

# Java : Sujet de projet - JTrafic

27 mars 2019 - 25 mai 2018

## 1 Contexte

On souhaite développer un simulateur de trafic routier. Le principe est de pouvoir simuler, sur un réseau routier donné, un certain nombre de véhicules dotés de vitesses variées. En fonction de la longueur des trajets, des feux, carrefours et intersections rencontrés. La finalité d'un tel projet est de pouvoir étudier la formation de bouchons ou de noeuds de trafic en fonction du trafic généré.

## 2 Détails

Votre modèle devra respecter les contraintes suivantes :

- On considèrera un réseau routier comme un graphe reliant différents points (les villes). Deux villes peuvent être reliées par plusieurs routes.
- Chaque route comporte 2, 4 ou 6 voies (route, nationale ou autoroute).
- Chaque véhicule doit ralentir à l'approche d'un feu, d'un croisement ou encore d'un autre véhicule ou obstacle.
- Lors d'un stop ou d'un feu rouge un véhicule doit s'arrêter (au moins 1 seconde, ou jusqu'à ce que le feu passe au vert).
- Le trafic doit être généré aléatoirement dans chaque ville (chaque véhicule doit avoir une destination donnée).
- Si un véhicule roule plus vite qu'un autre, il lui est possible de doubler si une voie est libre.
- Chaque véhicule accélère naturellement jusqu'à une vitesse limite (définie par le type de route). Dès l'approche d'un obstacle (autre véhicule, feu, croisement, ...) sa vitesse doit décroître progressivement.

## 3 Fonctionnalités

- Le programme devra permettre un affichage graphique du réseau routier complet ou de zoomer sur certaines parties
- Le programme devra pouvoir simuler le trafic selon différents paramètres : nombre de véhicules moyens par minute depuis chaque ville, vitesse moyenne des véhicules

- Le réseau devra comporter au moins 6 villes, toutes reliées entre elles par minimum 2 routes de types différents. Si il y a croisement entre routes d'importances différentes, la route la moins importante doit comporter un stop. En cas d'importance équivalente, la règle de la priorité à droite doit s'appliquer.
- Une fois les paramètres d'entrée définis, le système doit pouvoir fonctionner de manière autonome pendant au moins plusieurs minutes.

## 4 Contraintes

- Le projet devra être effectué en trinôme (éventuellement binôme).
- Le projet rendu devra contenir la totalité des classes, fichiers ressources et autres éléments nécessaires au projet (identifiants de connexion, documentation, ...). Le rendu devra également comporter un rapport **au format pdf** détaillant la structure du projet (la structure des packages et le diagramme de classes notamment. Ce diagramme pourra être scindé en plusieurs parties si besoin.)
- Deadline de rendu projet : Le 25/5 à 0h00, à renvoyer par mail à [olivier.cros@yncrea.fr](mailto:olivier.cros@yncrea.fr).
- Le rendu du projet pourra se faire via une archive (zip, rar, tar.gz, ...) ou directement via un lien vers vos dépôts github, bitbucket ou dropbox.
- Le projet **ne doit pas** être rendu via des liens périssables dans le temps (ex : WeTransfer).

## 5 Notation

Lors de la correction, il sera notamment prêté attention aux éléments suivants

- Qualité et structure du code
- Qualité rédactionnelle des rendus
- Précision et stabilité des fonctionnalités implémentées
- Apport ou proposition de fonctionnalités nouvelles ou originales
- Respect des formats et deadlines donnés
- Clarté et cohérence de l'interface graphique