

Département d'informatique

IFT 609 - Informatique cognitive

Rapport de projet - Automne 2021

# Conduite sur autoroute



Encadré par : M. Mohamed Mehdi Najjar

Réalisé par : Andreas Duquenne Lilian Favre Garcia

Andrianihary Tsiory Razafindramisa

<b>Rapport de projet - Automne 2021</b>	<b>1</b>
Introduction	3
Description du domaine	3
Problématique	3
Cas de figure possibles	3
Représentation des connaissances	5
Description textuelle des types de chunks	5
Buts et sous-buts	6
Procédures	6
Scénario	7
Déroulement	7
Trace complète	7
Apprentissage	11
Graphe d'apprentissage avec optimisation des paramètres	11
Interprétation des résultats des simulations	12
Ce que nous avons essayé d'implémenter	12
Une complexification de notre modèle	12
Conclusion	13
Limites	13
Améliorations	13
Possibles extensions	14

## Introduction

### Description du domaine

Conduire sur autoroute est un exercice que chaque conducteur doit apprendre. Cette conduite est différente de celle pratiquée en ville et nécessite dans un certain sens plus de prudence et de vigilance. Nous avons donc choisi de traiter ce sujet.

### Problématique

Un conducteur conduit une voiture sur une autoroute. Un accident survient devant lui. Comment peut-il faire en sorte de s'arrêter ou d'éviter l'accident sans sortir de la route ?

Le but de notre étude sera de minimiser les dégâts causés par un accident sur une autoroute en évitant de créer un suraccident. Pour cela, le modèle devra analyser la situation (distance, poids du véhicule) entre sa voiture et l'accident devant lui et choisir entre effectuer un freinage ou effectuer un changement de voie.

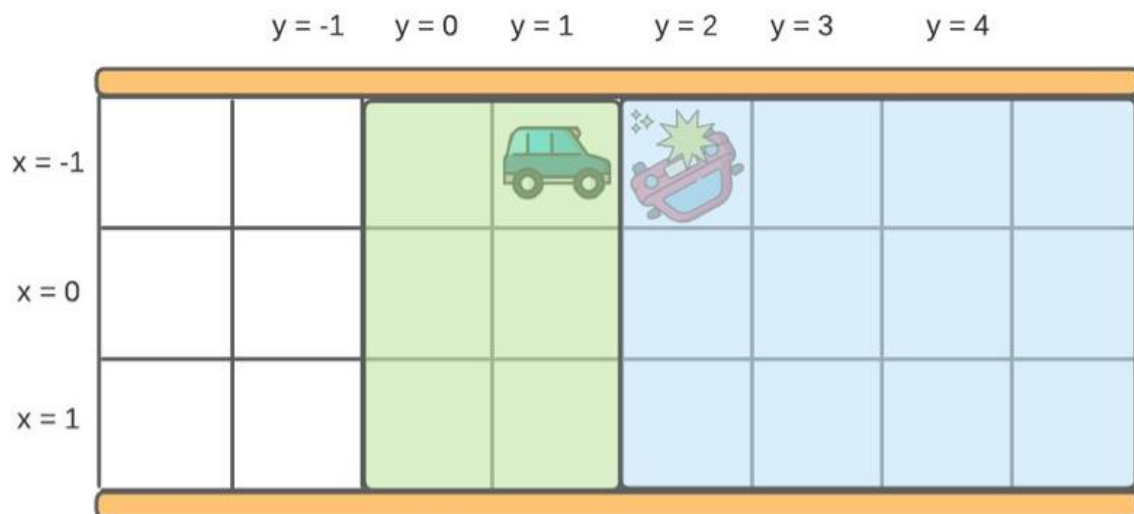
### Cas de figure possibles

Afin de simplifier notre étude, nous proposons plusieurs hypothèses.

