



Cahier des charges du TER FMIN200  
du Master IFPRU Informatique  
du 19 Janvier au 18 Mai 2009

GROUPE A

# Navigation interactive dans un paysage urbain

**Encadrant :** Nancy RODRIGUEZ

**Etudiants :** BENABADJI Mehdi  
DELMAS Gabriel  
LOMBARD Matthieu  
OCHMAN Cécile

# I. Présentation du sujet :

## 1) Introduction

Les modèles 3D sont actuellement de plus en plus détaillés pour répondre aux besoins des simulations. Mais, pour pouvoir visualiser ces modèles de manière interactive, nous devons réduire leur complexité. La stratégie pour atteindre ce but est plutôt simple : seuls seront considérés pour le rendu du modèle, les éléments qui seront visibles dans l'image final.

Dans ce sens, les recherches se focalisent sur trois aspects :

- La subdivision spatiale, permettant de décider si un objet est potentiellement visible étant donné sa position dans l'espace.
- La visibilité, permettant de décider si un objet est visible ou non étant donné sa position relative à l'observateur (dans le volume de vue, faces cachées ou occultées)
- Les niveaux de détails ou LOD (Level of Detail), permettant de créer une version simplifiée d'un objet lorsque l'affichage détaillé n'est pas nécessaire (e.g. objet lointain, en mouvement rapide, en périphérie du volume de vue)

La recherche en LOD a débuté en 1976, date de l'introduction par James Clark [Clark 76] des principes et des avantages de la représentation d'un objet à plusieurs résolutions. Des dizaines de techniques ont été proposées pour la création et la gestion des LOD.

Pour construire un modèle plus simple, il est nécessaire de simplifier la géométrie du modèle original de manière à réduire le coût de rendu de l'objet tout en essayant de préserver au maximum son apparence visuelle. Deux éléments sont à considérer lors de la simplification : la géométrie (l'ensemble des sommets) et la topologie (la connectivité du maillage). La simplification de la géométrie n'amène pas forcément à la simplification de la topologie.

Suivant la distance à laquelle est vue un objet en 3D, l'œil humain ne perçoit pas les détails de la même manière. Afin d'offrir une meilleure fluidité sans grande perte de détail, il y a possibilité pour le développeur de réduire le nombre de polygones en fonction de la distance.

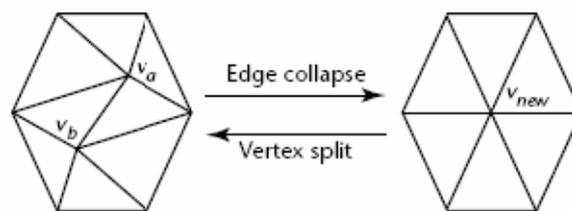
- \* Ainsi, pour un objet éloigné, on créera un modèle peu précis.
- \* Pour un modèle plus proche, l'objet 3D sera un peu plus complexe et travaillé.
- \* Pour une vue de très près, l'ensemble des détails sera visible.

Les algorithmes de simplification géométrique permettent de créer une version simplifiée (niveaux de détails ou LOD) d'un modèle, qui sera utilisée lorsque l'affichage détaillé n'est pas nécessaire (exemples : objet lointain, en mouvement rapide, en périphérie du volume de vue)

### a) Les opérateurs locaux

Il existe deux grandes familles d'opérateurs utilisés pour la simplification géométrique : les opérateurs locaux qui simplifient le maillage de manière locale (e.g. les méthodes de décimation qui suppriment un sommet, une arête ou une facette) et les opérateurs globaux qui opèrent sur un très grand nombre de sommets au même temps (parfois tout le maillage). Un exemple d'opérateur local est "l'edge collapse" proposé par Hoppe [Hoppe 96].

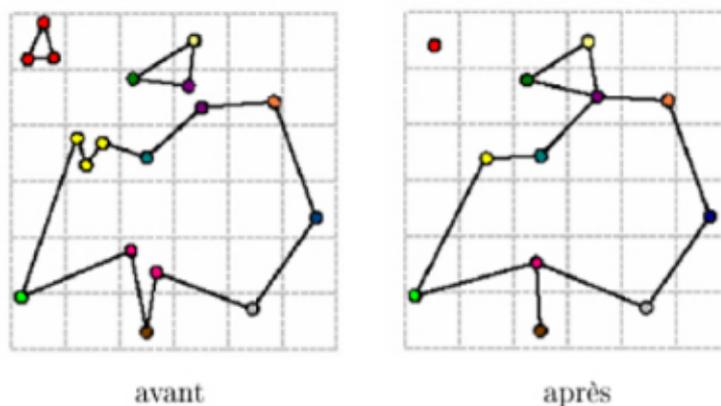
Cet opérateur transforme une arête ( $v_a, v_b$ ) en un sommet ( $v_{new}$ ), il supprime l'arête ainsi que les triangles auxquels elle appartient. L'opération inverse est le "vertex split", qui additionne au maillage une arête et ses triangles adjacents (voir figure suivante). L'opérateur "edge collapse" permet donc de simplifier la géométrie tandis que l'opérateur "vertex split" ajoute des détails au modèle.



