

# Gestion automatique de la traversée d'une intersection intelligente

## Projet d'Orientation Master - M1 Informatique

**BRUNEAU Richard**

MAI 2020

---

**Résumé :** Ce projet a pour but de concevoir une gestion automatisée de la traversée d'une intersection intelligente par des véhicules autonomes. L'objectif est d'optimiser le passage de ces véhicules ainsi que leur temps d'attente en cas d'arrêt à l'intersection.

**Mots-clés :** Voiture autonome, intersection intelligente

## 1. Introduction

Ce projet s'inscrit dans le cadre de l'Unité d'Enseignement Projet pour l'Orientation en Master et est supervisé par Monsieur Samir AKNINE, chercheur au LIRIS.

L'objectif de ce projet était de réfléchir et de modéliser une intersection intelligente pour des voitures autonomes. En effet, le trafic urbain devient de plus en plus condensé provoquant de nombreuses réactions tels que de la pollution ou des embouteillages. Parmi les méthodes de résolutions à ses problèmes, la régulation du trafic de manière intelligente grâce à la communication entre véhicules semble une solution adéquate. Cette solution rendu envisageable grâce aux véhicules de plus en plus connectés et à l'essor de l'intelligence artificiel présente de nombreux avantages. Entre autres, ce procédé limiterais le nombre de freinage et donc l'usure des plaquettes et des disques de freins, le temps passé à l'arrêt ainsi que les périodes d'accélérations très énergivores. Une telle résolution permet d'impacter directement les conséquences évoquées précédemment.

Afin de faciliter la réflexion et de réduire la problématique, ce projet se concentre spécifiquement sur les véhicules autonomes à l'échelle d'une intersection avec quatre routes qui rentrent et sortent du croisement.

Dans un premier temps, je vais aborder les différents outils qui m'ont permis de réaliser ce projet, puis je détaillerais le travail réalisé en différente sous partie. Enfin, j'expliquerai ce que j'aurais souhaiter ajouter à mon projet si j'en avais eu la possibilité puis je conclurais ce rapport.

## 2. Outils utilisés

Dans un premier temps, nous avons réalisé une étape de recherches sur les différents outils que nous pouvions utiliser afin de réaliser ce projet. A l'aide des enseignements que nous avons suivi précédemment en Licence et au cours du premier semestre du Master, nous avons choisi de programmer en Java. Par conséquence, nous avons opté pour l'utilisation du framework open-source JavaFX afin de visualiser les effets de l'implémentation de notre code. Ce framework me permet d'obtenir une interface graphique afin de pouvoir visualiser une représentation séquentiel des agissements de chaque véhicule en fonction du comportement que j'ai implémenté.

Au cours de nos discussions sur les outils, j'avais fait la concession, afin de satisfaire l'autre partie du groupe, de ne pas utiliser Maven ni JUnit. Aujourd'hui, je regrette et assume seul ce choix qui aurait sans doute faciliter et accélérer le déroulement du projet.

### 3. Travail réalisé

#### 1. L'algorithme

Après avoir discuté au sujet des outils que nous utiliserions, nous avons réfléchi à l'élaboration d'un algorithme pour le passage d'un véhicule sur l'intersection. L'algorithme que nous avons élaboré et auquel j'ai apporté quelques modifications durant l'implémentation repose sur trois étapes.

##### 1. L'inscription

Au cours de cette étape, le véhicule est en mouvement et s'approche de l'intersection, comme vous pouvez le voir sur la figure 1 ci-dessous.

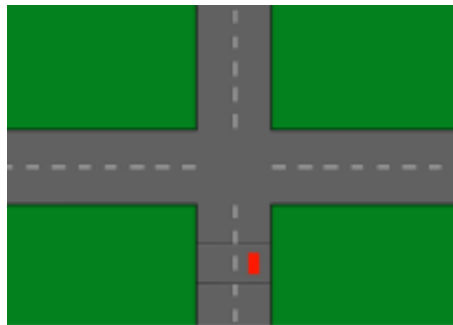


Figure 1 : Le véhicule s'inscrit sur liste d'attente

Le véhicule accède à l'intersection et s'inscrit sur une liste d'attente afin de pouvoir passer sur le croisement. La file d'attente est une file First In First Out (voir représentation dans la figure 2, ci-dessous) où le premier véhicule à s'être inscrit est le premier véhicule à passer.