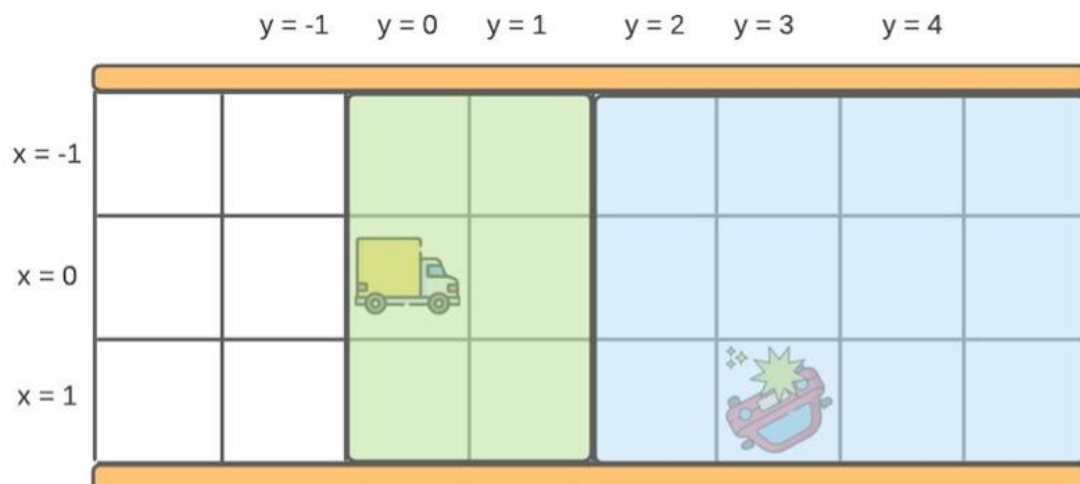
- On suppose que l'autoroute possède toujours trois voies. Il n'y a pas de voie d'arrêt d'urgence.
- On suppose qu'il n'y a que notre voiture et les voitures impliquées dans l'accident sur la route. Personne n'est derrière notre voiture et les voies latérales sont vides.
- Le mouvement de départ de la voiture est rectiligne uniforme.
- Il n'y a que deux puissances de freinage : une forte et une faible.
- L'état de la route est uniforme et il n'y a pas de dégâts causés par des intempéries.
- La gravité de l'accident sera négligée. Si notre voiture érafle un obstacle, ce sera considéré comme un accident.

Selon ces hypothèses, nous nous retrouvons avec différents cas de figure possibles puisque notre conducteur peut se retrouver dans différentes situations.

- Un véhicule en amont freine brusquement ou un obstacle se trouve devant notre voiture.



- Un élément dangereux se trouve sur une des voies latérales ou une collision se produit sur une des voies latérales.



Nous allons agir en prenant en compte la distance de freinage disponible, la localisation de l'obstacle ou de l'accident, et la caractéristique de notre voiture, le poids de la voiture pouvant varier. Ce changement va altérer la capacité de freinage du véhicule, voilà pourquoi nous relevons deux puissances de freinage. Le conducteur aura soit une voiture compacte soit un camion.

## Représentation des connaissances

### Description textuelle des types de *chunks*

```
(chunk-type check-state state result m_weight m_positionX m_positionY m_vitesse a_positionX a_positionY a_vitesse action)
(chunk-type learned-info result m_weight m_positionX m_positionY m_vitesse a_positionX a_positionY a_vitesse)

(chunk-type car id weight)
(chunk-type position id positionX positionY)
(chunk-type speed id vitesse)
```

Le chunk-type check-state est un chunk destiné au goal. Il permet de récupérer les informations des fonctions LISP.

- Le slot state sert à contrôler l'exécution des productions.
- Le slot result représente la réussite ou l'échec de l'essai. Après avoir déplacé la voiture, si notre véhicule est sur la case de l'accident, ou s'il est sur la même position en x que l'accident mais une position en y plus grande (notre voiture est devant l'accident en lui passant au-travers), alors c'est un échec car il n'aura pas évité l'accident.
- Les autres slots représentent les paramètres de notre véhicule et les paramètres de l'accident : le poids de notre voiture, les positions et les vitesses de notre voiture et de l'accident.

Le chunk-type learned-info est celui qui est destiné à l'apprentissage. Il permet de conserver les informations après un essai. Ce chunk permet de récolter de nouvelles connaissances, de les joindre dans la mémoire déclarative en sauvegardant le poids, la position et la vitesse de notre véhicule, ainsi que la position et la vitesse de l'accident.

Les chunks-types car, position et speed paramètrent le modèle et l'accident, et sont utilisés pour les connaissances. Ils déterminent différentes combinaisons de véhicule, de position et de vitesse. Les slots id permettent de faire le lien entre les paramètres d'un même véhicule.

- Le chunk-type car paramètre le poids de notre véhicule et précise donc s'il s'agit d'une voiture ou d'un camion.
- Le chunk-type position donne la position en x et en y de notre véhicule et de l'accident.
- Le chunk-type speed donne la vitesse de notre véhicule et de l'accident.

## Buts et sous-buts

Le but principal de notre projet est d'éviter un suraccident lors d'un accident sur une autoroute. Pour cela, notre véhicule doit basculer à gauche ou à droite, ou freiner doucement ou fortement. La simulation devra apprendre en comparant les situations déjà rencontrées et les actions choisies en conséquence afin de choisir la meilleure suite d'actions à effectuer dans une situation spécifique.

Voici comment le modèle prend une décision :

Si le modèle ne se souvient de rien, le modèle freine doucement, puis enregistre son résultat.