Simplification et ajout de détails

### b) Les opérateurs globaux

Le cell collapse est, quant à lui, un opérateur global. Cet opérateur est un opérateur de clusterisation, c'est-à-dire qu'il regroupe les sommets du maillage en un nombre fini d'ensembles. L'opération permet de transformer un ensemble de sommets, situés à l'intérieur d'un certain volume ou cellule, en un seul sommet. La cellule contenant les sommets peut appartenir à une grille ou à une subdivision spatiale telle qu'un octree [Rossignac & Borrel 93].



Cell collapse avec une grille régulière



## II. Déroulement :

### 1) Analyse des différents algorithmes de LOD

Pour mener à bien ce projet, nous nous appuyons sur plusieurs algorithmes de construction des LOD :

- Un algorithme global par échantillonnage pour la création des LOD pour un objet [Garland 97] et pour toute la scène [Luebke 01]
- Un algorithme hiérarchique utilisant des LOD par objet [Strauss]

### 2) Création de LOD discret pour des modèles tests

Plus un objet est petit, moins ses détails sont perceptibles (comme nous pouvons le voir d'après la figure suivante)

Lorsqu'on se trouve près de l'objet, il est important de le représenter avec le niveau de détails de plus élevé possible.

La création de LOD discrets se fait en utilisant la bibliothèque QSLIM

### 3) Visualisation avec les bibliothèques Ogre3D et GLOD de modèles de tests en utilisant les LOD créés

Pour implémenter/visualiser les algorithmes cités précédemment, nous allons utiliser deux méthodes différentes. Ogre3D et GLOD.

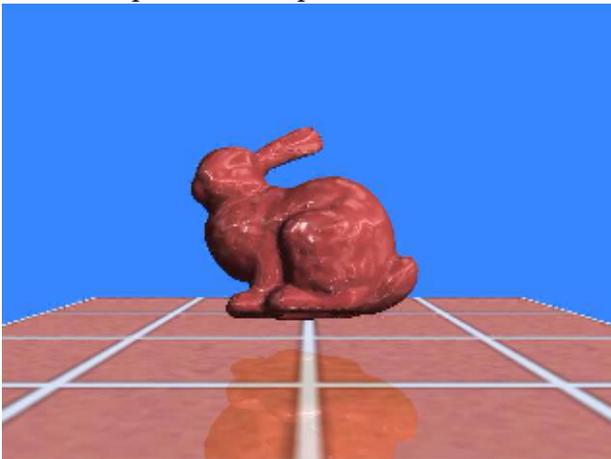


Ogre 3D est un moteur 3D libre qui va nous permettre de créer facilement une scène contenant nos modèles 3D. Nous pourrons donc faire des tests et des comparatifs (de performance et de rendus) entre les différents algorithmes de LOD discrets que nous allons étudiés et celui disponible dans Ogre3D lui même.

#### GLOD :

La bibliothèque GLOD permet de créer et de visualiser des modèles LOD discrets et continus, à la différence d'OGRE 3D qui ne permet l'utilisation que de LOD discret. Nous pourrons ainsi faire des analyses et comparatifs des deux sortes de LOD.

Par exemple ici nous pouvons voir un modèle rendu avec GLOD.



L'utilisation des mêmes modèles et du même algorithme de création (QSLIM) dans les deux bibliothèques nous permettra d'avoir un cadre précis de comparaison.

#### Objectif Final :

L'objectif final est de pouvoir créer un modèle urbain qui offrira à la fois performance et qualité de rendu. Grâce à nos précédentes implémentations, nous serons à même de choisir la meilleure solution (LOD par objet/ par scène). Nous pensons actuellement que les LOD par scène seront plus adaptés à cet usage mais ceci n'est qu'une hypothèse qu'il nous faudra vérifier.

L'utilisation de LOD est ici indispensable car un modèle urbain implique l'affichage simultané de beaucoup d'objets 3D et sans LOD cela demanderait beaucoup trop de puissance pour avoir un affichage fluide.

### III. Méthode de travail :

#### 1) Organisation et répartition des tâches

Pour réaliser ce projet, nous avons prévu plusieurs réunions de groupe afin de coordonner nos actions. Ainsi, un rendez-vous est prévu avec Mme Rodriguez chaque jeudi après midi au lirmm pour les différentes questions que nous aurions à lui poser.

