

CHRONO PARK

Rapport de soutenance n°1

Charlotte BUAT
Anthony VERKINDERE
Nathan LABERNARDIERE



EPITA, Mai 2022.

Table des matières

1	Introduction	4
2	État de l’art	5
2.1	Technologies existantes	5
2.2	Notre solution	5
3	Découpage des tâches	6
3.1	Algorithme du plus court chemin	6
3.1.1	Ce qui était prévu	6
3.1.2	Ce qui a été fait	6
3.2	Options	8
3.2.1	Options principales	8
3.2.2	Options secondaires	8
3.3	Graphismes	9
3.3.1	Ce qui était prévu	9
3.3.2	Ce qui a été fait	10
3.4	Plan du parc	11
3.4.1	Ce qui était prévu	11
3.4.2	Ce qui a été fait	12
3.5	Déplacement des humains	13
3.5.1	Ce qui était prévu	13
3.5.2	Ce qui a été fait	14
3.6	Début de l’intelligence artificielle	14
3.7	Ajout de traits humains	16
3.7.1	Ce qui était prévu	16
3.7.2	Ce qui a été fait	16
3.8	UI avec options	16
3.8.1	Ce qui était prévu	16
3.8.2	Ce qui a été fait	17
3.9	Site Web	18
3.9.1	Ce qui était prévu	18
3.9.2	Ce qui a été fait	18
3.10	Répartition des tâches	20

3.11	Avancement	20
4	Ressenti des membres	21
4.1	Charlotte Buat	21
4.2	Nathan Labernardiere	21
4.3	Anthony Verkindere	22
5	Conclusion	23
6	Bibliographie	23

1 Introduction

Depuis toujours, un des plus gros défaut des parcs d'attractions sont les files d'attente. En effet, les files d'attente sont la hantise des visiteurs.

Prenons par exemple le célèbre parc Disneyland : un visiteur moyen passe 3 à 4 heures dans des files pour une journée de 8 à 9 heures soit plus d'un tiers de son temps. Il y a plusieurs facteurs à prendre en compte pour comprendre pourquoi le temps d'attente des files peut sembler interminable.

Ainsi, nous avons décidé de travailler sur un simulateur de trafic dans un parc d'attraction le plus complet et réaliste possible pour comprendre quels sont les facteurs qui influent sur les files d'attente et, par conséquent, voir comment les optimiser.

2 État de l’art

2.1 Technologies existantes

En ce qui concerne l’état de l’art en lien avec notre projet, il existe déjà plusieurs technologies aidant à l’optimisation de file d’attente.

Parmi celles-ci, nous pouvons citer toutes les applications directement créées par et pour certains parcs d’attractions tels que Disneyland Paris ou le parc Astérix en France mais aussi plein d’autres à travers le monde. Celles-ci permettent aux personnes dans le parc de savoir en temps réel quel serait le temps d’attente pour chaque attraction ou encore de localiser des attractions à travers le parcs. Mais aussi, depuis peu sur l’application Disneyland Paris, de réserver des « coupes files » pour certaines attraction ce qui permet de ne pas faire la queue du tout. Tout ceci dans le but de diminuer au maximum les temps d’attente de chaque visiteur.

Mais il existe aussi des sites Internet et applications indépendantes tels que **ParkTrips.fr**. Ceux-ci donnent des astuces afin d’éviter l’afflux touristique ou encore d’organiser son parcours au préalable. Tout ceci permettant aux visiteurs d’éviter les longues files d’attente.

2.2 Notre solution

Concernant notre projet, nous avons donc prévu de faire en sorte que les utilisateurs puissent optimiser leur visite de parc d’attraction en réduisant le temps passé dans les files d’attente.

Pour cela, nous allons recréer un parc avec un nombre défini d’attractions pour l’instant, tout en essayant plus tard de rendre le nombre d’attractions personnalisable. Puis, un algorithme permettra de définir un chemin optimal entre ces attractions tout en prenant en compte les goûts renseignés au préalable par l’utilisateur. Nous prendrons aussi en compte toute panne éventuelle d’une attraction ou encore la météo car la pluie diminuerait l’affluence générale du parc.

Tout ceci pour au final renvoyer le ou les chemins optimaux pour un utilisateur afin qu’il ait au final beaucoup moins d’heures d’attente.

Nous avons aussi décider d'implémenter des graphismes afin de laisser l'utilisateur voir ce qu'il se passe exactement avec les IA, les humains présents dans le parcs, se déplaçant entre les attractions.