Figure 2 : Représentation d'une liste First In First Out

Sur l'image ci-dessus, sur la première ligne, la liste est vide. Le véhicule bleu vient s'ajouter à la liste puis le véhicule orange. Le véhicule bleu, qui est le premier à s'être inscrit sort de la liste puis le véhicule orange sort, respectant ainsi le principe du « premier rentré, premier sortie ».

En revanche, afin d'optimiser le temps d'attente et le nombre de véhicules qui peuvent franchir l'intersection au cours d'un temps donné, le dernier véhicule inscrit n'est pas forcément le dernier à passer. Nous verrons ce point dans la partie suivante qui détaillera l'optimisation du passage des véhicules.

## 2. L'attente

Tout d'abord, cette étape d'attente est une étape facultative. En effet, bien que physiquement (voir figure 3) la voiture est obligée de passer sur le lieu de l'attente, elle n'est pas obligée de s'y arrêter. Si aucun véhicule n'est présent sur le croisement, ou bien si plusieurs véhicules peuvent franchir le croisement simultanément alors l'attente n'aura pas lieu.

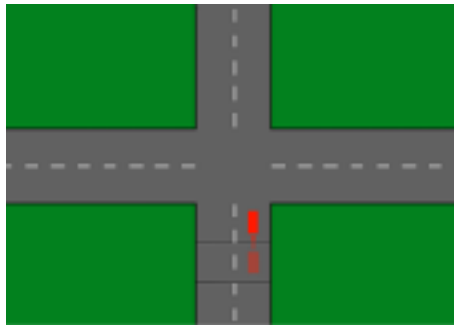


Figure 3 : La voiture est physiquement sur la case d'attente mais n'attend pas forcément

## 3. Le franchissement

Au fil de cette étape, le ou les véhicules pouvant franchir l'intersection sans être à l'origine d'un accrochage vont la traverser. J'ai isolé une zone dite de conflit au centre de l'intersection. C'est ici que le risque d'accrochage est présent. Cette zone de conflit peut être décomposée en quatre cases.

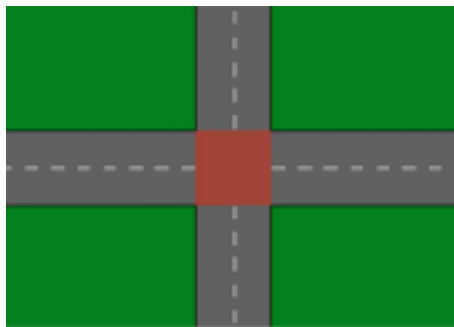


Figure 4 : Représentation graphique de la zone de conflit

L'algorithme a été spécialement réfléchi pour qu'à l'intérieur de cette zone les véhicules soient en permanence en mouvement sans risque de collision. En effet, si deux, trois ou quatre véhicules se trouvent simultanément sur le croisement, en aucun cas ils ne partagent case sur le trajet, comme sur l'exemple de la figure 5 contrairement à la figure 6 où la case en rouge est partagée.

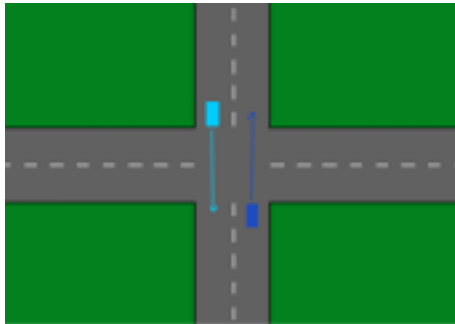


Figure 5 : Deux voitures franchissent le croisement sans partage de case

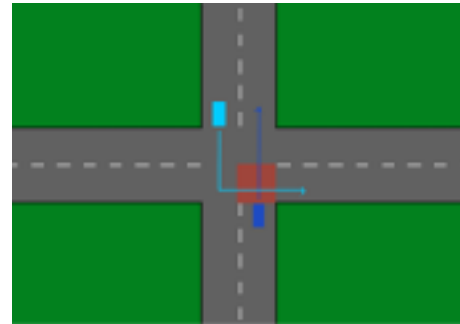


Figure 6 : Deux voitures partagent la même case

J'ai pris la décision de ne pas partager la même case pour une même séquence pour différentes raisons. Dans la réalité du terrain, je pense qu'il est nécessaire de se laisser une marge d'erreur. Une erreur sur le calcul de la vitesse, un temps de latence trop important où bien un piéton qui traverse inopinément la chaussée et c'est l'accrochage. Le débit réel du nombre de voiture au cours d'un temps donné sera toujours inférieur au débit théorique à cause des précautions qui sont prises.

