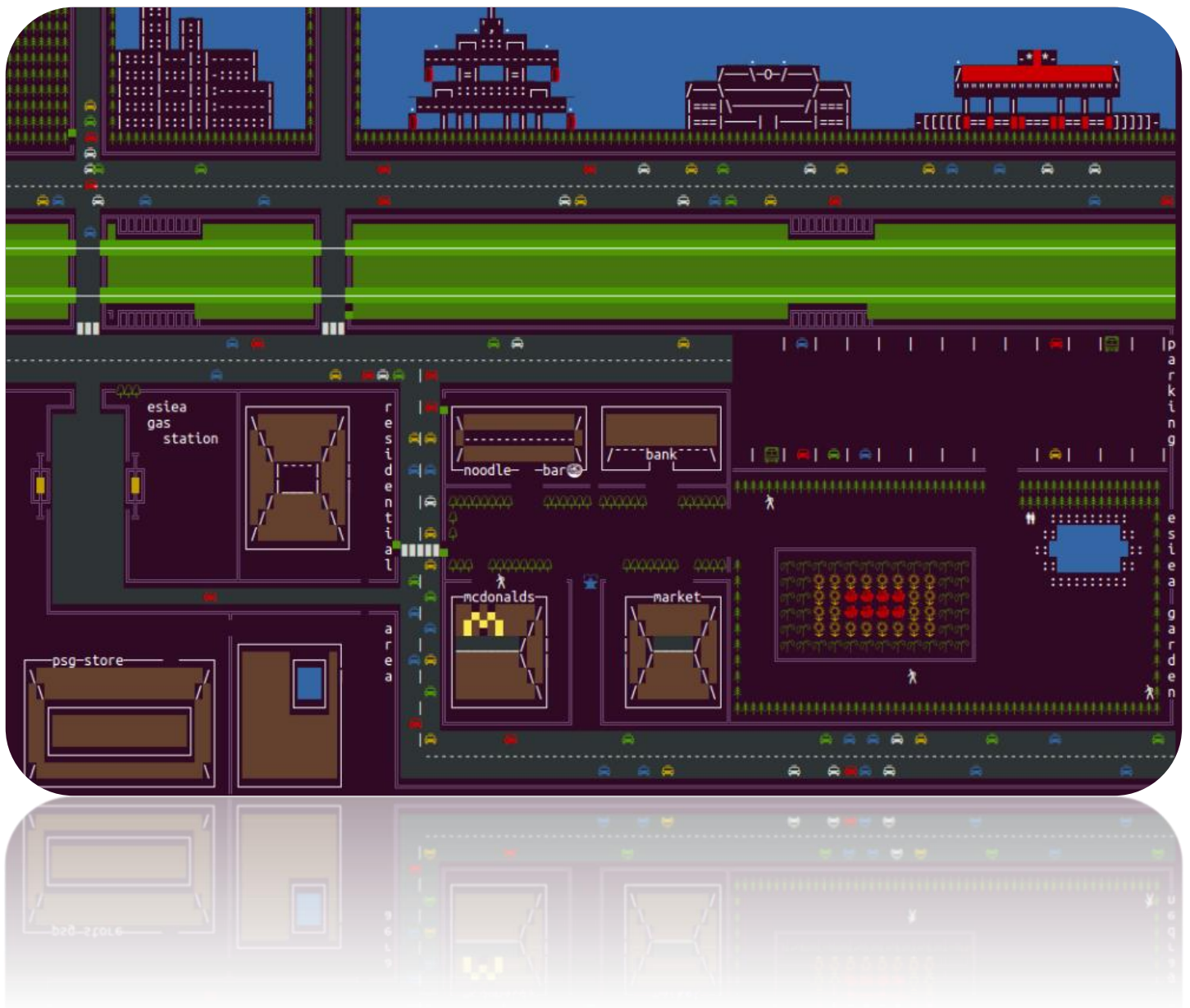


Rapport Projet C

The 13th District



Sommaire

Le jeu	3
La carte	3
Le déplacement des voitures	4
Les feux et le tramway	5
Les piétons.....	5
La station-service.....	6
Le parking et la vie des piétons dans la ville	6
Le mode danger.....	7
Organisation	8
Conclusion	9