Si le modèle se souvient d'une situation et que cela a bien fini, il répète l'action.

Si le modèle se souvient d'une même situation mais que cela a fini en crash, il tente une autre action.

## Procédures

La procédure start, autrement dit la première procédure, se lance au tout début afin d'initialiser les chunks. Elle récupère les informations envoyées par le programme LISP.

Les procédures de type set\_model et set\_accdt servent pour l'enregistrement des données dans les chunks. Ils sont nombreux comme on a plusieurs chunks, et comme on veut se rapprocher le plus du comportement humain.

Les procédures try\_load, remember\_placement, remember-win, remember-lose-b-soft, remember-lose-b-hard, remember-lose-t-right et doesnt-remember-placement sont utilisées pour se souvenir si une situation similaire a déjà été vécue auparavant. Si l'on se souvient d'une situation de réussite, on applique la même action. Mais, si l'on se souvient d'un échec, on applique une stratégie différente :

- Si l'échec s'est produit lors d'un freinage doux, alors on freine fortement.
- S'il y a eu un échec avec un freinage fort, alors on tourne à droite.
- Si l'échec s'est produit au cours d'un changement de voie par la droite, alors on tourne à gauche.

Si l'on n'a jamais vécu de situation similaire, on freine doucement.

La procédure remember-placement s'active si la requête agit favorablement et que le modèle s'est rappelé d'une situation. Elle met dans les slots `result` et `action` les actions correspondantes aux souvenirs.

La procédure doesnt-remember-placement se déclenche si la requête échoue. Comme le modèle n'a pas pu se rappeler d'une situation similaire, il va freiner doucement.

La procédure remember-win s'active si la variable result envoyée par la procédure remember-placement est de type "esquive". Elle fait appliquer le résultat mémorisé.

Les procédures remember-lose-b-soft, remember-lose-b-hard, remember-lose-t-right et remember-lose-t-left s'enclenchent lorsque la variable result est "crash" et que l'action souvenue est respectivement 1,2 et 3. Elles renvoient les actions 2,3 et 4.

La procédure brakeSoft ralentit la voiture lentement pour satisfaire le but "freiner" lorsque l'action est 1.

La procédure brakeHard ralentit la voiture rapidement et de façon brusque afin de satisfaire le but "freiner" lorsque l'action est 2.

La procédure turnR tourne le véhicule à droite afin qu'il évite l'accident lorsque l'action est 3.

La procédure turnL tourne le véhicule à gauche afin qu'il évite l'accident lorsque l'action est 4.

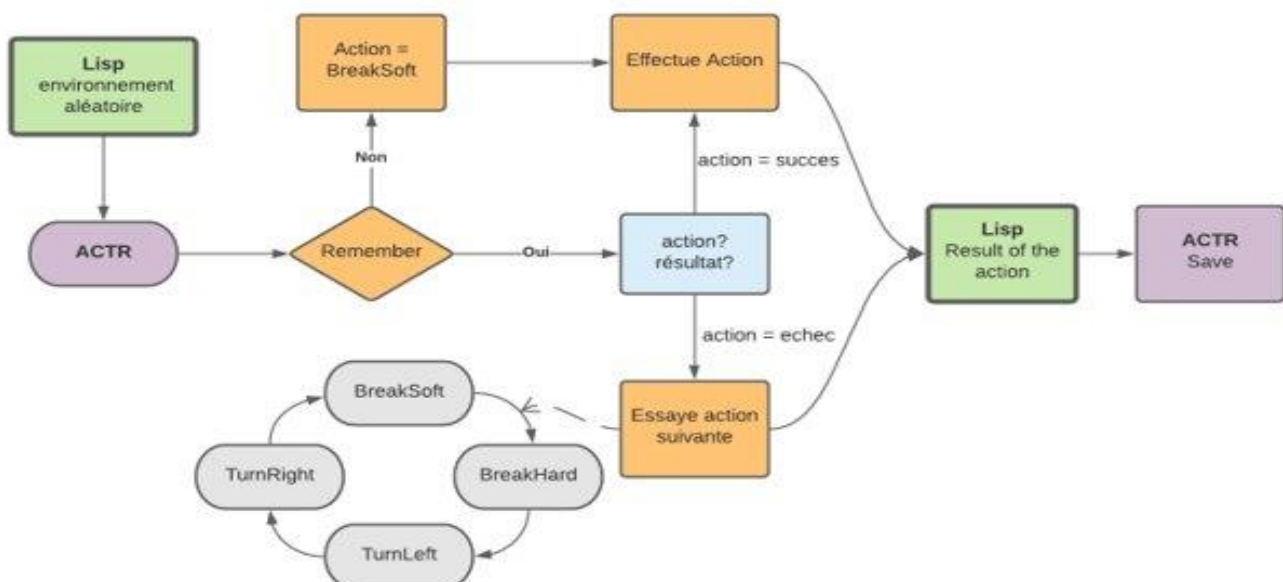
Les procédures start-save et save permettent de sauvegarder les actions effectuées dans le buffer imaginal.