De plus, dans l'implémentation de mon programme j'ai dû légèrement décaler les temps d'actions de chacun des véhicules. De ce fait, il était important d'éviter que visuellement deux véhicules se superposent alors qu'ils n'étaient pas censé être au même endroit au même moment dans la réalité du code.

## 2. L'optimisation

### 1. Situations

Sans aucune optimisation, les véhicules passent individuellement sur l'intersection. Dans la situation de la figure 7 ci-dessous bien que violet et bleu puissent passer en même temps, ainsi que noir et orange, les véhicules passeront un par un. Cette situation prendra quatre temps pour faire franchir l'intersection à toutes les voitures. Alors qu'avec l'optimisation implémentée dans mon code, nous pouvons en faire passer deux à la fois et ainsi réduire le nombre de temps nécessaire à deux.

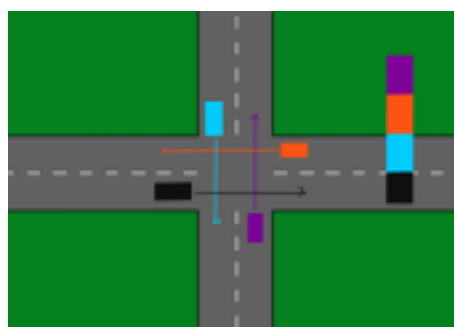


Figure 7: Représentation d'un cas où l'optimisation est possible

Bien évidemment, il n'est pas forcément possible de faire franchir plusieurs véhicules en même temps. Ci-dessous dans la figure 8, chaque paire de véhicule partage au moins une case

avec chaque autre paire bloquant ainsi l'optimisation. Quatre temps seront nécessaire pour faire passer ces quatre voitures.

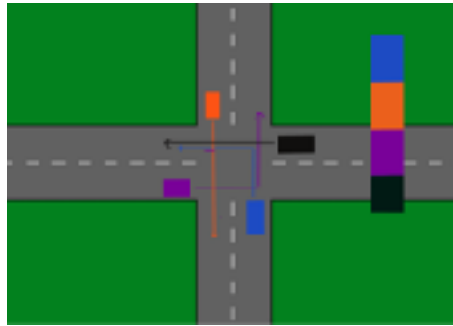


Figure 8: Représentation d'un cas où aucune optimisation n'est possible

Je vais maintenant présenter une seconde situation où deux voitures recouvrent l'intégralité des cases de la zone de conflits. Dans une partie des cas, il est possible de faire circuler deux véhicules en même temps comme nous l'avons vu précédemment dans le cadre de la figure 5. En effet il était instinctif que deux voitures circulantes dans des directions contraires n'ont aucun risque de se croiser. Dans ce type de cas, les deux voitures recouvrent à elles seules toutes les cases dites de conflits, du croisement. En revanche, une autre situation moins évidente où deux véhicules recouvrent l'entière des cases de l'intersection est la situation de la figure 9, quand l'un des deux véhicules tourne à droite et le second tourne à gauche.

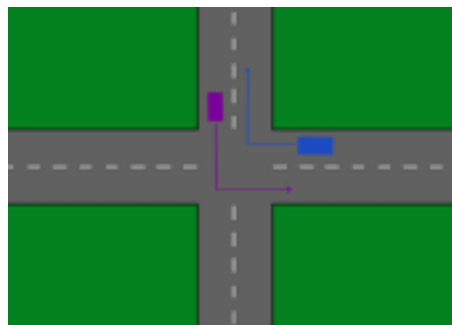


Figure 9 : Deux voitures recouvrent l'intégralité des cases conflits

Ces deux situations différentes nous permettent de constater qu'il n'est pas possible de se baser sur la direction de la voiture pour déterminer le nombre de véhicules qui peuvent franchir l'intersection.

Il existe aussi des possibilités où trois véhicules peuvent franchir l'intersection simultanément comme dans la figure 10 sur la page suivante ainsi que quatre véhicules avec la figure 11 à ses-côtés.