3 Découpage des tâches

3.1 Algorithme du plus court chemin

3.1.1 Ce qui était prévu

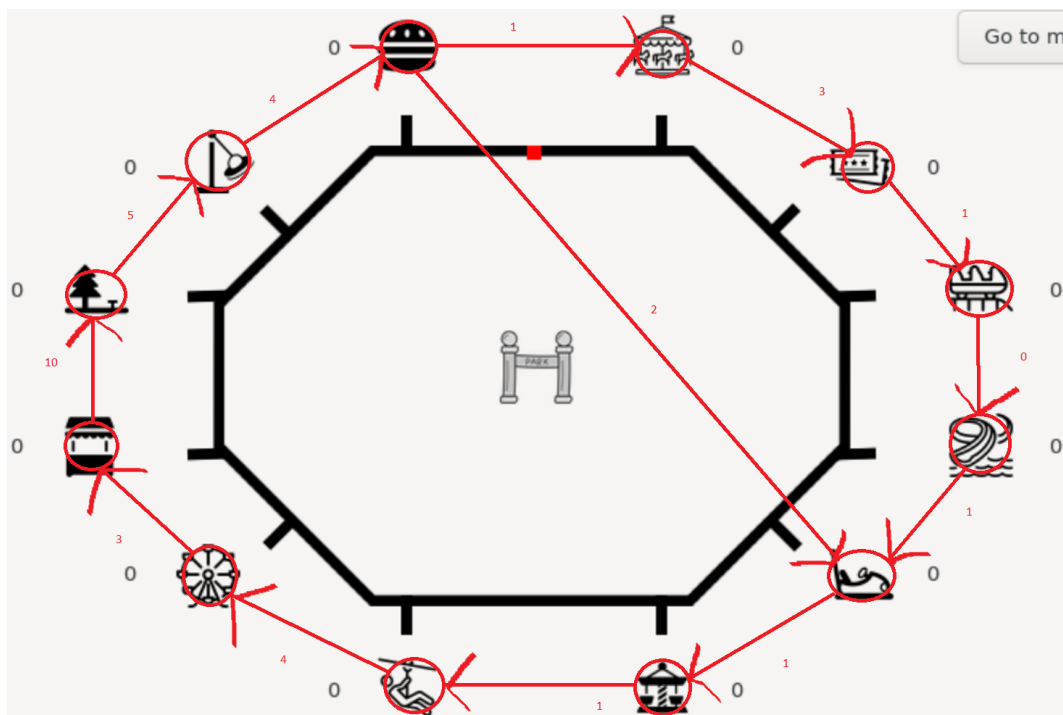
L'objectif du logiciel est de proposer un enchaînement d'attractions en fonction de l'affluence qu'il y a sur chaque attraction. Dans l'objectif que le parc soit terminé avec un temps d'attente dans les files le plus faible possible.

C'est pourquoi nous avons besoin de faire un algorithme du plus court chemin, qui peut prendre en compte le fait qu'une file d'attente pour une attraction soit plus ou moins pleine.

Nous voulons permettre à nos utilisateurs de parcourir un parc réaliste, il y aura donc plusieurs chemins possibles pour atteindre les différentes attractions, et des circuits possibles. Ainsi, nous ne pouvons pas utiliser l'algorithme de Dijkstra qui n'accepte pas les circuits. Étant donné que nous voulons proposer des parcours permettant à l'utilisateur de parcourir les attractions de son choix, nous allons plutôt nous tourner vers des algorithmes tels celui de Floyd ou de Johnson.

3.1.2 Ce qui a été fait

Pour commencer nous avons du développer une classe "Graph". Comme nous avons pu voir en cours d'algorithme mais ici en C dans le but de représenter un parc d'attraction sous forme de graphe. Dans notre graphe, les sommets correspondent à une attraction et chaque chemin entre deux attractions a un poids qui correspond à plusieurs critères comme la distance et le nombre de personne dans l'attraction. De plus, nous avons implémenté une classe liste pour faciliter l'implémentation de notre algorithme. Voici comment l'on peut visualiser notre graphe, ici c'est un exemple les liaisons entre chaque sommet et leur coût peuvent changer et nous pouvons avoir plus ou moins de sommets.



Nous avons développé deux algorithmes de plus court chemin, Floyd et Dijkstra. Floyd était celui que nous avions prévu de développer et c'est celui que nous utilisons pour le moment dans notre logiciel. Mais par la suite nous souhaitons pouvoir utiliser les différents algorithmes qui vont nous permettre de voir une différence de chemin. Bien évidemment, nous allons adapter nos graphes pour que les deux algorithmes soient compatibles avec celui-ci. Nous pouvons donc dire que notre algorithme est terminé. La seule chose qui pourra potentiellement être ajouté pour la soutenance finale est l'algorithme de Johnson pour pouvoir comparer les trois algorithmes dans notre logiciel. Ceci sera fait seulement si nous avons terminé tout ce que nous avions prévu de faire.