De même, un rendez-vous hebdomadaire est prévu pour que nous puissions discuter de l'avancement du projet. Ayant plusieurs cours en commun ce semestre, nous aurons également l'occasion de nous voir pendant ces derniers. En ce qui concerne la communication en dehors de l'université, nous avons bien sûr les adresses mails des différents membres du groupe à disposition et aussi celle de Mme Rodriguez.

Pour ce qui est de l'échange de documents et d'informations, nous avons prévu un FTP avec Subversion(SVN) pour que l'on puisse visualiser l'avancement du projet dans sa globalité.

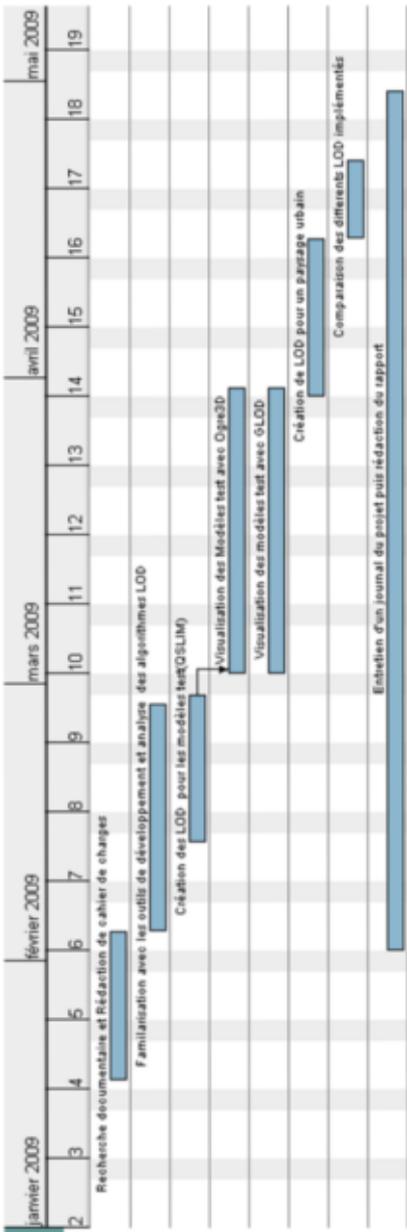
Ce serveur, nous permettra donc de disposer en temps réel des différentes versions du code, informations importantes, et des avancements sur les différentes parties du projet. Nous disposons également des outils de messagerie instantanée et de nos numéros de téléphone pour le cas où l'ensemble des autres moyens de communication ne conviendraient pas ou seraient indisponibles.

En ce qui concerne la répartition du travail entre les différents membres du groupe, nous travaillons le plus souvent par binômes, nous avons prévu quatre grandes phases dans ce projet : Création de LOD discret pour des modèles test et Visualisation avec la bibliothèque Ogre3D de modèles de tests en utilisant les LOD créés qui seront traités par Gabriel et Mehdi. Quant à la visualisation des modèles test à l'aide de la bibliothèque GLOD et la création des LOD pour un paysage urbain seront traités par Cécile et Matthieu en utilisant les LOD scène fournis par GLOD.

Nous travaillerons aussi individuellement pour les petites parties de codage, pour les recherches de documentation et pour la configuration des outils.

#### 2) Le planning prévisionnel

Voici le diagramme de Gantt qui est la synthèse représentant la répartition temporelle de chacune des tâches à accomplir tout au long de ce TER :



### 3) Outils de travail

Pour la partie programmation, nous avons choisi le C++. Nous voulions un langage familier mais surtout qu'il nous permette d'utiliser les bibliothèques graphiques ainsi que les API nécessaires à notre projet tel que GLOD, QSLim, Ogre3D.

Nous avons choisi Visual C++ 2008 express comme environnement de développement car il intègre un assistant de développement pour l'API Ogre3D

# Bibliographie

[Clark 76] James Clark. Hierarchical Geometric Models for Visible Algorithms. *Communications of the ACM*, 19(10) :547–554, 1976

[Hoppe 96] Hugues Hoppe. Progressive meshes. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99-108. ACM Press, 1996.

[Rossignac & Borrel 93] J. Rossignac and P. Borrel. Multi-resolution 3D approximations for rendering. B. Falcidieno and T. L. Kunii, editors, *Modeling in Computer Graphics*, pages 455-465. Springer-Verlag, June-July 1993.

[Garland 97] Michael Garland and Paul Heckbert. Surface Simplification Using Quadric Error Metrics. *Proceedings of SIGGRAPH 97*.

[Luebke 01] David Luebke. A Developer's Survey of Polygonal Simplification Algorithms. *IEEE Computer Graphics & Applications* (May 2001)