

Rapport de TX
Réalisation d'un calculateur pour une
simulation de trafic urbain

Semestre de printemps 2005

Cyril Crassin

Sylvain Juge

GI04

Responsable : Olivier Grunder

Sommaire

Sommaire	2
1. Introduction	3
2. Analyse	4
2.1. <i>Structure générale de la simulation</i>	4
2.2. <i>Modélisation Logique</i>	4
Objets Mobiles (Mobile Item)	4
Routes (Roads).....	5
Voies (Lanes).....	6
Connecteurs (Connectors)	7
Carrefours (Crossroads).....	7
2.3. <i>Modélisation physique</i>	8
Simulation de files et lois de poursuite.....	8
Modèle d'interpolation.....	8
3. Réalisation.....	10
3.1. <i>Modélisation initiale</i>	10
3.2. <i>Implémentation effective</i>	10
3.3. <i>Outil de débogage</i>	10
3.4. <i>Construction d'un réseau</i>	10
4. Améliorations et extensions possibles.....	12
Interfaçage avec interface graphique via le réseau	12
Interpolation réaliste.....	12
Stockage des données du réseau.....	12
Outil d'édition du réseau.....	12
5. Annexe : diagramme de classe initial.....	13

1. Introduction

Notre sujet de TX était la création d'une simulation de flux de véhicules qui serait ensuite intégrée dans une simulation 3D temps réel qui devrait servir de support à un projet du laboratoire SET de l'UTBM.

Cette simulation est décomposée en deux éléments logiques, d'une part l'interface graphique en 3D, et d'autre part la simulation elle-même dont nous étions plus particulièrement en charge.

Ces deux parties sont créées de manière distincte et doivent communiquer par le biais de l'interface réseau, ce qui permet de faire évoluer la simulation et la visualisation de manière indépendante en fonction des ressources disponibles sur une ou deux machines.

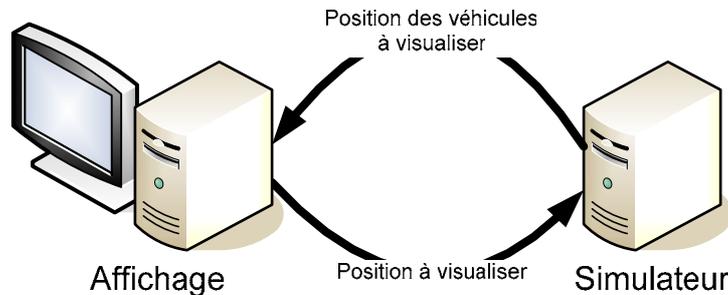
Notre simulateur ayant vocation à servir de base à des projets futurs, nous avons cherché à le rendre le plus efficace possible, tout en gardant une relative flexibilité et une forte adaptabilité pour pouvoir facilement modéliser l'ensemble des situations se présentant sur un réseau urbain réel.

En conséquence, dans ce projet, une grande importance a été donnée à l'analyse, laquelle devrait permettre une grande finesse de modélisation. Nous détaillerons celle-ci dans une première partie puis nous préciserons ensuite les détails de l'implémentation actuelle, et enfin nous détaillerons les points qui mériteraient d'être améliorés dans des évolutions futures.

2. Analyse

2.1. Structure générale de la simulation

Comme nous l'avons brièvement expliqué dans l'introduction, cette simulation est indépendante de la partie affichage, la communication se faisant par le biais de l'interface réseau. On a donc une architecture comme représenté ci-dessous.



Dans un souci de simplification nous avons découpé la simulation en deux sous problèmes : d'une part la simulation d'un point de vue logique qui correspond à la navigation dans un réseau urbain, les interconnexions entre différents types de routes et la gestion des itinéraires et d'autre part la localisation exacte des véhicules dans l'environnement à partir de cette simulation logique.

L'objectif général que nous voulions atteindre avec cette simulation était d'être capables de modéliser l'ensemble du réseau à partir des mêmes composants, en favorisant la généricité des objets créés, afin d'être capables de pouvoir représenter avec la même facilité des petites ruelles à sens unique aux autoroutes en passant par les carrefours complexes où il faut gérer les feux et les interconnexions avec divers types de sous-réseaux empruntés par les bus, les piétons et divers autres véhicules.