3.2 Options

3.2.1 Options principales

Ce qui était prévu :

Nous allons implémenter de nombreux paramètres afin d'améliorer le réalisme du simulateur. La première option essentielle est le temps d'attente dans les files d'attente en fonction du nombre de visiteurs présents dans le parc. Idéalement, nous allons essayer de donner la possibilité à l'utilisateur de faire varier le nombre de personnes présentes.

Ce qui a été fait :

Pour la première soutenance, nous avons implémenté la variation du nombre de personnes présentes dans le parc. De plus, nous avons la possibilité de positionner ces personnes aléatoirement sur chaque attraction du parc tout en tenant compte certains critères précis expliqués plus tard. Cependant, nous n'avons pas encore implémenté un code nous permettant de récupérer un temps d'attente en file. Cependant, il y en a bien un "caché". C'est à dire quand nous arrivons à une attraction nous ne pouvons pas directement entrer dans celle ci sans attendre que toutes les personnes (l'IA) soient sorties de l'attraction.

Ce qui est prévu :

Pour la soutenance finale. Nous allons essayer de mettre sur l'interface graphique plus d'informations à propos du temps d'attente quand nous arrivons à une attraction tout en essayant de le complexifier en le faisant varier en fonction de critères spéciaux à chaque attraction.

3.2.2 Options secondaires

Ce qui était prévu :

Nous allons donc donner la possibilité à l'utilisateur de changer le nombre de visiteurs manuellement et automatiquement. Implémenter le fait qu'une attraction puisse être en maintenance ou tout simplement hors service (le but est que cela se fasse aléatoirement). De plus nous allons ajouter un système de « météo » qui va nous permettre d'influencer le nombre de personnes présentes dans le parc. Pour optimiser encore notre algorithme

et la fiabilité de notre logiciel, nous allons prendre en compte la densité de personnes présentes sur un trajet entre deux attractions pour avoir le parcours le plus optimal dans le but de terminer le parc tout en ayant un temps d'attente dans les files le plus faible possible. Pour finir, nous essaierons de proposer plusieurs parcours différents en fonction des préférences de l'utilisateur.

Ce qui a été fait :

Comme dit précédemment, le nombre de visiteurs présents dans le parc au démarrage peut être saisi manuellement. Pour ce qui est de la météo, l'interface est déjà prête à l'avoir cependant cela n'a pas encore été implémenté. Notre algorithme a la possibilité de donner toutes les distances qu'il peut y avoir en partant de chaque attraction.

Ce qui est prévu :

Nous allons donc implémenter le système de météo pour la soutenance finale. De plus le système d'attraction hors service / maintenance sera mis en place pour ajouter du réalisme à notre logiciel. Nous allons essayer de prendre en compte le nombre de personnes présentes sur le chemin pour améliorer la précision de notre algorithme de plus court chemin mais c'est une partie assez longue et difficile donc nous allons privilégier les autres fonctionnalités avant celle-ci.