Le jeu

The 13th District est une simulation de trafic routier. Nous nous sommes inspirés du 13^{ème} arrondissement de Paris pour réaliser les routes et les décors de notre carte. Dans ce simulateur des voitures peuvent se déplacer librement sans se rentrer dedans ou générer d'accidents.

Elles respectent les ordres de priorité, les feux, et n'écrasent pas les piétons.

Ces derniers peuvent sortir de leur voiture pour remettre de l'essence dans leur véhicule si besoin, sinon pour aller faire des courses au supermarché ou bien aller au restaurant tout en passant par le parc.

Cette application étant une simulation, l'utilisateur a juste à choisir le mode souhaité puis il n'a plus besoin de faire quoi que ce soit. Il peut observer la vie dans le 13^{ème} arrondissement de Paris.

Pour jouer : Avoir installé la librairie « sox » afin de pouvoir profiter du son.

```
sudo apt-get install sox
sudo apt-get install libsox-fmt-all
```

Si l'affichage vous parait étrange. Réduisez l'affichage de la console.

La carte

Comme précisé plus tôt, la carte de ce jeu a été réalisée avec comme « *modèle* » le célèbre quartier asiatique de Paris.

Nous avons dessiné la carte de notre jeu dans un fichier texte avec des symboles :



Nous avons ensuite importé ce fichier texte dans notre programme puis dans une matrice. Ainsi nous pouvions remplacer ces symboles par ceux dont nous avons besoin pour l’affichage (*ex : ‘R’ est le symbole pour une voiture dans le jeu*) et leur affecter une couleur et/ou une couleur d’arrière-plan.

Cette partie ne nous a pas causé de gros problèmes. Mettre cette carte sous forme de symboles à taper dans un fichier texte fut surtout très long. Nous avons seulement à réfléchir à l’agencement de nos routes, par où pourront passer les piétons et quels décors mettre.

