Diagram

Description automatically generated**Simulation**

**Contrôle lumières:**

L’intelligence artificielles que nous avons fait se sert des données de la simulation pour décider de laisser passer certaines voitures au lieu de d’autres. C’est pourquoi il fallait que notre simulation soit la plus robuste possible, en ajoutant des éléments aléatoires pour s’assurer que les données varient assez.

Pour s’assurer que les voitures arrêtent aux lumières rouges et avance quand elles sont vertes, il y a, à chaque lumière, une zone de déclenchement qui permet au IA d’indiquer aux voitures d’arrêter ou de continuer. Si cette zone est active, la voiture qui y fait contact arrête et fait en sorte que les voitures la suivant arrêtes aussi. Lorsque la zone de déclenchement est désactivée par la IA, la voiture y faisant contact se remet à accélérer et les autres arrêtées à l’arrière suivent.

La couleur des lumières est simplement une indication visuelle pour l’utilisateur. Les voitures et la IA ne ce servent pas des couleurs, seulement des zones de déclenchement. Cependant, l’utilisation de la lumière jaune permet de nous assurer que les voitures, étant encore dans l’intersection lors du changement de lumière par l’IA, aient le temps de passer l’intersection avant de changer pour une lumière verte. Ceci est réaliser en ajoutant un délai entre chaque action du IA.

**Voitures :**

A picture containing graphical user interface

Description automatically generatedLes voitures sont programmées pour suivre un trajet qui peut changer lorsqu’elles rencontrent une intersection. Aucune voiture à un trajet prédéfinit, donc la IA ne peu pas s’avoir d’avance où chaque voiture s’en va. Les autos ont simplement un point de cheminement à atteindre, une fois celui-ci atteint, on lui en assigne un nouveau pour qu’elle puisse continuer son chemin. Si, à un certain point de cheminement, plusieurs options sont possibles, la voiture choisis au hasard un des points suivants.

* *On aurais pu modifier les chances pour chaque waypoint, comme ca les petites rues sont moins utilisé.*

Pour s’assurer qu’il y aille du trafic fluide et constant, nous avons créé un algorithme pour gérer les points d’apparitions des voitures. Lorsqu’il y a moins de 50 voitures sur la simulation, nous faisons apparaitre de 1 à 10 voitures simultanément selon 9 points d’apparitions différents. Puisque plusieurs voitures peuvent apparaitre au même endroit, nous devons nous assurer qu’aucune voiture ne sois une par-dessus l’autre. Pour ce faire, nous devons déplacer les doublons d’une longueur de voiture selon son axe des ordonnées.

Lorsqu’une voiture atteint la fin de son chemin, nous devons détruire cette voiture pour pouvoir en ajouter d’autre. Pour ce faire, nous supprimons tous les voitures qui non plus de point de cheminement à suivre. Les points de cheminements mènent tous à l’extérieur de la zone de simulation pour qu’aucune voiture ne disparaisse visuellement dans la simulation. Au moment de la suppression, la IA est informée du nombre de temps que cette voiture a arrêter et est récompensé en conséquence. \*Pour plus d’info sur les récompenses, voir XXX

A picture containing text, indoor

Description automatically generatedLes voitures, qui ne sont pas des IA en tant que tel, doivent pouvoir ralentir, arrêter et accélérer quand il est nécessaire. Nous avons donc ajouté 3 capteurs de collisions sur tous les voitures. Ceux-ci regardent la vitesse, l’accélération ainsi que la distance entre une voiture et celle en avant d’elle. Il est possible de visualiser les capteurs dans le mode scène de Unity. Chaque capteur fonctionne indépendamment les uns des autres, mais ils ont tous les mêmes fonctions. Dans un premier temps, si le capteur détecte une voiture, elle ajuste la vitesse de cette voiture pour qu’il n’y aille pas de collision. De plus, quand le capteur détecte que la voiture à l’avant est trop proche, elle va ralentir rapidement la vitesse de celle-ci ou, subitement, arrêter la voiture pour éviter une collision. Les capteurs de gauche et de droite servent surtout pour ne pas faire de collision avec des voitures qui ne sont pas directement en avant de celle-ci.

Dans notre projet, nous avons encore des problèmes de collision quand il y a du trafic qui s’accumule. Cela est probablement dû au fait que nos voitures fonctionnent avec un système de physique, ce qui rend la tâche d’arrêter et de ralentir plus complexe et ainsi semble faire que certaine voiture arrête et d’autre ne font que ralentir.