Les procédures finish\_saving et clear\_goal permettent d'effacer le buffer imaginal afin que les nouvelles connaissances s'ajoutent à la mémoire déclarative.

## Scénario

### Déroulement

Un conducteur conduit une voiture sur une autoroute. Un accident survient devant lui.



Voici comment le modèle prend une décision :

1. Il cherche dans sa mémoire si une situation similaire a déjà été rencontrée. Si aucune situation similaire n'a été sauvegardée, alors le modèle freine doucement et le résultat de cette action est sauvegardé.
2. Si une situation similaire existe et que l'action effectuée a amené à une réussite, le modèle répète cette action et le résultat est sauvegardé.
3. Si une situation similaire existe et que l'action effectuée a amené à un échec, l'action suivante est choisie comme suit :
  - a. Si l'échec s'est produit pour un freinage doux, on freine fortement.
  - b. S'il y a eu un échec avec un freinage fort, on tourne à gauche.
  - c. Si l'échec s'est produit lors d'un changement de voie par la gauche, on tourne à droite.

## Trace complète

La commande pour lancer x fois le modèle est (place\_elements x).

Les images suivantes montrent la trace d'exécution de notre modèle après avoir lancé la commande (place\_elements 1). Le déroulement détaillé avec le stepper se trouve dans notre présentation PowerPoint de la phase 3.



```
(place_elements 1)
```

```

^   ^   ^   ^
|   |   |   |
|   A   |   |   avec position Y = 2
|   M   |   |   avec position Y = 1, vitesse = 3, poids = 1
|   |   |   |

```

On est dans la boucle 1 fois

0.000	GOAL	SET-BUFFER-CHUNK GOAL CHECK-STATE0 NIL	
0.000	PROCEDURAL	CONFLICT-RESOLUTION	
0.050	PROCEDURAL	PRODUCTION-FIRED START	
0.050	PROCEDURAL	CONFLICT-RESOLUTION	
0.100	PROCEDURAL	PRODUCTION-FIRED SET_MODEL_0	← Enregistrement du poids du modele
0.100	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
0.100	PROCEDURAL	CONFLICT-RESOLUTION	
0.300	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK0	
0.300	PROCEDURAL	CONFLICT-RESOLUTION	
0.350	PROCEDURAL	PRODUCTION-FIRED SET_MODEL_1	← Enregistrement de la position du modele
0.350	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
0.350	PROCEDURAL	CONFLICT-RESOLUTION	
0.550	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK1	
0.550	PROCEDURAL	CONFLICT-RESOLUTION	
0.600	PROCEDURAL	PRODUCTION-FIRED SET_MODEL_2	← Enregistrement de la vitesse du modele
0.600	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
0.600	PROCEDURAL	CONFLICT-RESOLUTION	
0.800	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK2	
0.800	PROCEDURAL	CONFLICT-RESOLUTION	
0.850	PROCEDURAL	PRODUCTION-FIRED SET_ACCDT_1	← Enregistrement de la position de l'accident
0.850	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
0.850	PROCEDURAL	CONFLICT-RESOLUTION	
1.050	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK3	
1.050	PROCEDURAL	CONFLICT-RESOLUTION	
1.100	PROCEDURAL	PRODUCTION-FIRED SET_ACCDT_2	← Enregistrement de la vitesse de l'accident
1.100	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
1.100	PROCEDURAL	CONFLICT-RESOLUTION	
1.300	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK4	
1.300	PROCEDURAL	CONFLICT-RESOLUTION	
1.350	PROCEDURAL	PRODUCTION-FIRED TRY_LOAD	← Lance le retrieval pour se remémorer
1.350	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
1.350	PROCEDURAL	CLEAR-BUFFER RETRIEVAL	
1.350	DECLARATIVE	start-retrieval	
1.350	PROCEDURAL	CONFLICT-RESOLUTION	
1.750	DECLARATIVE	RETRIEVAL-FAILURE	← Cette situation de s'est jamais présentée
1.750	PROCEDURAL	CONFLICT-RESOLUTION	
1.800	PROCEDURAL	PRODUCTION-FIRED DOESNT-REMEMBER-PLACEMENT	←
1.800	PROCEDURAL	CLEAR-BUFFER RETRIEVAL	
1.800	PROCEDURAL	CONFLICT-RESOLUTION	
1.850	PROCEDURAL	PRODUCTION-FIRED BRAKESOFF	← Le modèle choisi de freiner doucement
1.850	PROCEDURAL	CLEAR-BUFFER MANUAL	
1.850	MOTOR	PRESS-KEY KEY 1	←
1.850	PROCEDURAL	CONFLICT-RESOLUTION	
2.100	PROCEDURAL	CONFLICT-RESOLUTION	
2.150	PROCEDURAL	CONFLICT-RESOLUTION	
2.250	PROCEDURAL	CONFLICT-RESOLUTION	
2.400	PROCEDURAL	CONFLICT-RESOLUTION	
10.000	-----	Stopped because time limit reached	
Le modele a choisi "1" ← Le programme LISP a bien reçu le choix du modele			
10.000	GOAL	GOAL-MODIFICATION	← Le programme LISP renvoie le résultat de l'action
10.000	PROCEDURAL	CONFLICT-RESOLUTION	
10.050	PROCEDURAL	PRODUCTION-FIRED START-SAVE	←
10.050	PROCEDURAL	CONFLICT-RESOLUTION	
10.100	PROCEDURAL	PRODUCTION-FIRED SAVE	
10.100	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
10.100	PROCEDURAL	CONFLICT-RESOLUTION	
10.150	PROCEDURAL	PRODUCTION-FIRED FINISH_SAVING	← Le modele enregistre la situation, son choix ainsi que le résultat
10.150	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
10.150	PROCEDURAL	CONFLICT-RESOLUTION	
10.300	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK5	
10.300	PROCEDURAL	CONFLICT-RESOLUTION	
20.000	-----	Stopped because time limit reached	