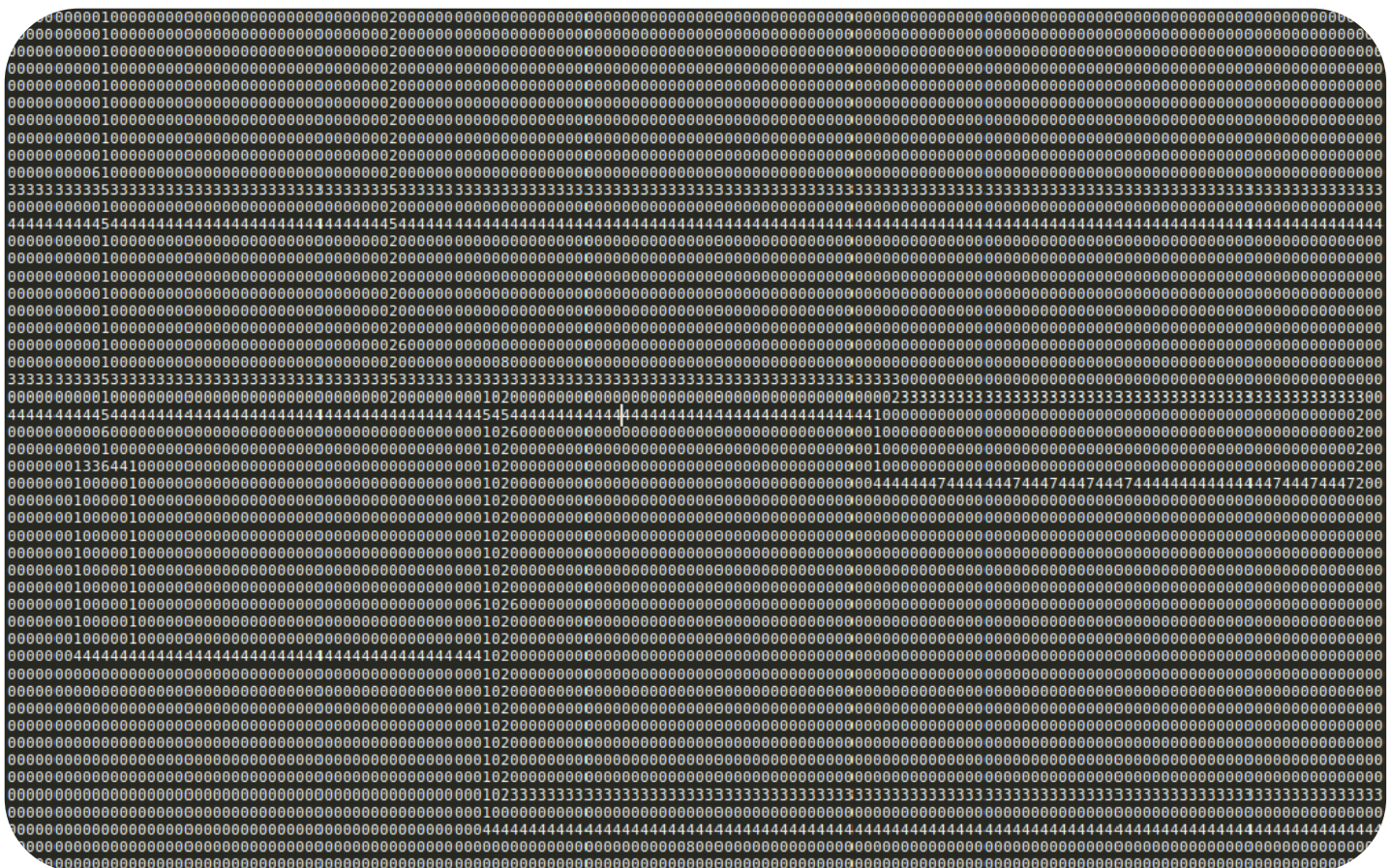
Le déplacement des voitures

Nous savions d’entrée de jeu que nos voitures allaient être contenues dans une liste chaînée. Cependant penser à la manière de les faire avancer et diriger à leur destination voulue nous a demandé un effort de réflexion.

Nous avons donc créé une matrice spécialement pour le déplacement des voitures. Cette matrice est de la même taille que la carte et ne contient que des chiffres.

Un chiffre correspond à un déplacement. Par exemple si une voiture se trouve sur la carte à la même position qu’un ‘1’ sur la matrice. Alors la voiture va descendre d’une case.

Nous avons donc placé tous ces chiffres sur notre matrice aux mêmes positions que les routes sur notre carte.



Faire avancer une voiture d'un point A à un point B fut alors chose facile avec cette méthode. Cependant une voiture doit pouvoir aller à plusieurs endroits. Pour cela nous avons dû rajouter à cette matrice un chiffre correspondant à un **point critique**.

Lorsqu'une voiture est sur la position d'un **point critique**. Elle va se demander quelle est sa destination (*Nous avons défini la destination de manière aléatoire lors de la création de la voiture*). Si elle veut aller au Nord et qu'il existe une route une case au-dessus d'elle, alors la voiture va **prioriser** cette route sur les autres.

Ainsi, à chaque boucle de jeu, nous parcourons toutes les voitures de la liste et, en fonction de sa position sur la Matrice, nous la déplaçons.

Ce déplacement ne peut se réaliser seulement s'il n'y a pas d'autres voitures ou piétons devant, et si la voiture ne sort pas de la carte. Dans ce cas la voiture sera effacée de la liste.

