous

Date de la prochaine réunion 13/06/23

Documents additionnels



Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
<ul style="list-style-type: none"> • complet • motivant • envoyé à l'avance 	~ ~ <input checked="" type="checkbox"/>
Introduction :	
<ul style="list-style-type: none"> • tous les objectifs rappelés • gestion du temps annoncée 	<input checked="" type="checkbox"/> ~
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	~
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
<ul style="list-style-type: none"> • gestion du temps • sujets importants traités • pas de perte de temps 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte rendu réunion n°8

13 Juin 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Ce compte rendu est un condensé des 2 réunions réalisés ce jour :

- La réunion de 15h a permis de définir les objectifs de la journée et répartir les tâches
- Celle de 19h40 fait le bilan des tâches effectivement réalisées ce jour et préparer la journée de travail suivante

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 12/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Implémentation partie 2 dans la partie 1	Maelan & Léo	<input type="checkbox"/>
Rapport (partie réalisable)	Killian	<input checked="" type="checkbox"/>

Ordre du jour

1. Debug **pathfinding**
2. Rapport et comptes rendus
3. Avancement interface web

Informations échangées

Debug **pathfinding**

- Enormément de modifications de l'algorithme ont été réalisées. Beaucoup d'arguments passés en paramètres sont maintenant passé sous forme de pointeur, quelques redéfinitions de fonctions
- L'algorithme tourne sans renvoyer d'erreur. Néamoins il ne semble pas à l'heure actuelle donner un chemin avec les limitations des caractéristiques de la voiture choisie.
- Une erreur d'allocation mémoire subsiste durant la phase de test, les différents correctifs appliqués amènent tantôt à un **double-free** ou un **memory leak**.

Rapport et comptes rendus

- La majorité du document est rédigé, les parties sur l'algorithme ne le sont pas, en effet elle est encore sujet à modifications.

- A part la SWOT, tous les outils de gestion de projet ont été fait
- Killian a réalisé le diagramme de Gantt prévisionnel ainsi que la majorité du diagramme de Gantt réel

Décisions

- Une soirée code est planifiée pour Léo & Killian dans l'optique d'avoir pathfinding fonctionnel
- Jeudi de 16h à 20h : Tout est push sur le gitlab en l'état. On réalise un Beamer de présentation, un peu de code commenting and cleaning.

TODO List

Description	Responsable	Délai	Livrable	Validé par
Debug partie 2	Maelan & Léo & Killian	14/06/23 (mardi)	Partie 2 correctement implémentée	Tous
Interface C	Killian	14/06/23 (soir)	app.c finie	Tous
Fin rapport	Léo	13/06/23 (mardi)	Rapport écrit	Tous

Date de la prochaine réunion 14/06/23

Documents additionnels

```

horizon_chaining((actualTime, actualTime))
pathfinding (usedCar, actual, goal, path, horaire_chaining, listAllBornees, maxTimeWaiting, actualTime)
    horaire_chaining(usedCar)
    if distance(actual, goal) < wait_time_max :
        | horaire_chaining(goal)
        | open(horizon_start, horizon_actual, horaire_chaining) + travelTime(actual, goal)
    sinon :
        | horaire_list = bornee.list - create()
        creationZone (Zone, goal, actual, vautours, P, listAllBornees)
        findBestBorne (Zone, actual, goal, usedCar, best, bestTime,
                        actualTime, travelTimeToGoal, travelTimeToBorneInTest,
                        pdcOptimal, chargeTime)
        horaire_list.pop(horizon_start, horaire_createWithValues(actualTime + travelTimeToBorneInTest, actualTime + bestTime - travelTimeBestToGoal))
        bestActualTime = bestTime - travelTimeBestToGoal
        usedCar->AutonomyAct += chargeTime * bestPower / 60 * usedCarCapacity * usedCarsAutonomyMax - distal(actual, best)
        horaire_list.insert(-1, best)
        horaire_createWithValues(actualTime + travelTimeActualToBest + waitingTime,
                                travelTimeToBorneInTest, maxTimeWaiting, actualTime)
        return Pathfinding (usedCar, best, goal, path, horaire_chaining, listAllBornees, maxTimeWaiting, actualTime)
    | charging
| findBestInZone(...):
    b1Temp = Zone
    for i in range(b1Temp.length(Zone) - 1, -1, -1):
        bornee = Zone[i].getBornee(temp)
        updateBestBorne(best, bestTime, borneeInTest, actual, goal, bestPdc, chargeTime, waitingTime, actualTime, travelTimeToGoal)
    temp = temp - wait

```

```

updateBestBorne(...):
int travelTimeActualToBorneInTestTemp = travelTime(actual, borneeInTest)
int borneeInTestToGoalTemp = ... (..., goal)
int chargeTimeTemp = timeToCharge(borneeInTest, maxTimeCharging, usedCar, actual)
int bestPdcTemp = 0;
int waitingTimeTemp = *WaitingTime,
betterWaitTime(borneeInTest, waitingTimeTemp, bestPdcTemp, actualTime, chargeTimeTemp)
    > if (travelTimeActualToB + travelTimeBToGoal + chargeTime + waitingTime >
        "Temp + " Temp + " Temp + ")
        * TravelTimeActToB = > Temp
        * " BToGoal = " Temp
        * chargeTime = chargeTimeTemp
        * bestPdc = bestPdcTemp.