2.2. Modélisation Logique

Comme brièvement expliqué dans le paragraphe précédent, modélisation logique correspond à la modélisation du réseau urbain d'un point de vue logique, c'est-à-dire qu'on va créer des notions de routes, de carrefours et autres types d'interconnexions afin de modéliser l'ensemble du réseau à simuler.

Etant donné que dans ce projet une grande importance a été donnée à la définition et à la modélisation du réseau, nous détaillons ici les objets manipulés par le simulateur afin de permettre une meilleure compréhension des structures manipulées.

Enfin, étant donné qu'il nous a été demandé de réaliser les sources de ce projet en anglais nous indiquons entre parenthèses les traductions dans la langue de Shakespeare.

Objets Mobiles (Mobile Item)

Les « Objets Mobiles » sont les objets qui vont être manipulés par la simulation, cette catégorie regroupe aussi bien les piétons que les différents véhicules. Cependant,

bien que l'implémentation permette théoriquement d'inclure les piétons dans cette simulation, nous nous sommes concentrés uniquement sur le cas des véhicules, ce qui est l'objectif principal du projet, ainsi nous parlerons généralement de véhicules pour désigner les divers éléments pouvant évoluer sur le réseau.

Afin de simplifier au maximum la simulation d'un grand nombre de véhicules, nous avons choisi de faire évoluer nos mobiles sur un réseau dont la structure pourrait s'apparenter à un réseau ferroviaire, ainsi, pour chaque objet, son comportement est uniquement déterminé par l'objet le précédant.

Cette solution apporte deux principaux avantages : d'une part on évite au maximum les tests de collisions qui s'avèrent relativement coûteux, et d'autre part on peut simuler les véhicules en suivant les règles qui s'appliquent au suivi des files de véhicules.

Cette solution implique d'avoir une double représentation du réseau, d'une part le réseau logique, sur lequel les véhicules vont avoir des positions relatives par rapport à un point donné, et la représentation physique qui elle calculera la position réelle dans l'environnement à partir de cette position relative. Le traitement des données est effectué sur le schéma suivant :

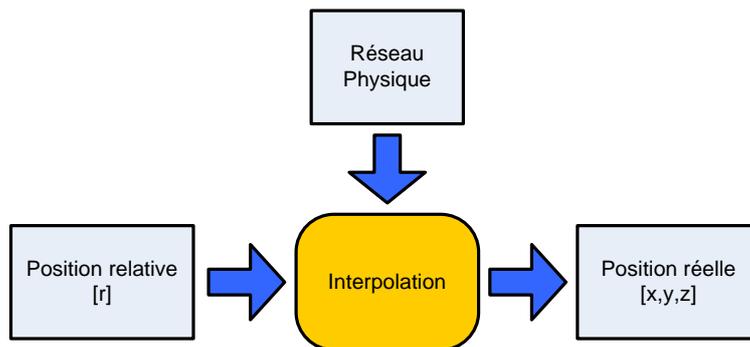


Illustration 1 - Interpolation entre modèle relatif et réel

Routes (Roads)

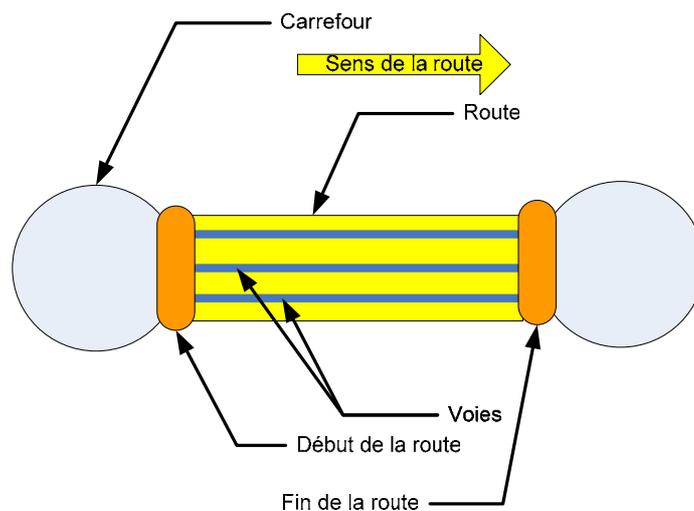


Illustration 2 - Modélisation des routes

Une route est constituée d'une ou plusieurs voies et relie un carrefour à un autre. Une route relie obligatoirement deux carrefours, éventuellement le même si on désire réaliser une boucle.