Les feux et le tramway

Pour réaliser ces deux fonctions, nous avons décidé de compter toutes nos boucles de jeu (*à chaque fois qu'on affiche la carte et que les voitures se déplacent donc d'une case*) et d'en faire des cycles.

Un cycle est composé de 60 boucles de jeu dans notre cas. Nous avons donc affiché **le feu vert** seulement entre la 1^{ère} et la 30^{ème} boucle. Et **le feu rouge** entre la 31^{ème} et la 60^{ème}.

Pour que les voitures s'arrêtent, nous avons rajouté une condition, si les voitures sont proches du feu et que nous sommes entre la 1^{ère} et 30^{ème} boucle d'un cycle, alors elles ne bougent pas.

Nous utilisons aussi cette méthode où nous comptons les boucles pour le tramway. Il apparaît donc de manière régulière au début d'un cycle sur trois.

Les piétons

Au départ nous comptons faire des piétons qui se déplacent seulement de manière aléatoire dans un parc. Nous avons réalisé plus tard qu'il serait impossible de faire sortir nos piétons d'une voiture avec cette méthode.

Nous avons donc, comme pour les voitures, créé une matrice sous forme de fichier texte, composée de chiffres correspondant à des déplacements. Et comme pour les voitures, nos piétons sont stockés dans une liste.

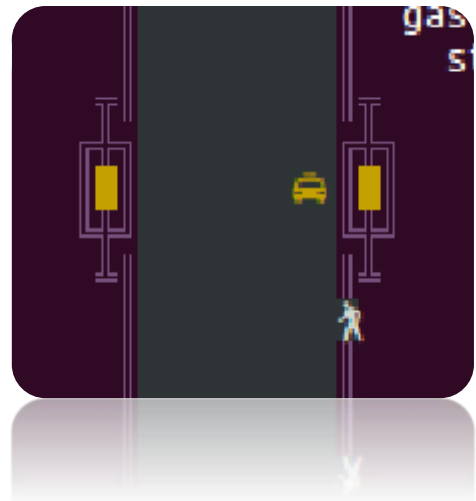
Comme nous avons déjà utilisé ce type de Matrice avant, faire les conditions et le fichier texte fut rapide et sans problèmes.

La station-service

Nous avons créé dans notre structure voiture. Des voitures de type « *ravitaillement* ». Qui ont besoin de passer par la station-service reprendre de l'essence avant de rouler jusqu'à leur destination.

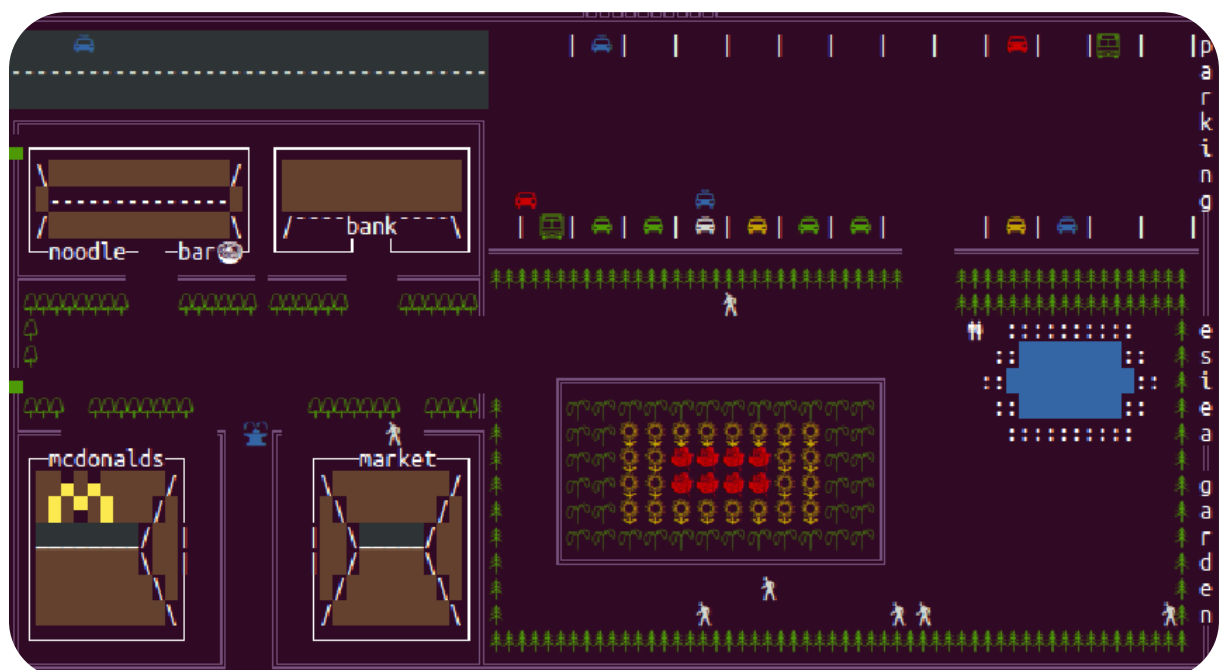
Elles sont générées aléatoirement, quand une voiture arrive à la station essence elle va en priorité sur la station de droite, (*qui est la plus proche de la sortie*) si elle est occupée elle va à celle de gauche.