```

Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
<ul style="list-style-type: none"> • complet • motivant • envoyé à l'avance 	<input checked="" type="checkbox"/> ~ <input checked="" type="checkbox"/>
Introduction :	
<ul style="list-style-type: none"> • tous les objectifs rappelés • gestion du temps annoncée 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	~
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
<ul style="list-style-type: none"> • gestion du temps • sujets importants traités • pas de perte de temps 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte rendu réunion n°9

14 Juin 2023

Killian Thuillier

Léo Vesse

Maelan Tiger (à partir de 19h)

Ce compte rendu est un condensé des 2 réunions réalisés ce jour :

- La réunion de 15h a permis de définir les objectifs de la journée et répartir les tâches
- Celle de 19h40 fait le bilan des tâches effectivement réalisées ce jour et préparer la journée de travail suivante

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Léo et Killian	Date : 14/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Debug partie 2	Maelan & Léo & Killian	<input checked="" type="checkbox"/>
Interface C	Killian	<input type="checkbox"/>
Fin rapport	Léo	<input type="checkbox"/>
Implémentation partie 2 dans la partie 1	Maelan & Léo	<input checked="" type="checkbox"/>

Ordre du jour

1. Debug **pathfinding**
2. Rapport et comptes rendus
3. Avancement interface web

Informations échangées

Debug **pathfinding**

- Pathfinding marche correctement, tous les test sont validés.
- Une erreur d'allocation mémoire subsiste durant la phase de test, les différents correctifs appliqués amènent tantôt à un **double-free** ou un **memory leak**.
- Les test d'horaires doivent être implémentés.
- Une erreur subsiste

Rapport et comptes rendus

- A part les schémas et l'ajout des CR, le rapport est terminé

Avancement interface web

- Recherche sur comment lancer un executable C depuis Python avec les arguments nécessaires
- Utilisation de l'API Google Maps pour réaliser le rendu du chemin à parcourir
- L'architectures des différentes pages du site a été réalisée ainsi que sa structure
- Les modifications à réaliser sur le code C pour obtenir un fichier de données exploitable vers Flask sont définies