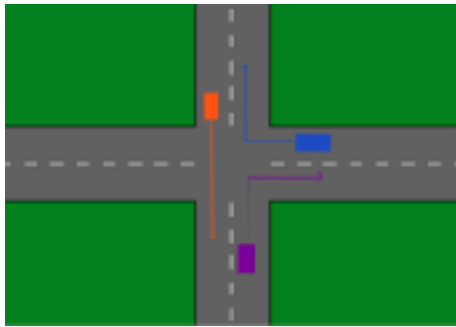


Figure 10: Trois véhicules recouvrent l'intégralité des cases

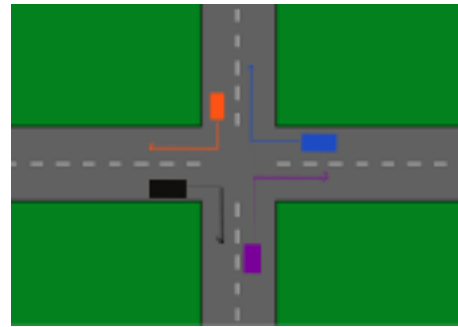


Figure 11: Quatre véhicules recouvrent l'intégralité des cases

## 2. La réflexion

Après avoir constaté les différentes situations, il a été important de chercher qu'elle était la priorité pour combler les cases vides. Le premier élément de la file d'attente est forcément le premier. Sur une situation comme la figure 12 ci-dessous, devais-je forcément faire passer le deuxième si il le pouvait ou bien devais-je faire passer le plus de véhicules possible en faisant passer le véhicule orange avec le bleu et le violet ? Devais-je chercher à vider les rues les plus remplies en premier ?



Figure 12 : Représentation d'une situation sujette à réflexion

J'ai pris la décision de faire passer le plus de véhicule simultanément pour combler les cases laissées par la tête de la file. Cela engendre la concession de ne pas faire passer le deuxième véhicule de la file quand on peut faire traverser le troisième et le quatrième. Cette concession n'influe pas énormément sur le temps d'attente du véhicule lésé, puisque celui-ci sera le prochain à traverser.

## 3. La réalisation

Pour réaliser cette optimisation, j'ai utilisé une fonction dans ma classe véhicule qui calcule le nombre de voitures pouvant circuler en fonction des quatre premiers inscrits sur la file d'attente. Cette fonction se sert de la direction que possède la voiture en tête de liste afin de déterminer si les trois autres voitures possèdent des directions complémentaires avec la première. En fonction de la complémentarité on obtient le nombre de véhicule pouvant circuler simultanément sur les cases de conflits.

### 3. Modularité

Au fil de ce projet semestriel, j'ai tenté de respecter le plus possible les principes de la programmation objet et du modèle vue-contrôleur.

J'ai créé une classe voiture qui hérite de la classe véhicule. J'aborderai cela dans la partie suivante, mais j'avais pour projet de faire circuler différent type de véhicules sur le croisement. Bien que ce projet est été réalisé en totale autonomie je pense qu'il est important d'essayer de conserver les bonnes habitudes.

### 4. Ce que j'aurais voulu ajouter

Mon objectif initial était d'arriver à ce niveau d'avancement beaucoup plus rapidement. En effet, cela m'aurait permis d'ajouter différentes choses afin de rendre ce projet plus complet et plus réaliste.

#### 1. Différents types de véhicules

##### 1. Des classes pour les véhicules long

L'une de mes premières idées d'améliorations était de rajouter une classe camion pour simuler les longs véhicules comme les camions, les bus ou les véhicules avec remorques. Ce type de véhicules particuliers aurait apporter de grandes contraintes sur un croisement. Un camion qui tourne à droite va empiéter sur des cases où les voitures ne vont pas pour effectuer la même manoeuvre. Cela implique même dans certains cas l'évacuation de la voiture qui attend avant que le camion ne tourne afin de ne pas gêner comme sur la figure 13, ci-dessous.

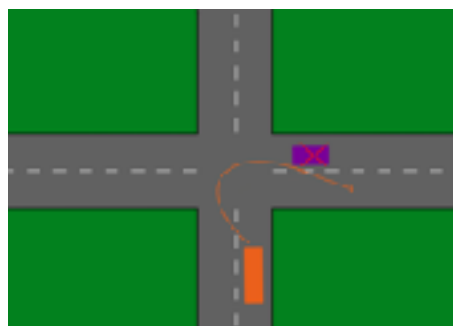


Figure 13: Représentation de la situation d'un camion qui tourne

##### 2. Une classe pour les deux roues motorisés.

Dans la lignée de l'amélioration précédente, il aurait été intéressant de définir une classe pour les deux roues motorisés. Je ne prends pas en compte les vélos car ce ne sont pas des véhicules motorisés autonomes, en revanche, une moto autonomes est envisageable sur le papier. Les motos n'ont pas les mêmes contraintes que les voitures. Il semble tout à fait envisageable de faire avancer une moto sur une partie du croisement afin de libérer la place pour les véhicules derrière elle dans le cas où celle-ci tourne à gauche. Sur la figure 14, ci-dessous, le fait de faire avancer la moto orange, permet de libérer la place pour la voiture noire ainsi que la voiture verte.