3.3 Graphismes

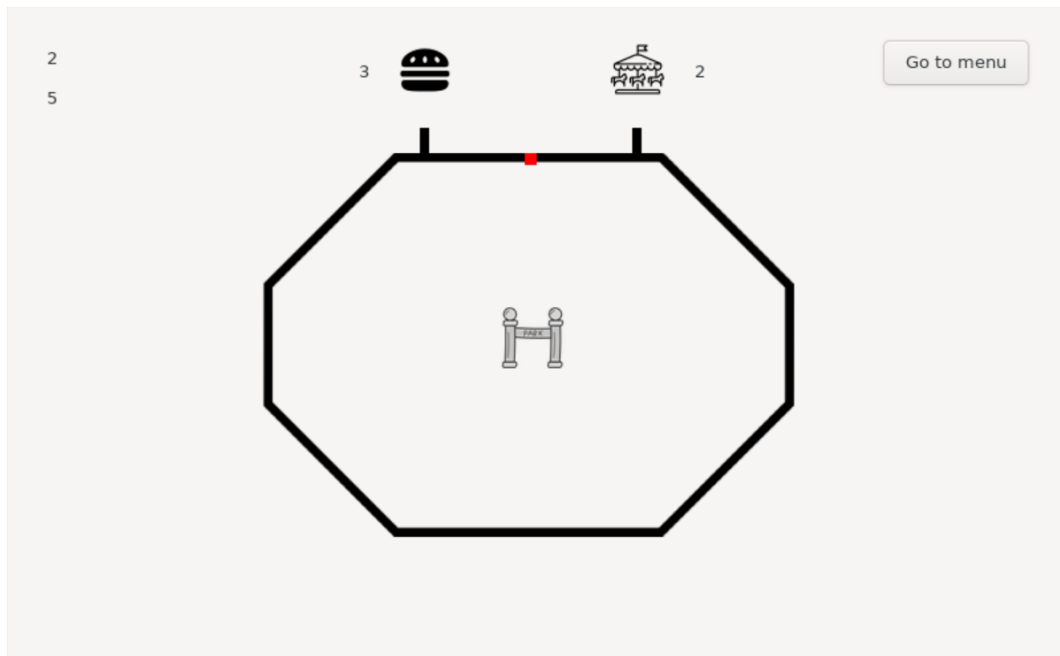
3.3.1 Ce qui était prévu

Notre projet ayant un sujet portant sur l'algorithmique, il était évident pour nous que notre objectif premier n'était pas de faire une interface avec des graphismes réalistes. Néanmoins, nous comptons faire une interface la plus soignée possible avec des dessins simples. Cela étant mieux que de simples pavés de chiffres et résultats illisibles. Une interface avec quelques graphismes permet de mieux voir et comprendre ce qui se passe derrière notre code et permet également de pouvoir expliquer plus simplement comment notre projet fonctionne.

3.3.2 Ce qui a été fait

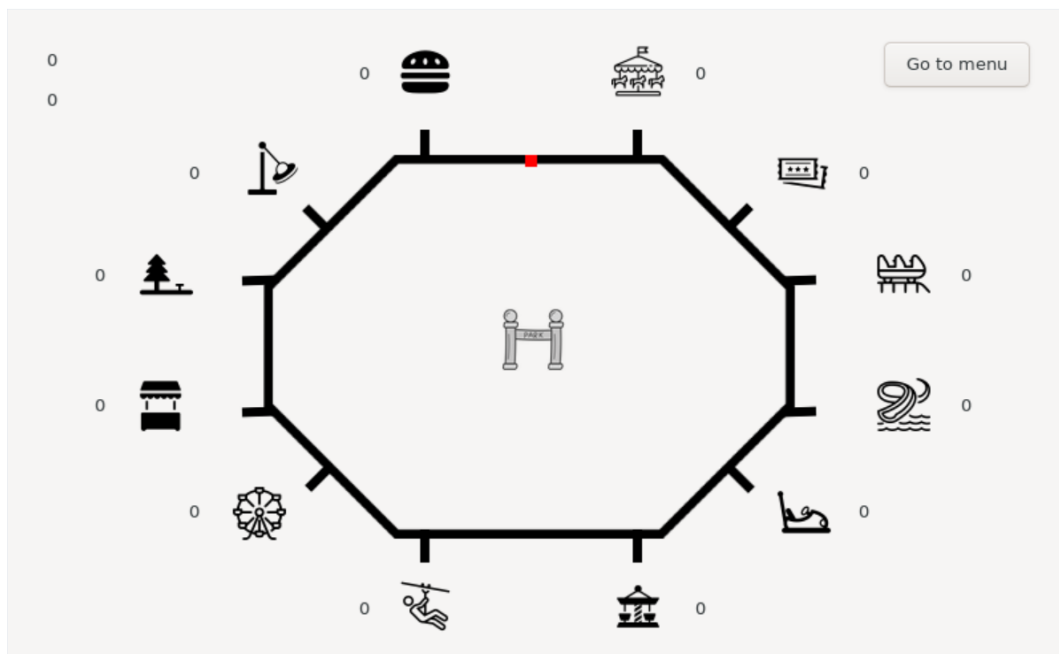
Ainsi, pour cette première soutenance nous avons utilisé l'application "Glade" pour concevoir les interfaces de notre projet. Nous allons dans cette partie nous intéresser à l'interface qui s'affiche une fois que l'on commence la simulation.

Voici une possibilité d'affichage ci-dessous :



Dans cet exemple, il y a le nombre minimal d'attractions qui est de 2. Pour réaliser cela, nous avons pris des illustrations sur Internet et dessinés une route faisant le tour du parc.

Voici un exemple d’affichage avec cette fois-ci 12 attractions :



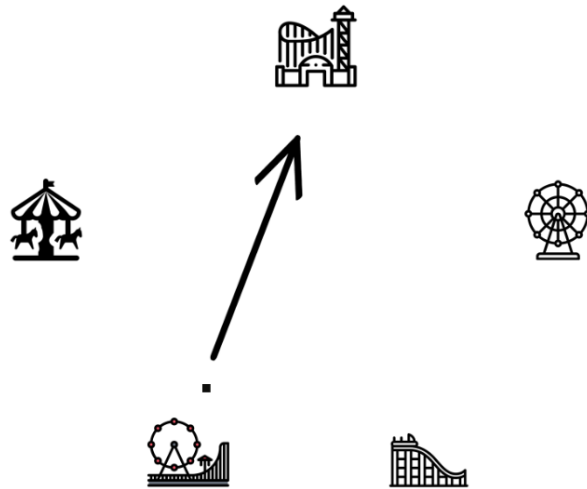
Sur l’exemple ci-dessus, il y a cette fois le nombre maximal d’attractions qui est 12. Nous sommes plutôt satisfaits du rendu qui est obtenu grâce aux petites illustrations. Cela rend le tout plus attractif.