Une route n'est pas orientée d'un point de vue de la simulation, c'est-à-dire que des véhicules peuvent emprunter une route dans les deux sens, sous réserve qu'il y ait des voies adéquates.

Cependant, étant donné qu'il était nécessaire de pouvoir distinguer les deux extrémités d'une route, il était nécessaire de les orienter, mais il ne faut pas confondre l'orientation d'une route et celle de ses voies.

Voies (Lanes)

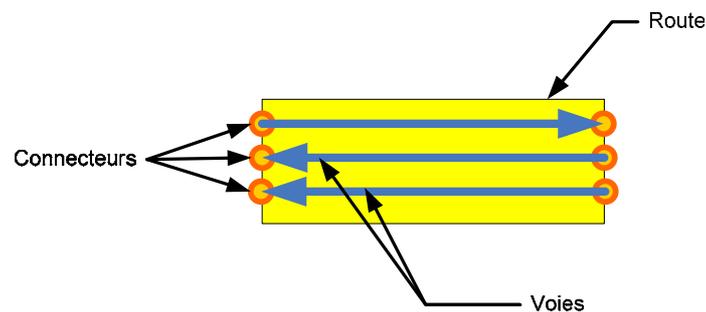


Illustration 3 - Modélisation des voies

Les voies sont les éléments constituant les routes, typiquement une voie est un axe de communication qui est emprunté par une file de véhicules.

Les voies sont orientées relativement au sens de la route, on aura donc des voies dans le même sens que la route, et des voies dans le sens contraire.

Une voie relie deux connecteurs, qui se trouvent à ses extrémités. Ces connecteurs permettent de regrouper plusieurs voies vers une seule, et aussi de marquer la séparation d'une voie en plusieurs autres voies.

Une voie peut être propre à un type d'objet mobile, tels que les bus, ou encore les piétons ou bien être « générique » et accepter par exemple les bus, les voitures et les poids lourds simultanément.

Connecteurs (Connectors)

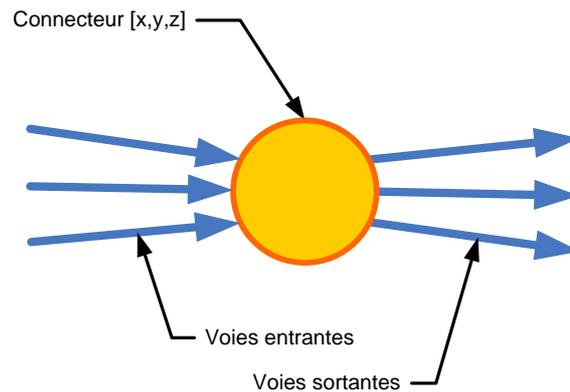


Illustration 4 - modélisation des connecteurs

Les connecteurs, partiellement décrits précédemment, permettent de connecter plusieurs voies par leurs extrémités.

Dans un connecteur, on distinguera les voies entrantes, c'est-à-dire dont l'extrémité terminale est branchée au connecteur, des voies sortantes où c'est leur extrémité initiale.

Cette distinction permettra aux objets simulés de naviguer facilement à travers le réseau, en utilisant les voies sortantes des connecteurs qu'ils rencontrent pour choisir les prochaines voies qu'ils peuvent emprunter, et exploiteront éventuellement les voies entrantes d'un connecteur pour prévenir les collisions dans certains cas.

Un connecteur comporte une position dans l'espace, laquelle est dépendante du tracé des voies comme nous l'expliquerons dans la partie modélisation physique.

Carrefours (Crossroads)