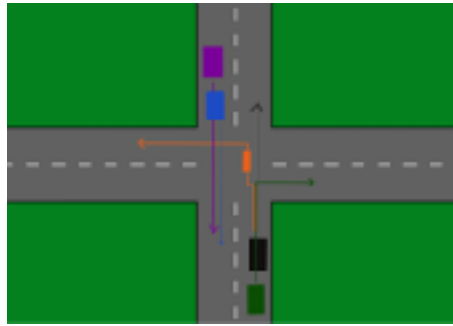


Figure 14: Représentation de la place libérée par une moto qui se déplace pour laisser passer des véhicules

## 2. Prise en compte de la vitesse

Ce projet ne prends absolument pas en compte la contrainte de vitesse, ni les principes d'accélération et de freinage. Si initialement cela n'a pas été pris en compte c'était afin de faciliter la réalisation du projet. En revanche, je regrette ne pas avoir eu le temps de me concentrer sur cet aspect intéressant plus tard dans le semestre. Une réflexion aurait dû être faite sur la manière dont il fallait accorder la priorité à certain véhicule. En effet, un véhicule en mouvement devrait être prioritaire vis-à-vis d'un véhicule en fonctionnement grâce au bénéfice écologique de lui éviter un freinage et une accélération.

En revanche, il est intéressant de se mettre également à la place du passager du véhicule qui voit passer quelqu'un devant lui, au sens propre comme au sens figuré, alors qu'il attend son tour sagement depuis plus longtemps.

## 3. Prise en compte de véhicule prioritaire

Dans notre société actuelle, il est important de prendre en considération les véhicules prioritaires dans ce genre de situation. Actuellement, l'algorithme ne considère pas la présence d'un véhicule de secours ou des forces de l'ordre. Pour incorporer cette contrainte au programme actuel, j'aurais créé une deuxième file d'attente. Celle-ci aurait été réservée aux véhicules prioritaires et les véhicules de l'autre file d'attente n'aurait pas pu s'engager si la première comportait un élément à l'intérieur de la liste.

Un second niveau d'incorporation de cette contrainte aurait été de « vider » la voie où roule le véhicule prioritaire afin de lui éviter de rouler sur la voie de gauche. Dans ce cas là, les véhicules présents devant lui auraient été prioritaires pour passer sur le croisement. Cette éventualité est envisageable uniquement si le véhicule prioritaire « contacte » le croisement suffisamment tôt pour que tous les véhicules de sa voie puissent partir à temps.

Enfin, un troisième niveau d'incorporation aurait été dans le cas d'un long camion de pompier, avec une grande échelle par exemple, qui tourne à droite, l'évacuation du véhicule potentiellement gênant comme dans le cadre de la figure 13 vue précédemment. Ce troisième niveau présente également l'avantage pour les véhicules léger de prendre un virage plus rapidement.

## 5. Conclusion

Durant ce projet pour l'orientation en master, j'ai effectué un travail de recherche et d'analyse afin de comprendre comment pouvait fonctionner une intersection intelligente. J'ai du réfléchir aux différentes façons d'augmenter le débit de voiture une fois la première partie



réalisée. J'ai été soumis à différentes contraintes et imprévus, durant ce projet, qui m'ont permis de me focaliser sur une problématique précise de l'intersection intelligente pour voiture autonome. Bien que le déroulement de ce projet fut très particulier, ce fut une expérience très intéressante et instructive.

Malheureusement, je n'ai pas pu ajouter tous les comportements que je souhaitais ce qui laisse un parfum d'inachevé. J'espère avoir été clair et compréhensible au cours de la partie précédente sur ce que j'aurais souhaité réaliser si j'en avais eu la possibilité. Dorénavant, j'aurais une approche plus précise sur la problématique des voitures autonomes et sur ce qui les entourent.

Pour conclure ce rapport, je souhaite ajouter que j'ai trouvé ce sujet passionnant. Réaliser ce projet m'a conforté dans l'idée de m'orienter en Master 2 Intelligence Artificielle.