

ROAD SIMULATOR 2014

1. Structure du programme

Le programme a été pensé comme requis pour utiliser des types abstraits. Ainsi, il est décomposé en un type route et un type voiture, chacun gérant plusieurs aspects du code. La route est ainsi le conteneur des voitures et permet de manipuler toutes ces dernières en même temps. Elle contient de plus toutes les limitations intrinsèques de la simulation (type de freinage/vmax/probabilités p et q). Le type voiture gère quant à lui les quatre étapes de la simulation, en se basant sur les informations provenant du Type route.

Ainsi, le programme se décompose en 5 fichiers :

- ❑ route.h et route.cpp, qui déclarent/documentent et décrivent respectivement les fonctions afférentes au type route.
- ❑ voiture.h et voiture.cpp, qui déclarent/documentent et décrivent respectivement les fonctions afférentes au type voiture.
- ❑ main.cpp qui met en place l'environnement et génère une route.

2. Déroulement du programme.

2.1. Main.cpp

A l'ouverture du programme, un timer est lancé, celui-ci remplace avantageusement le système de "tours" permettant de voir progresser la simulation en temps réel, selon un tempo contrôlé par l'utilisateur. Une fois ce timer déclaré, un objet de type Route est créé, avec des paramètres par défaut.

On écoute à chaque cycle pour une pression de touche de la part de l'utilisateur, réagissant de façon appropriée. Puis, on met à jour la route courante, ce qui a pour effet de la mettre à jour.

2.2. Route+ Voiture.

La mise à jour de la route se fait de façon très terre à terre. Outre un appui de touche modifiant une variable interne de l'objet, la seule opération à faire consiste à itérer un vecteur interne de voitures "m_road", et à lancer la fonction maj_voiture() sur chacun de ces objets.

Cette fonction va en fait en appeler quatre autres, chacune responsable d'une étape (telle que décrite dans le projet) de mise à jour. Chaque voiture va donc se mettre à jour indépendamment des autres, en suivant un protocole dépendant néanmoins d'une autre voiture. Ainsi, à chaque fois, la voiture accélère d'abord d'une unité, puis vérifie si un freinage d'urgence doit être fait pour éviter une collision, puis est affecté une chance aléatoire de freiner.

Enfin, quand toutes ces étapes ont été réalisées, la position de la voiture sur la route est mise à jour. Cette position est en fait la distance totale parcourue par la voiture, permettant ainsi de faire varier la taille de la route à volonté (non inclus dans le programme)

3. Possibilité d'extension.

Il est possible, grâce au modèle de développement choisi d'implémenter très facilement plus d'une route et de faire passer une voiture de l'une à l'autre, tout simplement en déplaçant l'objet désiré d'un vecteur à l'autre.

De plus, on peut imaginer différents types de véhicules, chacun avec sa propre chance de freiner et sa propre vitesse maximale.