Décisions

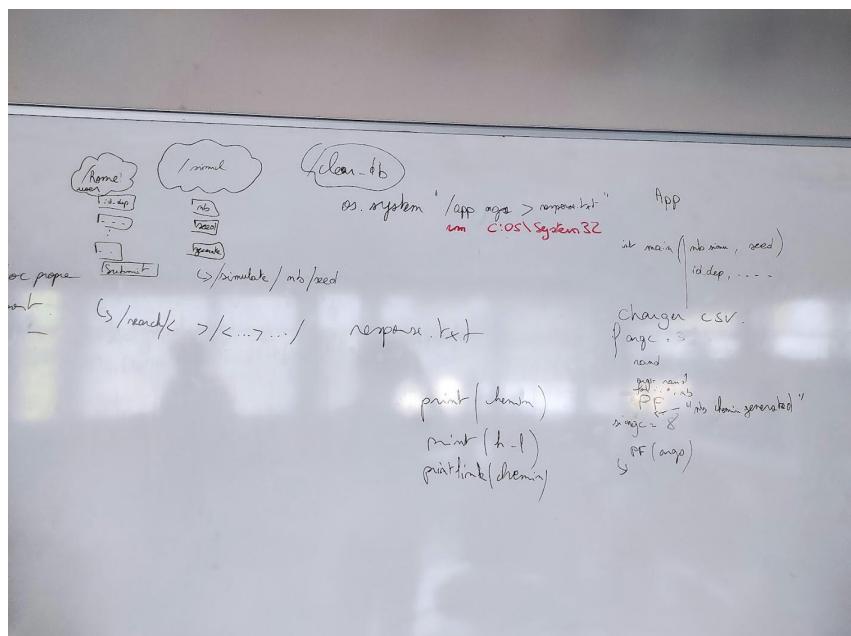
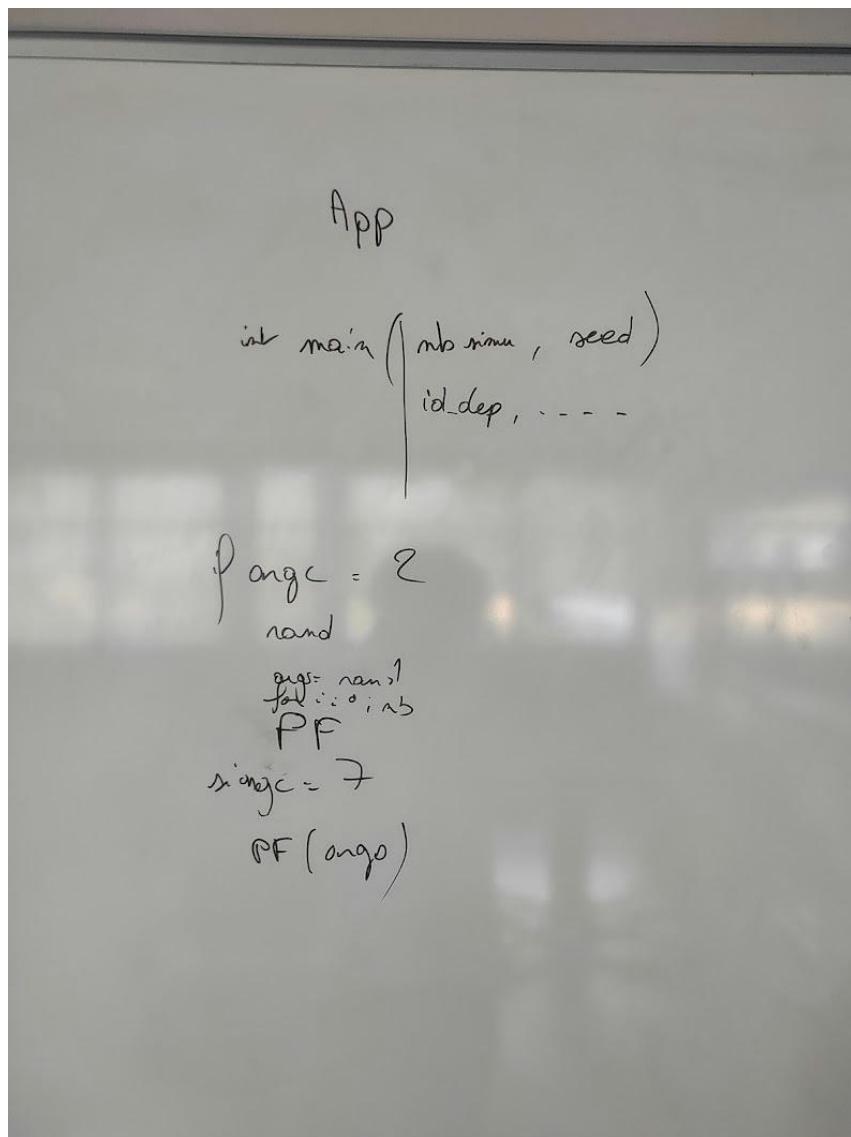
- Toutes les parties personnelles du rapport seront à réaliser avant demain midi en autonomie

