# Sujet

## 10. Simulation de trafic routier \*\*\*

Ce projet consiste à réaliser une application destinée à simuler le trafic routier en un noeud donné du réseau (typiquement un carrefour avec des feux tricolores). Le programme doit permettre après avoir configuré le rythme des feux et les flux entrant pour chacune des voies (nombre de voitures et piétons arrivant sur le carrefour par unité de temps), de visualiser une simulation et d'éditer des statistiques de congestions (combien de temps met une voiture / un piéton à traverser le carrefour). Ainsi, en faisant varier les rythmes des feux tricolores, le programme doit permettre de mettre en évidence un rythme idéal pour diminuer les temps de trajet des utilisateurs.

Pour que la simulation soit réaliste, il conviendra de définir quelques règles :

* Distance minimale de sécurité entre les voitures / piétons;
* Modèle d'accélération et de freinage;
* Vitesse maximale de déplacement.

## Attentes

Carrefour avec feux tricolores

Configuration rythme des feux et flux entrant pour chaque voie

- nombre de voitures arrivant sur le carrefour

- nombre de piétons arrivant sur le carrefour

- rythme des feux tricolores

Règles à définir

- Distance minimale de sécurité entre les voitures / piétons

- Modèle d’accélération et de freinage

- Vitesse maximale de déplacement

Visualiser une simulation

Produire un rapport des statistiques de congestion

- combien de temps met une voiture à traverser le carrefour

- combien de temps met un piéton à traverser le carrefour

--> identifier le rythme idéal

# Exemples

## NetLogo Intersection trafic model

Wilensky, U. (1998). NetLogo Traffic Intersection model. <http://ccl.northwestern.edu/netlogo/models/TrafficIntersection>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

<http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/Sample%20Models/Social%20Science/Unverified/Traffic%20Intersection.nlogo>

## The Traffic Grid Goal model

Wilensky, U. & Rand, W. (2015). Introduction to Agent-Based Modeling: Modeling Natural, Social and Engineered Complex Systems with NetLogo. Cambridge, MA. MIT Press.

<http://www.intro-to-abm.com/>

<http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/IABM%20Textbook/chapter%205/Traffic%20Grid%20Goal.nlogo>

<http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/Sample%20Models/Social%20Science/Traffic%202%20Lanes.nlogo>

<http://www.traffic-simulation.de/>

# Projets en Java

Pour imaginer l’expérience utilisateur et les paramètres de simulation

<https://hooktube.com/watch?v=Sf5xfZAWOgs>

(simulation time step, simulation runtime, grid size, set trafic pattern, set car entry rate, set road lengths, set intersection lengths, set car length, set max car velocity, set car stop distance, set car break distance, set trafic light green times, set trafic light yellow times)

Un mini-projet avec quelques éléments partagés

<https://mantlist11.blogspot.com/2012/01/mini-projet-en-programmation-java.html>

Du code pour un projet similaire

<https://github.com/lelelewy/Traffic-Intersection-Simulation>

# Analyse

## Objectifs

Formaliser une analyse du sujet avec les outils UML.

Mettre en œuvre les concepts de l'orienté objet.

Produire une documentation du code.

## Spécifications fonctionnelles (fonctionnalités à implémenter)

### Ouvrir / Quitter l’application

## Evaluer chaque développement --> diagramme de GANTT

## Hypothèses retenues

# Diagrammes UML

## Logiciel

Star UML (compatible Windows / Linux) : <http://staruml.io/download>

Umbrello (compatible Linux / Windows) : <https://umbrello.kde.org/installation.php>

## Diagramme de cas d’utilisation

Configuration

Visualisation

Production rapport stat

## Diagramme de classes

### Voiture

### Piéton

### Route : englobe les voies de circulation dans les deux sens

### Voie

# Version simple

## Unités d’espace et de temps

L’unité de base est définie par la taille et le déplacement d’un piéton.

En effet, un piéton avance à une vitesse moyenne de 3 km/h, ce qui équivaut à un peu plus de 0,8 m/s. Cette vitesse minimale correspond à un déplacement d’une case par tour dans notre simulation.

**1 case = 0,8 mètre**

**1 tour = 1 seconde**

Ainsi, une cellule correspond à une superficie de 0,8\*0,8 m², soit à peu près l’espace couvert par un piéton sur le trottoir. Une voiture “par défaut” couvre un rectangle de 5\*3 cellules, soit 4\*2,4 mètres environ.

En ville, une voiture est limitée à 50 km/h, soit une avancée de 50/3 = 17 cases en un tour.

Le code de la route donne le calcul suivant pour la distance d’arrêt = distance parcourue pendant le temps de réaction + distance de freinage.  
Temps de réaction = 1 seconde (conducteur en forme, sinon le double voire le triple), soit Dist = Vitesse \* Temps.  
Distance de freinage = Vkmh \* 3/10 (en m/s).

Pour 50 km/h, un arrêt brusque nécessite (50000/3600 + 50\*3/10) \* 3600/3000   
soit 50/3 + 5 \* 3600/1000 = 17 + 18 = 35 cases.

# Idées pour version complète

# Biblio

<https://en.wikipedia.org/wiki/Traffic_simulation>