3.4 Plan du parc

3.4.1 Ce qui était prévu

Nous avions prévu que pour le plan du parc, le nombre d’attractions soit défini pour que cela soit plus simple à réaliser. Nous avons pensé également qu’il serait pratique de faire en sorte que les attractions soient arrangées de manière circulaire afin de faciliter la représentation des déplacements humains.

Voici un schéma du parc que nous avons imaginé :



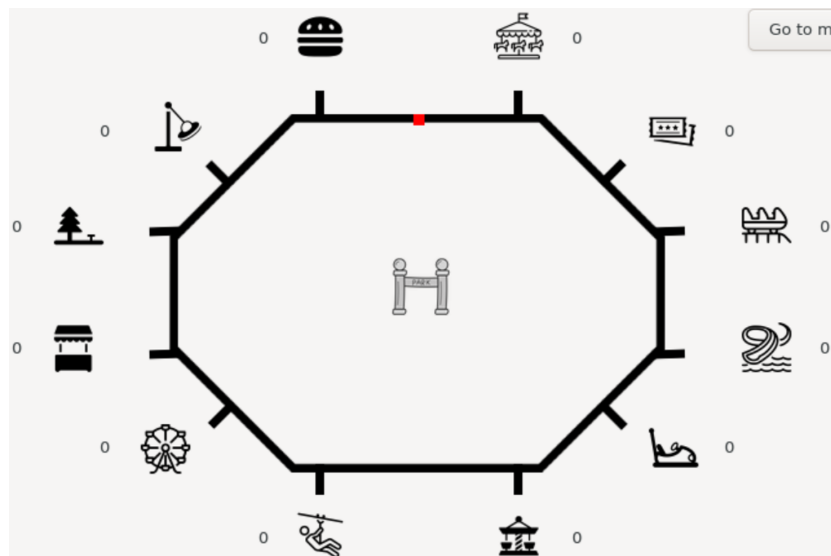
Sur le schéma ci-dessus, il y a 5 attractions et on a un humain, représenté par le point noir, qui s'apprête à se déplacer en ligne droite vers une nouvelle attraction.

3.4.2 Ce qui a été fait

Nous avons réalisé un parc d'attraction en forme d'octogone qui varie entre 2 et 12 attractions et ces attractions sont reliées par une seule route.

Afin de représenter le parc, nous avons créé une structure comprenant une liste des attractions et le nombre de gens présents dans le parc. Chaque attraction est aussi représentée par une structure qui comprend elle aussi le nombre de gens mais cette fois propre à l'attraction, mais aussi un nombre représentant à quel point chaque utilisateur aime l'attraction que l'on appelle : *likeness* et enfin le nombre de gens qui peuvent rentrer dans l'attraction par tour.

Voici le plan ci-dessous :



3.5 Déplacement des humains

3.5.1 Ce qui était prévu

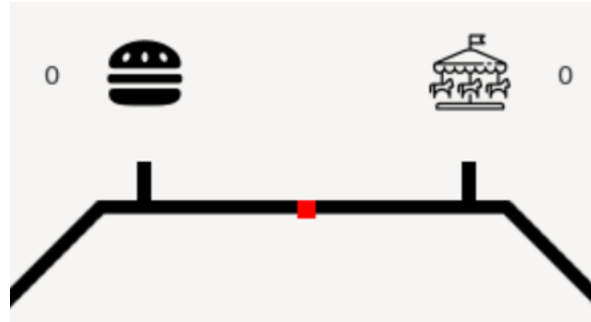
Pour ce qui est des déplacements, nous n'avions pas prévu de modéliser des déplacements complexes car cela relève plus du détail pour l'affichage que de l'algorithmique. Cependant, nous voulions quand même représenter les humains par des points et les faire se déplacer en ligne droite, d'attraction en attraction. Cependant, nous avons abandonné cette idée car cela ferait beaucoup trop de points affichés à l'écran et rendrait le tout illisible et incompréhensible. De plus, comme nous avions prévu de ne pas non plus représenter graphiquement les files d'attente, nous voulions ajouter des jauges allant du vert (file vide ou presque) au rouge (beaucoup d'attente) permettant d'avoir une idée du nombre de personnes qu'il y a.

