Embouteillages

Projet informatique

Rapport 1er partie

Vellu Clément – Quici Mipam

Table des matières

Introduction

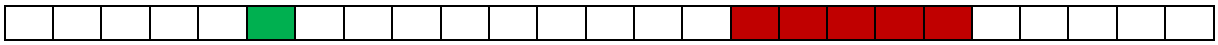
Les embouteillages sont des phénomènes ennuyants que l'on rencontre très couramment sur nos routes et peuvent, dans certains cas, prendre des proportions gigantesques.

Le but de ce projet est donc de modéliser une route avec différents véhicules dessus. La simulation devra, en laissant évoluer assez de véhicules, créer un embouteillage et montrer en même temps l'effet « accordéon » dans le mouvement de masse des véhicules. Il devra aussi être possible de déclencher un embouteillage en insérant un élément perturbateur, mais aussi de trouver des méthodes de résolution de ceux-ci, notamment en permettant aux voitures de changer de voies.

La simulation se basera dans un premier temps sur des modèles simplifiés de la physique, avec une modélisation particulière de la vitesse et de la position des véhicules par exemple, avant d'améliorer le modèle pour qu'il soit plus fiable et proche de la réalité.

1 Pistes et hypothèses

Nous avons envisagé dans un premier temps de modéliser la route par une liste, mais avons retenu l'utilisation des tableau numpy. Les tableaux et l'affectation d'une seule position par véhicule discrétise leur position, éloignant ainsi le modèle de la réalité. Nous avons utilisé ce modèle-ci pour faciliter la création et le fonctionnement de la simulation. Une des pistes à envisager serai d'utiliser les positions du tableau comme unité de mesures. Un véhicule prendrait par exemple plusieurs cellules du tableau et notre simulation gagnerai en précision.



Nous pouvons bien distinguer les cellules du tableau sur le schéma de la route ci-dessus. La modélisation actuelle des véhicules est celle de la case verte, chaque véhicule occupe une seule case. Les 5 cases rouges représentent aussi un seul véhicule mais le modèle représentant la position du véhicule est plus précis.

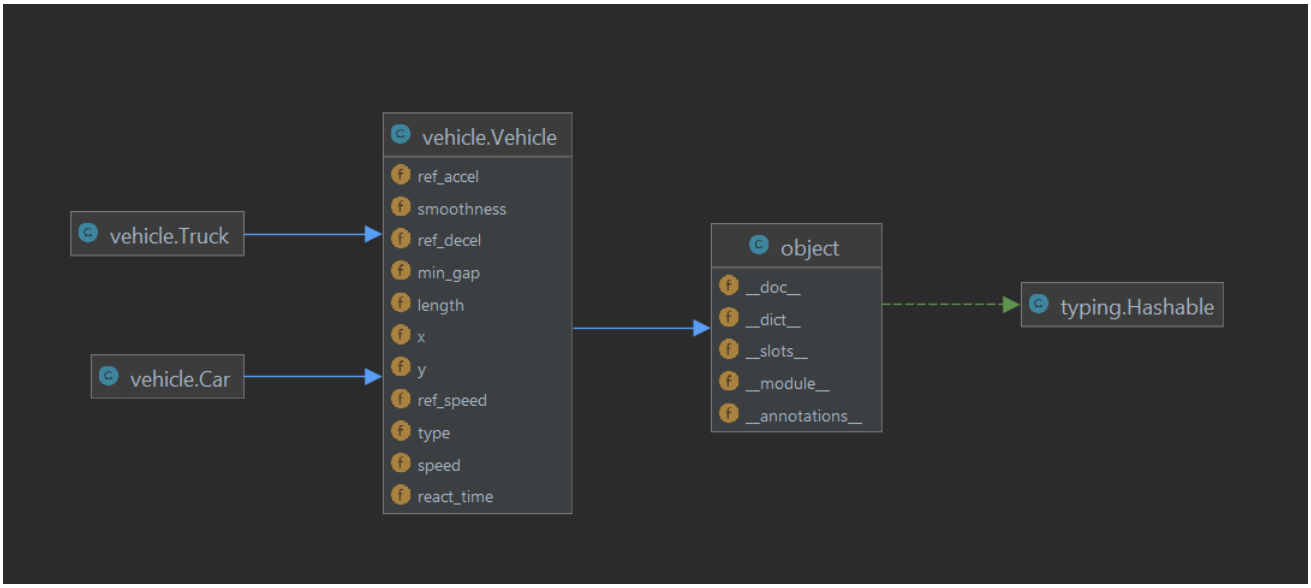
2 Présentation du programme

2.1 Description des classes

2.1.1 La classe Vehicle

La classe Vehicle recense les différents véhicules présents, leurs caractéristiques et comportement sur la route. Elle possède deux sous-classes pour les deux types de véhicules : car et truck.

Ci-dessous le diagramme de classe de Vehicle :



La méthode get_gap :

Cette méthode prend en paramètre la route et un véhicule et renvoie la distance séparant les deux.

La méthode calculate_accel :

Cette méthode permet de calculer l'accélération d'un véhicule avec un modèle physique, l'approximation ballistique. Cette accélération sert ensuite à calculer la nouvelle vitesse du véhicule.