On est dans la boucle 2 fois			
20.000	GOAL	GOAL-MODIFICATION	
20.000	PROCEDURAL	CONFLICT-RESOLUTION	
20.050	PROCEDURAL	PRODUCTION-FIRED TRY_LOAD	
20.050	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
20.050	PROCEDURAL	CLEAR-BUFFER RETRIEVAL	
20.050	DECLARATIVE	start-retrieval	
20.050	PROCEDURAL	CONFLICT-RESOLUTION	
20.090	DECLARATIVE	RETRIEVED-CHUNK CHUNK5-0	
20.090	DECLARATIVE	SET-BUFFER-CHUNK RETRIEVAL CHUNK5-0	
20.090	PROCEDURAL	CONFLICT-RESOLUTION	
20.140	PROCEDURAL	PRODUCTION-FIRED REMEMBER-PLACEMENT	
20.140	PROCEDURAL	CLEAR-BUFFER RETRIEVAL	
20.140	PROCEDURAL	CONFLICT-RESOLUTION	
20.190	PROCEDURAL	PRODUCTION-FIRED REMEMBER-LOSE-B-SOFT	
20.190	PROCEDURAL	CONFLICT-RESOLUTION	
20.240	PROCEDURAL	PRODUCTION-FIRED BRAKEHARD	
20.240	PROCEDURAL	CLEAR-BUFFER MANUAL	
20.240	MOTOR	PRESS-KEY KEY 2	
20.240	PROCEDURAL	CONFLICT-RESOLUTION	
20.390	PROCEDURAL	CONFLICT-RESOLUTION	
20.440	PROCEDURAL	CONFLICT-RESOLUTION	
20.540	PROCEDURAL	CONFLICT-RESOLUTION	
20.690	PROCEDURAL	CONFLICT-RESOLUTION	
30.000	-----	Stopped because time limit reached	
Le modele a choisi "2"			
30.000	GOAL	GOAL-MODIFICATION	
30.000	PROCEDURAL	CONFLICT-RESOLUTION	
30.050	PROCEDURAL	PRODUCTION-FIRED START-SAVE	
30.050	PROCEDURAL	CONFLICT-RESOLUTION	
30.100	PROCEDURAL	PRODUCTION-FIRED SAVE	
30.100	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
30.100	PROCEDURAL	CONFLICT-RESOLUTION	
30.150	PROCEDURAL	PRODUCTION-FIRED FINISH_SAVING	
30.150	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
30.150	PROCEDURAL	CONFLICT-RESOLUTION	
30.300	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK6	
30.300	PROCEDURAL	CONFLICT-RESOLUTION	
40.000	-----	Stopped because time limit reached	

Le programme LISP annonce au modele de recommencer car il a crash précédemment

Le modele regarde si cette situation s'est déjà présentée (oui le tour précédent)

Le modele se souvient qu'il a crash précédemment, il tente donc une autre approche : freiner fort