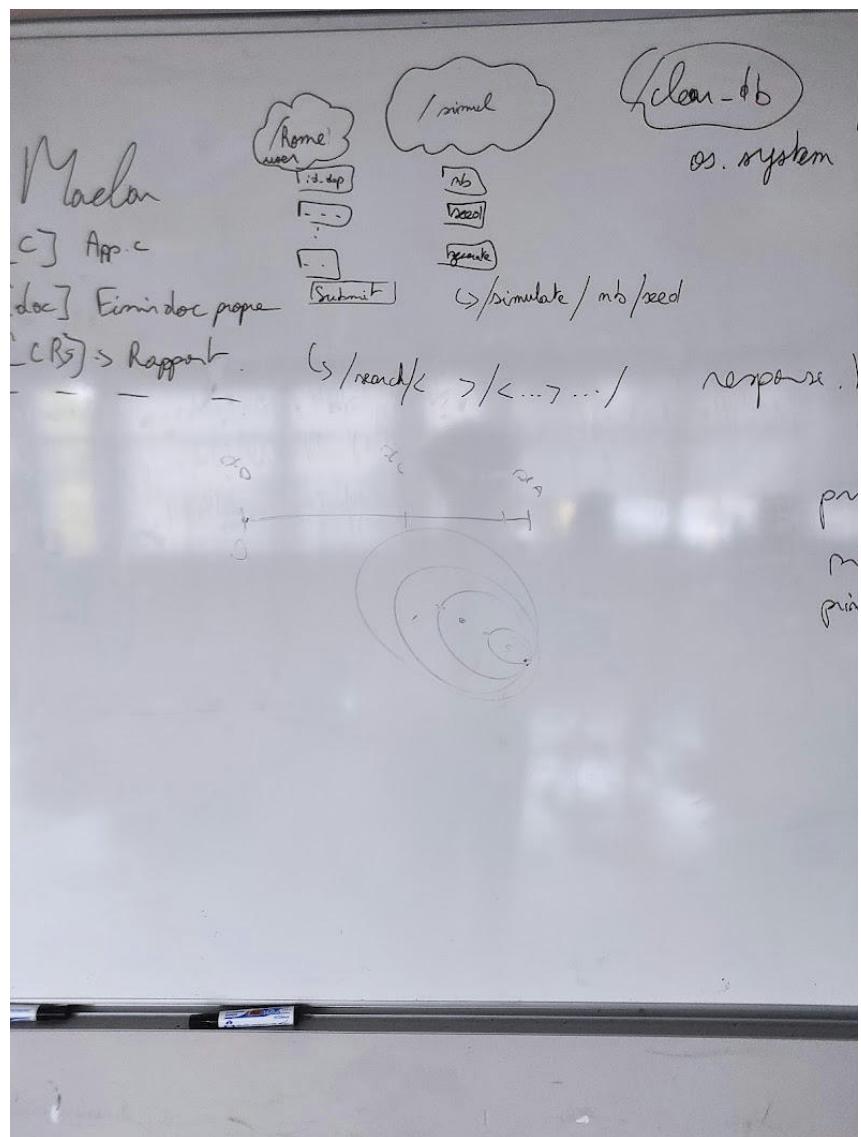
TODO List

Description	Responsable	Délai	Livrable	Validé par
Code Cleaning Printing Commentary	Killian	14/06/23 (soir)	Code concis et rigoureux	Tous
CR du rapport	Maelan	15/06/23 (matin)	CR inclus dans le rapport	Tous
app.c	Maelan	15/06/23 (midi)	app.c sans memory-leak	Tous
Documentation	Maelan	14/06/23 (soir)	Documentation complète	Tous

Date de la prochaine réunion 15/06/23

Documents additionnels





Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour :	
<ul style="list-style-type: none"> • complet • motivant • envoyé à l'avance 	<input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Introduction :	
<ul style="list-style-type: none"> • tous les objectifs rappelés • gestion du temps annoncée 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	<input type="checkbox"/>
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace :	
<ul style="list-style-type: none"> • gestion du temps • sujets importants traités • pas de perte de temps 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	<input checked="" type="checkbox"/>
Compte-rendu propre en 48h	<input checked="" type="checkbox"/>

Compte rendu réunion n°10

15 Juin 2023

Maelan Tiger

Killian Thuillier

Leo Vesse

Motif de réunion : Réunion d'avancement	Lieu : Salle de travail de TN
Présents : Tous	Date : 15/06/23

Suivi des objectifs précédents

Objectifs :	Responsable :	Validé
Code Cleaning Printing Commentary	Killian	<input checked="" type="checkbox"/>
CR du rapport	Maelan	<input type="checkbox"/>
app.c	Maelan	<input type="checkbox"/>
Documentation	Maelan	<input type="checkbox"/>

Ordre du jour

1. Site
2. app.c
3. Beamer

Informations échangées

Site

- Toutes les pages sont implémentées sans CSS pour l'instant
- La gestion de mémoire en écriture est terminée, mais pas de lecture des chemins précédents

app.c

- Encore et toujours du debug, réécriture de plusieurs fonctions dans `horaire_list.c` et traque de `direct leaks` dans plusieurs fichiers

Beamer

- Il semble que Typst puisse générer une présentation du même acabit que Beamer, on le préfèrera donc pour la réalisation des slides de présentation

Décisions

TODO List

Description	Responsable	Délai	Livrable	Validé par
Presentation Typst	Tous	24 Juin à 9h	pdf de présentation	Tous

Pas de prochaine réunion

Documents additionnels

Check-list efficacité réunion

Points	Etat
Pre-work fait	<input checked="" type="checkbox"/>
Ordre du jour : <ul style="list-style-type: none"> • complet • motivant • envoyé à l'avance 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Introduction : <ul style="list-style-type: none"> • tous les objectifs rappelés • gestion du temps annoncée 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Rebouclage avec le CR précédent	<input checked="" type="checkbox"/>
Ambiance sérieuse et constructive	~
Supports présentés de qualité	<input checked="" type="checkbox"/>
Animateur efficace : <ul style="list-style-type: none"> • gestion du temps • sujets importants traités • pas de perte de temps 	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> ~
Conclusion réalisée	<input checked="" type="checkbox"/>
Prochains objectifs définis et clairs	<input checked="" type="checkbox"/>
Date prochaine réunion fixée	\
Compte-rendu propre en 48h	<input type="checkbox"/>