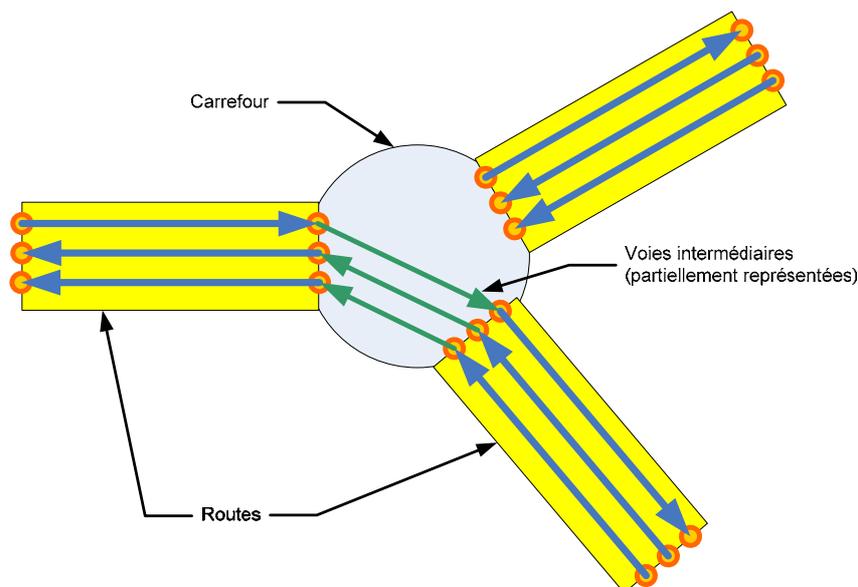


Illustration 5 - Modélisation des carrefours

Un carrefour constitue l'intersection de deux routes, ou plutôt la réunion d'extrémités d'une ou plusieurs routes.

Etant donné qu'il réalise l'interconnexion des routes, il gère en interne un ensemble de voies intermédiaires pour relier les diverses « entrées et sorties » que constituent les voies des routes connectées.

Enfin, en interconnectant les diverses voies, on doit bien veiller à respecter les sens des voies connectées, afin d'éviter toute incohérence.

Le comportement par défaut des carrefours est de réaliser toutes les connexions possibles entre les différentes voies des routes, il est cependant possible de supprimer certaines de ces voies intermédiaires afin de représenter certaines configurations particulières.

2.3.Modélisation physique

Simulation de files et lois de poursuite

La simulation telle qu'elle se présente, permet de simuler les véhicules sous forme de files de véhicules, car ceux-ci constituent des files lorsqu'ils sont sur les mêmes voies. Ainsi, on peut utiliser les lois de poursuite de véhicules qui nous ont été présentées lors d'une réunion.

Soit une file de véhicules numérotés de $n, n-1, n-2, n$ étant le premier de la file, X_n étant la position relative x du véhicule n , et $V_n(t)$ sa vitesse.

Vitesse absolue : $V_n(t) = X'_n(t)$

Distance inter-véhiculaire : $DIV = | X_{n-1}(t) - X_n(t) |$

Vitesse relative : $V_r(t) = V_{n-1}(t) - V_n(t)$

Temps de réaction T

Accélération : $ACC_n(t+T) = X''_n(t+T)$

$ACC_n(t+T) = C1 \times V_r(t) + C2 \times [DIV - D_n(t)]$

Avec $D_n(t) = L + h * X'_n(t)$

Avec $C1 = 0,4m/s, C2 = 0.14m/s^2, T = 1s, L = 5m$ et $h = 1.8s$

Modèle d'interpolation

Comme expliqué précédemment, afin de simplifier la simulation, nous avons décidé d'abstraire la représentation physique du réseau. Cette représentation est matérialisée par la notion de « tracé » qui stocke les coordonnées spatiales des voies. Elle permet de faire la jonction entre coordonnées relatives utilisées pour la simulation et coordonnées tridimensionnelles réelles qui seront nécessaires à la visualisation des objets mobiles.

Un tracé est une suite de points formés d'une position (X, Y, Z) et d'une orientation (vecteur d'orientation) ainsi qu'une méthode d'interpolation permettant de rendre un

tracé continu. L'interpolation est réalisée entre les coordonnées d'un tracé deux à deux directement entre deux positions.

Un changement de méthode d'interpolation pourrait permettre de gérer des courbes, et ainsi de mieux s'adapter à l'environnement à représenter.

3. Réalisation

3.1. *Modélisation initiale*

Lors du démarrage de ce projet, nous avons longuement analysé et étudié la structure de représentation du réseau ainsi que celle de la simulation elle-même.

Le schéma UML initial à l'issue de cette analyse est présenté en annexe, bien qu'il y ait quelques différences entre la structure présentée et celle réellement implémentée, l'essentiel de l'architecture est restées identique.

3.2. *Implémentation effective*

Nous avons généré une documentation au format Doxygen, ce qui permet de mieux se représenter la structure du programme, d'autant plus que celle-ci contient des diagrammes de collaboration entre les objets.

De plus, le code source est commenté et a été réalisé en anglais, ce qui permet une collaboration éventuelle d'étudiants étrangers au projet.