Quand une voiture est arrêtée sur une station essence, elle passe à l'état inactif. Un piéton apparaît alors, passe à la station essence, et quand il revient à la même position que celle de la voiture, elle repasse à l'état actif et le piéton est effacé de la liste. La voiture perd alors son état « *ravitaillement* » et peut donc retourner à sa destination.



Le parking et la vie des piétons dans la ville

Le parking et la vie des piétons fut la partie la plus complexe à réaliser de ce projet pour nous. Nous avons dû répondre à diverses problématiques, « *Comment faire un parking sachant qu'avec les matrices de déplacement les voitures et les piétons ne peuvent pas revenir sur leur pas ?* » ou bien « *Avec le peu de place disponible comment faire pour que les piétons puissent tous aller jusqu'à la zone de restaurants et puis revenir sans croiser ceux qui repartent ?* »



Pour pouvoir s'adapter à notre façon d'approcher ces problématiques, nous avons modifié un peu la carte afin de faire plus de place pour les piétons et pour que les voitures puissent faire le tour du parking pour en sortir.

Comme pour les voitures nous avons dû ajouter des points critiques sur la matrice de déplacement des piétons. Ils vont aléatoirement dans les 4 bâtiments présents sur la carte. Nous avons aussi ajouté la position de la voiture qui a déposé le piéton dans la structure du piéton. Il est ainsi capable de retrouver la voiture qui l'a déposé et de pouvoir repartir.