3.5.2 Ce qui a été fait

Comme prévu, nous n'avons donc pas modélisé chaque personne une par une mais nous les affichons uniquement dans les attractions (le nombre qui est à côté de l'icône de l'attraction).



Cependant nous avons créé un petit point rouge que nous avons la possibilité de contrôler pour se déplacer dans le parc.



Ce point rouge sera finalisé pour la soutenance finale, il n'est donc toujours pas entièrement fonctionnel. Nous pouvons cliquer sur les icônes des attractions et à l'avenir cela sera entièrement fonctionnel pour que le carré rouge se déplace vers l'attraction et s'arrête et ainsi de suite. Pour le moment le carré rouge se déplace uniquement dans le parc sans se stopper comme souhaité.

3.6 Début de l'intelligence artificielle

L'intelligence artificielle contrôle la population constituée du nombre de gens entré par le joueur initialement. Chaque personne de cette population va, en entrant dans le parc, choisir une attraction vers laquelle aller. Ce choix se fait aléatoirement. En effet, chaque attraction a un paramètre *likeness* qui représente à quel point les gens aiment l'attraction permettant grâce à des probabilités de répartir la population dans le parc.

Une fois la population répartie, les humains doivent se déplacer, comme dans un vrai parc. Pour ce faire, toutes les 2 secondes, un nombre défini de personnes par attraction va sortir de cette attraction et aller se mettre dans la file d'une autre attraction encore une fois cela étant choisi grâce au paramètre *likeness* des attractions.

Durant le choix des attractions, il y a aussi une probabilité que la personne sorte du parc, en allant dans "l'attraction" sortie. En effet, celle-ci a aussi un paramètre *likeness* qui, lui, démarre à 0 et augmente au fur et à mesure de la journée. Ce faisant, de plus en plus de personnes sortent du parc. Nous avons aussi fait en sorte que le parc soit fermé à la fin de la journée. C'est-à-dire qu'au bout d'un certain nombre de tours de boucle, tous les visiteurs sortent du parc.

Pour la prochaine et dernière soutenance, nous avons aussi prévu de rajouter le fait que les personnes ne soient plus seulement influencées par leur attirance pour certaines attractions mais aussi par le nombre de gens qu'il y a dans la queue.

Ceci permettrait de répartir plus naturellement, comme on pourrait le voir dans un vrai parc, les visiteurs.

Nous allons aussi devoir implémenter le fait que les visiteurs réagissent à la pluie, en sortant du parc par exemple.

Nous pourrions également rajouter certains visiteurs qui viennent en groupe, avec leur famille ou leurs amis, et qui donc resteraient en groupe. Ou encore le fait qu'ils fassent des pauses et ne soient pas tous toujours dans une attractions.

Enfin nous pourrions aussi rajouter le fait que tous les visiteurs ne rentrent pas en même temps dans le parc, mais plutôt les faire rentrer petit à petit.

3.7 Ajout de traits humains

3.7.1 Ce qui était prévu

Un facteur que nous trouvions important à prendre en compte était les traits humains. En effet, pour rendre le tout vraiment réaliste il est intéressant d'implémenter des traits humains comme la fatigue après une attraction ou encore la faim. Ces implémentations sont importantes pour pouvoir étudier une simulation la plus réelle possible même si ces implémentations passent après les autres car le plus important pour nous reste en priorité de réussir au bon fonctionnement du programme.

3.7.2 Ce qui a été fait

Pour l'instant nous nous sommes majoritairement focalisés sur le reste donc cette partie n'a pas beaucoup évolué. Cependant, nous avons quand même un système de préférence des attractions où les personnes auront plus tendance à choisir une attraction qui est appréciée par les gens qu'une autre.