Le programme LISP a bien reçu le choix

Le modele LISP renvoie le résultat de l'action

Le modele enregistre la situation, son choix ainsi que le résultat

On est dans la boucle 3 fois			
40.000	GOAL	GOAL-MODIFICATION	
40.000	PROCEDURAL	CONFLICT-RESOLUTION	
40.050	PROCEDURAL	PRODUCTION-FIRED TRY_LOAD	
40.050	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
40.050	PROCEDURAL	CLEAR-BUFFER RETRIEVAL	
40.050	DECLARATIVE	start-retrieval	
40.050	PROCEDURAL	CONFLICT-RESOLUTION	
40.080	DECLARATIVE	RETRIEVED-CHUNK CHUNK6-0	
40.080	DECLARATIVE	SET-BUFFER-CHUNK RETRIEVAL CHUNK6-0	
40.080	PROCEDURAL	CONFLICT-RESOLUTION	
40.130	PROCEDURAL	PRODUCTION-FIRED REMEMBER-PLACEMENT	
40.130	PROCEDURAL	CLEAR-BUFFER RETRIEVAL	
40.130	PROCEDURAL	CONFLICT-RESOLUTION	
40.180	PROCEDURAL	PRODUCTION-FIRED REMEMBER-LOSE-B-HARD	
40.180	PROCEDURAL	CONFLICT-RESOLUTION	
40.230	PROCEDURAL	PRODUCTION-FIRED TURNR	
40.230	PROCEDURAL	CLEAR-BUFFER MANUAL	
40.230	MOTOR	PRESS-KEY KEY 3	
40.230	PROCEDURAL	CONFLICT-RESOLUTION	
40.380	PROCEDURAL	CONFLICT-RESOLUTION	
40.430	PROCEDURAL	CONFLICT-RESOLUTION	
40.530	PROCEDURAL	CONFLICT-RESOLUTION	
40.680	PROCEDURAL	CONFLICT-RESOLUTION	
50.000	-----	Stopped because time limit reached	
Le modele a choisi "3"			
50.000	GOAL	GOAL-MODIFICATION	
50.000	PROCEDURAL	CONFLICT-RESOLUTION	
50.050	PROCEDURAL	PRODUCTION-FIRED START-SAVE	
50.050	PROCEDURAL	CONFLICT-RESOLUTION	
50.100	PROCEDURAL	PRODUCTION-FIRED SAVE	
50.100	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
50.100	PROCEDURAL	CONFLICT-RESOLUTION	
50.150	PROCEDURAL	PRODUCTION-FIRED FINISH_SAVING	
50.150	PROCEDURAL	CLEAR-BUFFER IMAGINAL	
50.150	PROCEDURAL	CONFLICT-RESOLUTION	
50.200	PROCEDURAL	PRODUCTION-FIRED CLEAR-GOAL	
50.200	PROCEDURAL	CLEAR-BUFFER GOAL	
50.200	PROCEDURAL	CONFLICT-RESOLUTION	
50.300	IMAGINAL	SET-BUFFER-CHUNK IMAGINAL CHUNK7	
50.300	PROCEDURAL	CONFLICT-RESOLUTION	
60.000	-----	Stopped because time limit reached	

Une fois de plus le programme LISP fait fonctionner le modele

Le modele regarde si cette situation s'est déjà présentée (oui 2x, il essaie de se souvenir du résultat le plus récente)

Le modele se souvient qu'il a crash en freinant fort et tente une autre approche : tourner à droite