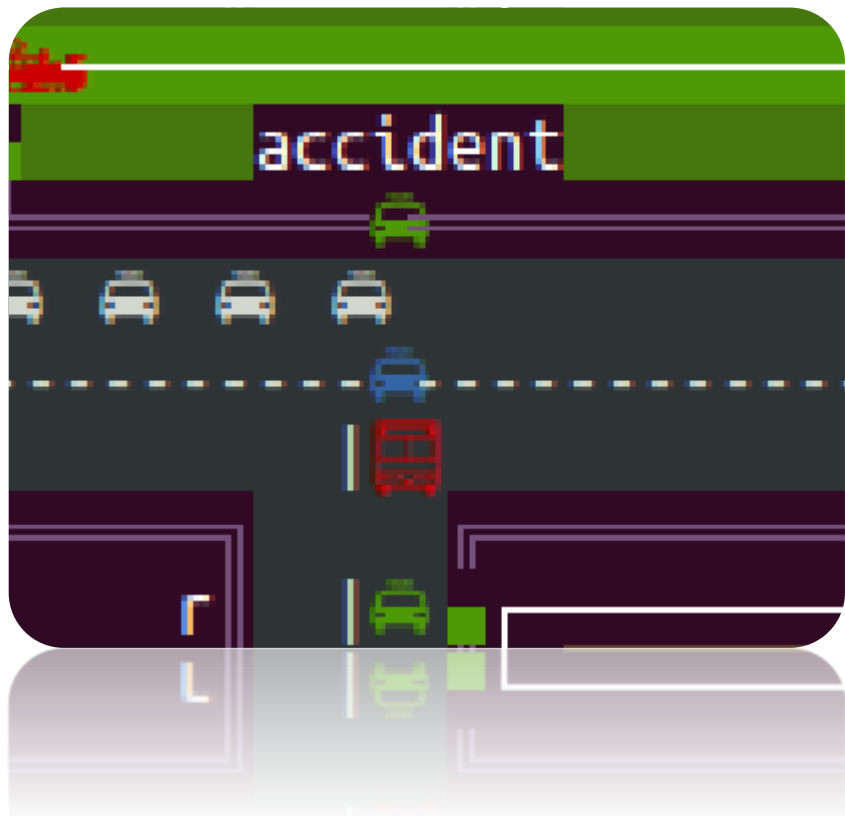
Quand une voiture récupère son piéton, elle sort de sa place de parking et perd son état « parking » elle ne cherche donc plus à se garer dans une place de parking et fait le tour pour sortir.

Le mode danger

Pour le mode danger, la première chose que nous avons faite est de se réunir pour discuter de quelles fonctionnalités nous pourrions rajouter. Nous étions limités par le temps et nous ne pouvions pas par conséquent faire des modifications sur la carte pour l'occasion.

Nous avons donc fait en sorte que des voitures rentrent dans des murs et y restent coincées jusqu'à l'arrivée des pompiers. A chaque fois qu'une voiture rentre dans un mur, un pompier apparaît sur la carte, quand il passe à côté de la voiture, elle est alors effacée de la liste.

Autre point du mode danger.
Au lancement du jeu une voiture de type « folle » apparaît sur la route EST de la carte. Cette voiture a des déplacements complètement aléatoires (avec un petit peu plus de chance d'aller vers la gauche pour ne pas qu'elle reste toujours au même endroit). Les autres voitures doivent donc s'adapter pour ne pas faire d'accident avec cette voiture « folle ».



Organisation

Afin de pouvoir rendre le projet dans les temps et d'être le plus efficace possible une bonne organisation est un élément à ne pas sous-estimer lors de la réalisation d'un projet.

Premièrement nous avons réfléchi à l'ordre dans laquelle nous allons réaliser chacune des fonctionnalités auxquelles nous avons pensés (*carte, voitures, piétons, feux...*). Nous avons donc pensé qu'avoir un support de base est le plus important avant de se lancer dans toutes autres choses.

La première étape était donc de réaliser la carte et de faire les voitures et leurs déplacements. Thomas s'est donc occupé de « dessiner » la carte et Johan de créer la liste pour les voitures ainsi que la matrice de déplacement et les conditions qui vont avec.

Une fois notre support de base fonctionnel, l'ordre de réalisation du reste des fonctionnalités n'était pas important. Nous avons donc essayé de nous débarrasser des étapes les plus simples. Donc premièrement les feux et le tram, ensuite la station-service et les piétons, puis enfin le parking. Le mode danger fut réalisé une fois que le mode normal fut fini.

Conclusion

Nous avons fait du langage C depuis la première année. Nous avons ainsi acquis beaucoup de connaissances à la fois théoriques mais aussi pratiques. Ce projet est le résultat de notre expérience, et nous sommes fiers de son aboutissement.

Ce travail représente nos efforts. Nous espérons que peu importe la personne qui lance notre simulation, elle appréciera le rendu de cet investissement.

Bien évidemment nous espérons aussi que vous aimerez votre voyage dans ce quartier exotique de Paris.