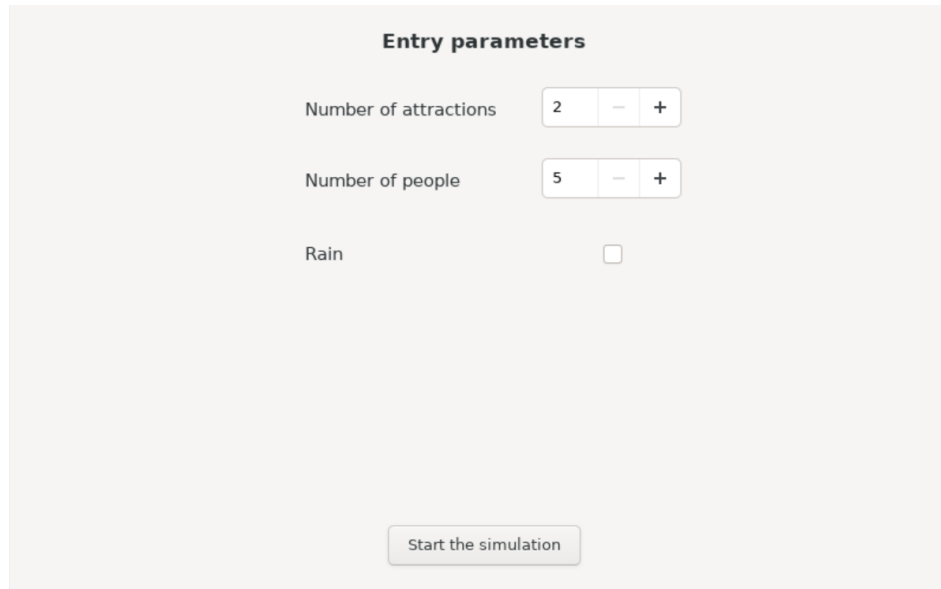
3.8 UI avec options

3.8.1 Ce qui était prévu

Pour rendre le logiciel plus simple d'utilisation, nous voulions faire une interface graphique, dans une esthétique simple et sobre qui donnera accès aux paramètres. Cela est beaucoup plus simple que de rentrer les paramètres via le terminal quand on lance le programme. L'interface permet de rajouter des préférences sur certaines attractions et la vitesse des humains contrôlée par l'intelligence artificielle (pour ajouter la possibilité de « courir »). L'ensemble rendra le logiciel plus modulable et permettra d'observer une grande diversité de situations.

3.8.2 Ce qui a été fait

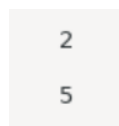
Voici l'interface qui a été réalisée pour l'instant :



The screenshot shows a web interface titled "Entry parameters". It contains three input fields: "Number of attractions" with a value of 2, "Number of people" with a value of 5, and "Rain" with an unchecked checkbox. Each of the first two fields has minus and plus buttons for adjustment. At the bottom, there is a button labeled "Start the simulation".

On voit donc que pour le moment seules 3 options sont possibles : le nombre d'attractions, le nombre de personnes ainsi que le choix de la pluie ou non. Il est évidemment prévu de rajouter d'autres options et de les rendre fonctionnelles pour la soutenance finale. Une fois que l'utilisateur a fini de choisir ses réglages, il suffit de cliquer sur le bouton "Start the simulation" et une fenêtre comme celles montrées dans la partie "3.3 Graphismes" apparaîtra.

D'ailleurs dans la seconde fenêtre qui se lance lors du démarrage de la simulation il y a des éléments en haut à gauche de la fenêtre qui sont importants dans le déroulement de la simulation.



Les 2 nombres ci-dessus représentent le temps écoulé (en haut) et le nombre de personnes restantes (en bas).

3.9 Site Web

3.9.1 Ce qui était prévu

Nous souhaitions créer un site Web pour représenter, à la fois, notre avancement tout au long de ce projet, mais aussi faire une page de présentation pour notre simulateur de trafic.

Cependant, le projet restant centré sur le côté algorithmique du simulateur, en C, nous ne souhaitions pas nous attarder de trop sur cet aspect de notre projet.

3.9.2 Ce qui a été fait

Concernant le site, nous avons donc décidé de le faire à l'aide du créateur de site Internet *Wix*.

Sur ce site, nous pouvons retrouver trois pages différentes :

- La page d'accueil, où nous présentons rapidement le projet et les membres du groupe.
- La page des téléchargements, où l'on peut télécharger le projet ainsi que les rapports de soutenance et le cahier des charges.
- La page à propos, permet quant à elle, de donner plus d'informations sur le projet.

Voici le lien de notre site :
<https://nathanlrbn.wixsite.com/projet-s4>
ainsi que des captures d'écran prise à partir de celui-ci :



Présentation

Nous sommes un groupe de 3 étudiants d'EPITA composé de **BUAT Charlotte**, **LABERNARDIERE Nathan** et **VERKINDERÉ Anthony**. Ce projet a été conçu pour être réalisé en équipe. L'objectif est de réaliser un programme de simulation de trafic dans un parc d'attraction en optimisant le plus possible le temps d'attente aux attractions.

Ce projet dans lequel l'algorithmique a une part très importante nous a grandement permis de nous améliorer dans ce domaine. Pour plus de détails, vous pouvez retrouver notre travail dans la section [Téléchargements](#) où vous pourrez télécharger le projet complet avec le cahier des charges et les rapports de soutenance et tester le logiciel qui a été fait.



