

# CHRONO PARK

## Cahier des charges

Charlotte BUAT  
Anthony VERKINDERE  
Nathan LABERNARDIERE



EPITA Paris, Mars 2022.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>État de l’art</b>	<b>4</b>
2.1	Technologies existantes . . . . .	4
2.2	Notre solution . . . . .	4
<b>3</b>	<b>Découpage des tâches</b>	<b>5</b>
3.1	Algorithme du plus court chemin . . . . .	5
3.2	Options . . . . .	5
3.2.1	Options principales . . . . .	5
3.2.2	Options secondaires . . . . .	5
3.3	Graphismes . . . . .	6
3.4	Plan du parc . . . . .	6
3.5	Déplacement des humains . . . . .	7
3.6	Début de l’intelligence artificielle . . . . .	7
3.7	Ajout de traits humains . . . . .	7
3.8	UI avec options . . . . .	8
3.9	Répartition des tâches . . . . .	8
3.10	Avancement . . . . .	8
<b>4</b>	<b>Conclusion</b>	<b>9</b>
<b>5</b>	<b>Bibliographie</b>	<b>9</b>

## 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 celle-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 visiteurs.

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 parcs 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**

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.

#### **3.2 Options**

##### **3.2.1 Options principales**

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 fonctions du nombre de visiteurs présent dans le parc. Idéalement, nous allons essayer de donner la possibilité à l’utilisateur de faire varier le nombre de personnes présentes.

##### **3.2.2 Options secondaires**

Nous allons donc donner la possibilité à l’utilisateur de changer le nombre de visiteur 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 personne présente 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.

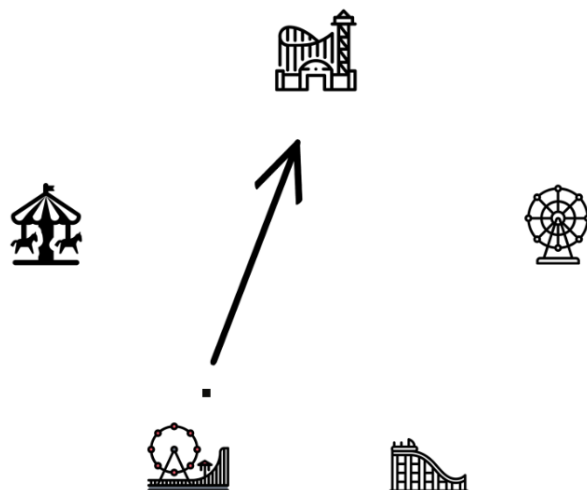
### 3.3 Graphismes

Notre projet ayant un sujet portant sur l'algorithmique, il est évident que notre objectif premier n'est pas de faire une interface avec des graphismes réalistes. Néanmoins, nous avons quand même prévu de faire une interface aux graphismes simplistes. Nous y afficherons également tous les chiffres et statistiques importants. Cela nous permettra de mieux voir et comprendre ce qui se passe derrière notre code et également de pouvoir expliquer plus simplement comment notre projet fonctionne.

### 3.4 Plan du parc

Le plan du parc se fera d'abord avec un nombre défini d'attractions pour que se soit plus simple à réaliser. Puis, nous essayerons d'implémenter une fonction telle que l'utilisateur pourra choisir lui-même combien d'attractions il souhaite avoir. Les attractions seront aussi arrangées de manière circulaire afin de faciliter la représentation des déplacements humains.

Voici un exemple de schéma pour visualiser à peu près comment sera organisé le parc :



Dans l'exemple ci-dessus, on a donc 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.

### **3.5 Déplacement des humains**

Pour ce qui des déplacements, nous n'avons pas prévu de modéliser des déplacements complexes car cela relève plus du détail pour l'affichage que de l'algorithmique. Cependant, nous modéliserons quand même les humains par des points et leurs déplacements ne se feront qu'en ligne droite, d'attraction en attraction. Ainsi, comme les files d'attente ne seront pas représentées graphiquement, nous ajouterons des jauges allant du vert (file vide ou presque) au rouge (beaucoup d'attente) permettant d'avoir une idée du nombre de personne qu'il y a. En plus de cela, il sera possible de suivre des statistiques précises telles que le nombre précis actuel de personnes dans une file ou autre.

### **3.6 Début de l'intelligence artificielle**

Notre intelligence artificielle aurait pour but de créer un mouvement des humains perpétuel entre les attractions pour rendre le logiciel plus réaliste. De plus cela va obliger notre algorithme à s'adapter à des circonstances non prévisibles et de recalculer les parcours en permanence.

### **3.7 Ajout de traits humains**

Un facteur important à prendre en compte dans nos algorithmes va être 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 qu'on ait vraiment des humains et non un simple point qui se déplace et également car les déplacements liés à ces traits ne seront pas négligeable pour l'optimisation recherchée.

### 3.8 UI avec options

Pour rendre le logiciel plus simple d'utilisation, une interface graphique sera mise en place, dans une esthétique simple et sobre qui donnera accès aux paramètres. Celui-ci permettra 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 infinité de situations.

### 3.9 Répartition des tâches

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



Responsable



Suppléant

### 3.10 Avancement

	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	25%	100%
IA	25%	100%
Ajout de traits humains	25%	100%



## 4 Conclusion

Notre projet Chrono Park va, par conséquent, permettre de représenter le trafic humain dans un parc d'attraction avec la possibilité de manipuler des paramètres en entrée tels que le nombre d'attractions ou encore le nombre d'humains. De plus, comme dit dans les précédentes parties nous devons pour cela implémenter de nombreux algorithmes et le plus d'options possible pour rendre le tout réaliste.

## 5 Bibliographie

DELAIGUE, Alexandre. "Classe éco été : comment réduire les files d'attente au parc d'attraction?". 2013. Franceinfo.

<https://blog.francetvinfo.fr/classe-eco/2013/07/24/classe-eco-ete-comment-reduire-les-files-dattente-au-parc-dattraction.html#:~:text=Le%20principe%20est%20le%20suivant,fera%20passer%20en%205%20minutes.>

CORRADO, Henry. "Parcs d'attractions et files d'attente". Tejix.

<https://www.tejix.com/fr/paperqueuelines.html>