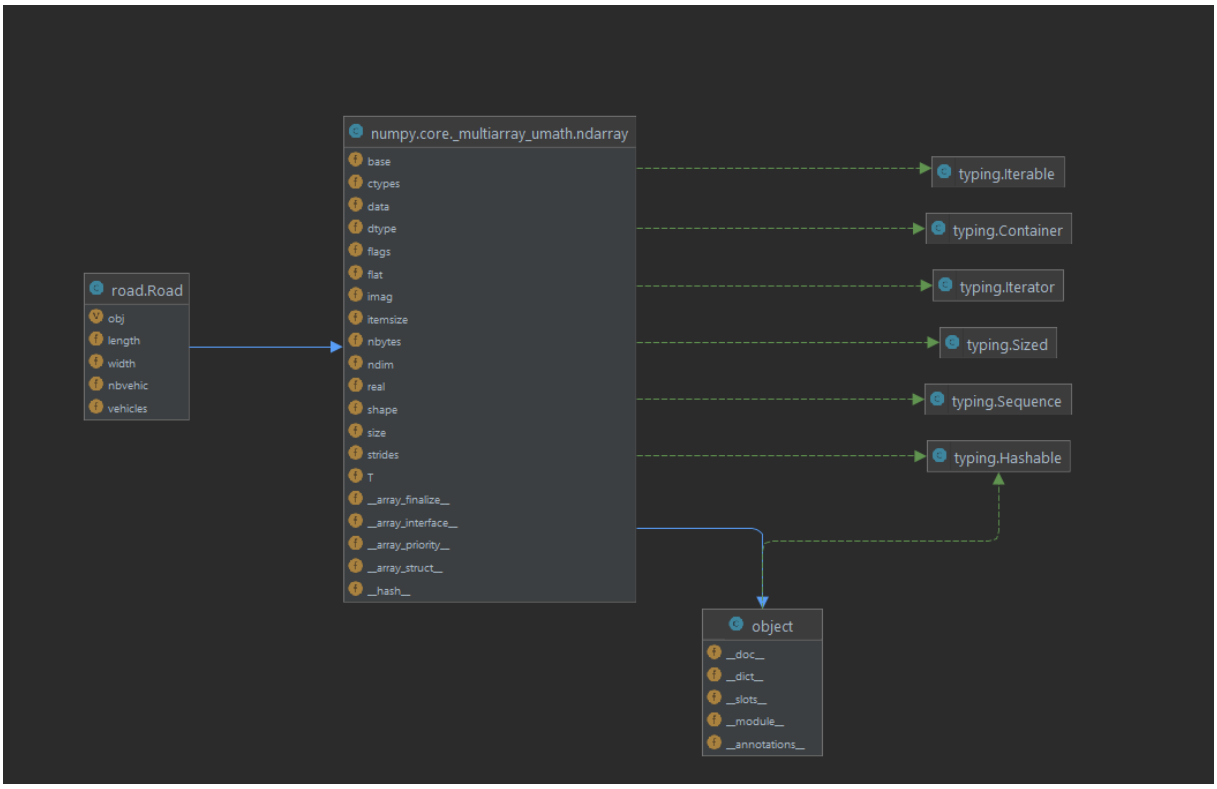
La méthode change_lanes :

Cette méthode prendra en paramètre la position du véhicule et la destination voulue et fera changer de voie le véhicule.

2.1.2 La classe Road

La classe road est l'écosystème qui modélise la route sur laquelle les véhicule vont évoluer, elle hérite d'un tableau de numpy. La taille de la route, longueur et nombres de voies, ainsi que le nombre de véhicules présents y sont renseignés.

Ci-dessous le diagramme de classe de Road :



La méthode place_vehicles_randomly :

Cette méthode permet de positionner les véhicules sur la route de manière aléatoire au tout début de la simulation.

La méthode __array_finalize__ :

Cette méthode permet, lorsqu'un nouvel objet est créé lors de la manipulation des ndarrays, de s'assurer que ce nouvel objet, qui devient l'objet principal, reçoive aussi les variables d'instances.

La méthode __str__ :

La réécriture de cette méthode permet de représenter la route sous forme de caractères. Dans une première modélisation nous allons utiliser les symboles suivants : les « _ » représentent la route vide, « C » représentent les voitures et « T » les camions.

3 Figures imposées

3.1 Héritage entre deux types créés

Nous avons choisi d'utiliser deux types de véhicules dans notre simulation : les voitures et les camions. Ainsi les classes Car et Truck héritent de notre classe Vehicule.

3.2 Héritage depuis un type intégré

Notre classe Road, qui modélise la route, hérite d'un ndarray numpy.

3.3 Fonction récursive

La méthode place_vehicles_randomly de la classe Road sera modifiée en une fonction récursive.

4 Tests unitaires

4.1 TestRoad

Cette classe de tests contient deux méthodes : testInit et testInitExcept. TestInit sert à tester les variables d'instances et testInitExcept sert à tester le déclenchement des exceptions dans le init.

4.2 TestVehicle

Cette classe de tests ne contient pour l'instant qu'une seule méthode : testInit qui permet de tester les variables d'instances.

5 Limites et amélioration

Pour l'instant notre autoroute ne possède pas d'entrées ni de sorties, les voitures présentes sur la route sont en nombre fixe. Une des améliorations à faire pour se rapprocher de la réalité serai donc de rajouter des entrées et des sorties.