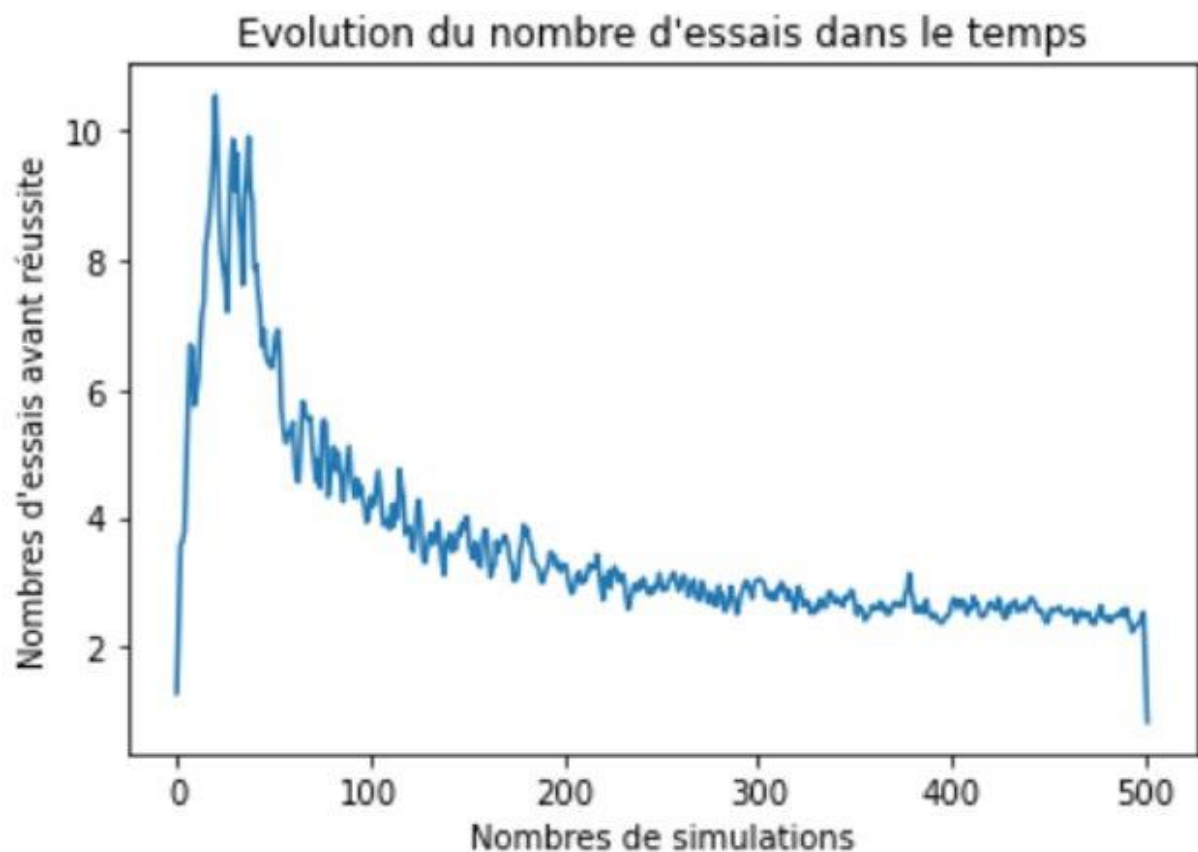
Le programme LISP a calculé que le modele a évité l'accident et ordonne une fois de plus la sauvegarde avec le résultat

Le modele enregistre la situation, son choix ainsi que le résultat

Comme le modele a esquivé l'accident et qu'on passe à la simulation suivante, on clear le goal

## Apprentissage

Graphe d'apprentissage avec optimisation des paramètres



Cette courbe a été obtenue sur Matplotlib. Ce graphique a été réalisé en important les données de 25 sessions de test sur une longueur de 500 simulations dans un programme d'affichage de graphique en python. La première valeur et la toute dernière qui sont au niveau du 0 sont juste la façon dont le graphe est représenté, ce ne sont pas des valeurs réelles.

Les paramètres d'optimisation choisis sont :

```
(sgp :v nil :esc t :lf 0.4 :bll 0.45 :ans 0.6 :rt 0 :ncnar nil)
```

On peut constater que le nombre d'essais avant réussite diminue de 10 à 3 environ.

## Interprétation des résultats des simulations

On remarque donc sur ce graphique que le modèle apprend à partir de 50 essais. On peut également remarquer un important pic au début, au moment où le modèle prend de plus en plus d'essais avant de réussir. Lors des dix premiers essais, il y a une augmentation puisque peu de chunks sont créés. Ainsi, le modèle arrive relativement à se souvenir des bonnes actions à réaliser. Par la suite, le nombre de cas différents rencontrés augmente plus vite que l'apprentissage, le modèle se souvient alors d'anciennes actions qui se sont soldées par un échec au lieu de se souvenir de la bonne action qu'il a peut-être déjà réalisée. Le modèle perd donc du temps en augmentant le nombre d'essais. Enfin, arrivé à un certain point, le modèle commence à se souvenir de plus en plus facilement des bonnes actions. Alors l'apprentissage se renforce et, en même temps, le nombre de cas observés augmente vers le nombre de cas possible maximum et toutes les situations possibles sont alors rencontrées.

En conclusion, notre modèle ne s'améliore plus après avoir atteint les trois essais avant de réussir, pour 500 itérations. Cela peut être représentatif d'un être humain qui ne peut pas se rappeler de tous les scénarios qu'il a déjà rencontrés et qui peut donc parfois se tromper (dans le cas où une personne a déjà subi plusieurs accidents).

## Ce que nous avons essayé d'implémenter

### Une complexification de notre modèle

Notre modèle apprenant correctement, nous avons essayé d'implémenter de nouveaux éléments, notamment un véhicule derrière notre modèle (aléatoirement sur une des trois voies), qui va représenter un usager supplémentaire. Un exemple est présenté dans l'infographie ci-dessous.