3.3. *Outil de débogage*

Etant donné qu'un simulateur n'a pas à proprement parler besoin d'une interface utilisateur, il n'était pas initialement prévu d'interface pour ce composant de l'application.

Cependant, étant donné qu'il était particulièrement complexe de vérifier sans interface de visualisation le bon fonctionnement du projet, nous avons décidé d'interfacer simplement notre simulation avec une représentation réalisée à l'aide de l'API OpenGL.

Bien que sommaire, celle-ci permet de visualiser en 3D les routes et les voies du réseau modélisé qui sont représentés par des ensembles de lignes, ainsi que les divers connecteurs représentés eux par des sphères de couleur, les véhicules sont représentés par des « théières » afin de facilement les distinguer.

Cet outil nous a permis de visualiser plus efficacement l'avancement de notre travail, tout en nous permettant de corriger bon nombre d'erreurs qui auraient été bien difficiles à identifier sans.

3.4. *Construction d'un réseau*

Dans notre implémentation, la construction du réseau se fait de manière statique en éditant un nombre de directives actuellement situées dans la classe Network.

Pour créer un réseau, il faut d'abord créer et placer les carrefours, puis ensuite créer les routes, et enfin connecter les routes aux carrefours, ce qui se fait via la méthode connectEndCrossroads et connectStartCrossroads de la class Road.

Ensuite, il faut spécifier pour chaque route les voies qui la composent par des appels successifs à la méthode Road::addLane, les paramètres de cette méthode

permettent de spécifier un type de voie ainsi que la direction relative par rapport à la route.

L'ajout de voies se fait en partant de la gauche de la route lorsqu'on se trouve dans le même sens que celle-ci.

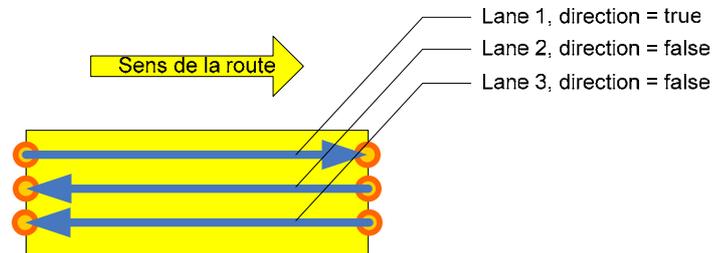


Illustration 6 - Construction d'une route

La dernière étape est la génération des voies intermédiaires dans les carrefours, qui se fait de manière automatique en appelant la méthode `Crossroads::generateAllConnections`. Il faut noter que cette génération doit se faire uniquement après l'allocation des voies aux différentes routes, dans le cas contraire les voies créées à posteriori ne seraient pas connectées au sein du carrefour.

4. Améliorations et extensions possibles

Etant donné que l'un des objectifs de ce projet est de servir de base à la création de simulations complexes, il existe un grand nombre d'extensions possibles à ce projet.

Interfaçage avec interface graphique via le réseau

Etant donné que nous nous sommes concentrés sur l'aspect simulation et modélisation du projet, nous n'avons pas pu réaliser l'interfaçage avec l'interface graphique via le réseau par manque de temps.

Cependant, étant donné que cela avait été prévu dès le départ, nous avons réalisé la structure MobileData qui contient les informations qui seront envoyés à l'affichage pour la visualisation.

Interpolation réaliste

Pour le calcul de l'interpolation nous avons fait une interpolation linéaire directe entre les points deux à deux.

Le résultat est que le tracé est constitué d'un ensemble de segments, et qui apparaissent donc pas très réalistes, une meilleure interpolation devrait pouvoir générer des courbes afin de restituer plus fidèlement l'environnement.

Stockage des données du réseau

Pour l'instant, la construction du réseau se fait de manière statique directement dans la classe Network. La problématique du stockage de ces données devient donc importante lorsqu'on désirera modéliser des réseaux de grande taille.

Il serait donc concevable d'interfacer le réseau avec une source de données externe, qui pourrait être soit un fichier, binaire ou XML, ou encore une base de données.

Outil d'édition du réseau

En parallèle du stockage externe des données du réseau, il serait particulièrement utile de développer un outil permettant d'éditer le réseau, afin de faciliter l'étape de modélisation.

5. Annexe : diagramme de classe initial