3.10 Répartition des tâches

Voici la répartition des tâches ci-dessous. Celle-ci n'a pas changée depuis le cahier des charges.

	Anthony	Charlotte	Nathan
Algorithmes de déplacement			
Création du parc			
Options			
Interface et graphismes			
Déplacement des Humains			
IA			
Ajout de traits humains			
Site web			



Responsable



Suppléant

3.11 Avancement

Voici ci-dessous le tableau d'avancement qui avait été fait lors du cahier des charges. Nous trouvons que dans l'ensemble nous avons plutôt bien respectés chaque pourcentage.

	Soutenance n°1	Soutenance n°2
Algorithmes de déplacement	100%	100%
Création du parc	60%	100%
Options	33%	100%
Interface et graphismes	50%	100%
Déplacement des Humains	50%	100%
IA	50%	100%
Ajout de traits humains	25%	100%

4 Ressenti des membres

4.1 Charlotte Buat

Personnellement, ce projet me faisais au début un peu peur à cause des consignes très larges par rapport au sujet et ne connaissant pas les membres de mon groupe. Au final, tous mes problèmes ont été résolus rapidement. En effet, nous avons réussi à trouver un sujet et en plus mon groupe est sympathique. De plus, nous avons aussi réussi à nous organiser et à respecter nos dates limites.

Par ailleurs, étant déjà habituée au langage de programmation C grâce au S3, la mise en place du projet s'est faite assez facilement et le travail a pu commencer plus vite que lors du S3 pour le projet OCR.

Donc le projet de S4 est pour l'instant une bonne expérience pour moi et m'apprends à travailler avec des personnes que je ne connaissais pas du tout et cela pourrait m'arriver plus tard dans le milieu du travail.

4.2 Nathan Labernardiere

Les consignes étant plutôt vagues, j'appréhendais un peu ce projet au départ. Cependant, je me suis rapidement documenté sur Internet et j'ai réussi à avancer plus facilement que je ne le pensais. Je pense que la bonne cohésion qu'il y a dans notre groupe a permis à chacun d'entre nous de travailler efficacement et réussir ce que nous entreprenions.

De plus, je trouve cela vraiment intéressant de faire pour la première fois un projet où l'on maîtrise déjà le langage contrairement en SUP où l'on découvrait le C en même temps et en S3 où l'on découvrait le C. Ce projet me permet donc d'approfondir mes connaissances du langage C.

Pour finir, je suis plutôt confiant pour la soutenance finale car je pense que nous allons réussir à développer ce qui était prévu à l'origine.

4.3 Anthony Verkindere

À première vue, le projet faisait assez peur car nous n'avions pas vraiment d'idée de ce que nous pouvions faire. Finalement, nous avons réussi à trouver une bonne idée qui validait les consignes du projet de S4. Au début du projet, j'ai eu un peu peur que le code et les algorithmes attendus soient difficiles à implémenter. Mais avec les bases du semestre 3 et les compléments du semestre 4 à notre grand bonheur cela a été plus facile que prévu. De plus, nous avons été plutôt bien organisés à trois et avons quasiment réussi à travailler sur le projet chaque semaine tous ensemble. Ce qui nous a fait avancer petit à petit sans avoir de surcharge de travail avec la soutenance. Au final, même en distanciel il y a une très bonne entente dans le groupe même si réussir à se mettre au travail dans les conditions actuelles reste assez difficile.

5 Conclusion

Enfin, nous avons terminé la première étape de la conception de ce projet d'algorithmique. Ce fut éprouvant mais cela nous a permis d'en apprendre beaucoup et que cela est utile. Pour ce qui est du projet en lui-même, ce dernier peut s'apparenter à celui que nous avons pu réaliser au cours du S3. Cependant, le fait de cette fois pouvoir travailler sur un sujet qui nous est libre rappelle la liberté du projet de S1 / S2 qui est vraiment pratique pour développer quelque chose qui nous tient à cœur. Concernant la suite, nous aimerions réussir à développer tout ce que nous avons prévu pour pouvoir étudier le déroulement de la simulation si les personnes allaient n'importe où ainsi que le déroulement lorsque les déplacements sont optimisés. Ainsi, il nous reste évidemment encore beaucoup de chemin à faire mais nous sommes optimistes au vu de ce que nous avons réussi pour cette soutenance et nous savons que nous allons réussir à aller jusqu'au bout de ce que nous avons prévu.

6 Bibliographie

- * "Classe éco été: comment réduire les files d'attente au parc d'attraction?"
Franceinfo 2013, Alexandre DELAIGUE
- * "Parcs d'attractions et files d'attente" Tejix, Henry CORRADO
- * Floyd Algorithm
- * Dijkstra Algorithm
- * Floyd Algorithm