Si le modèle freine alors que le troisième usager est juste derrière notre modèle, nous considérons cela comme un crash. C'est le cas d'une voiture qui ne respecterait pas la distance de sécurité. Le seul moyen de sortir de l'accident est donc de tourner. Cependant, notre modèle n'arrivait pas à apprendre et l'implémentation n'était pas totalement finie. Nous n'avons donc pas présenté ce modèle non-fini et avons préféré présenter un modèle, certes plus simple, mais fonctionnel.

Les parties qui considèrent le troisième véhicule que nous avons enlevé sont les dernières lignes commentées à la fin du code ACT-R dans le fichier Lisp remis.

## Conclusion

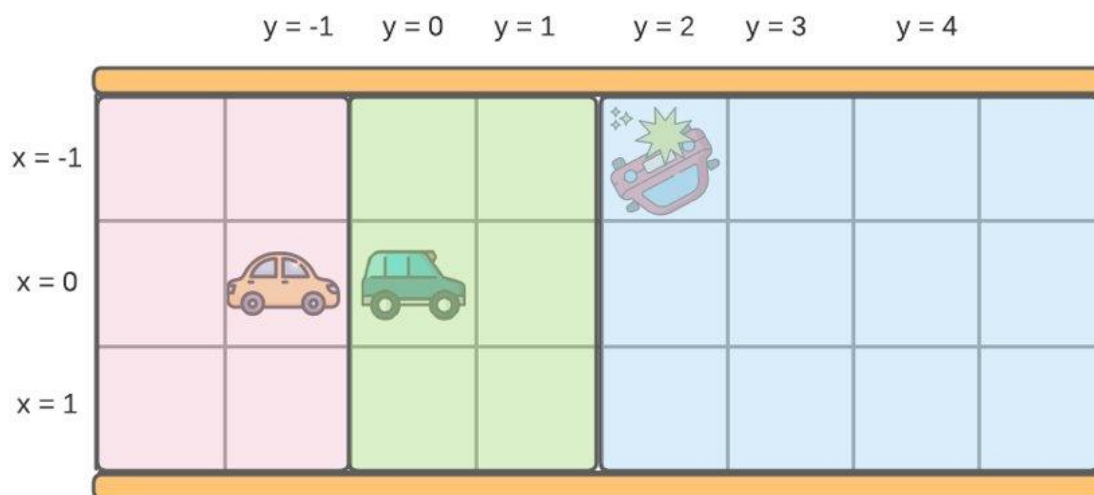
### Limites

- Le nombre de types de véhicules est limité à deux, voiture et camion.
- Dans le cas où un accident se produit devant notre conducteur pour la première fois, il va freiner doucement. Cependant, il serait peut-être plus humain de freiner brusquement dans la panique.
- Notre représentation de la conduite sur autoroute est simplifiée : il n'y a qu'un seul véhicule accidenté et seulement deux usagers sur la route.
- Le changement de voie et le freinage sont grandement simplifiés à cause de la modélisation de la route sous forme de cases.

Pour des raisons de simplification, nous avons choisi d'appliquer les limites présentées précédemment. Il pourrait y avoir un grand potentiel de complexification de notre modèle, en ajoutant notamment des usagers autour de notre véhicule.

### Améliorations

- Finaliser d'implémenter l'utilisateur derrière notre modèle
- Avoir un nombre aléatoire d'utilisateurs sur l'autoroute, borné par un nombre minimal et un nombre maximal.
- Ajouter une procédure qui pourrait vérifier si un utilisateur se trouve à droite ou à gauche avant de tourner.



- Ajouter des dimensions de véhicules.
- Considérer le nombre d'années d'expérience du conducteur et son état (fatigue, alcool).

## Possibles extensions

- Optimisation des paramètres pour l'apprentissage.
- Génération automatique des graphes d'apprentissage pour faciliter le suivi de l'apprentissage.
- Analyse des différentes phases des graphes d'apprentissage dans le but d'améliorer notre modèle.
- Permettre différents types de véhicule et ajouter des remorques à certains afin de ne pas avoir un véhicule paramétré parfaitement dans une case.
- Considérer la gravité de l'accident : effleurer le véhicule accidenté est par exemple moins grave que le heurter.
- Avoir un nombre de voies variables à cause de travaux ou d'une voie fermée par exemple, ou à cause de l'ajout d'une voie d'arrêt d'urgence.
- Prendre en compte la météo pour l'état de la route (pluie, neige ou brouillard).

## Remerciement

Nous tenons à exprimer nos remerciements à M. Mohamed Mehdi Najjar pour la qualité de son enseignement et pour les directives qu'il nous a fournies pendant la durée du projet.