

# Rapport de projet géomatique

OpenGL/LibCityGML

Maryam MARIH - Hamza BIADI

# Contents

<b>1</b>	<b>Description du projet</b>	<b>2</b>
1.1	Reformulation du sujet : . . . . .	2
1.2	Définition des bibliothèques utilisées : . . . . .	2
1.3	Description de la démarche suivie : . . . . .	3
1.4	Le code : . . . . .	3

# Chapter 1

## Description du projet

### 1.1 Reformulation du sujet :

Le programme crée est écrit en langage `c++`. Il permet de charger, via une interface QT, un fichier de type `cityGML`, de l'analyser, à l'aide des fonctions de la bibliothèque `LibCityGML`, et ensuite de dessiner son contenu en 3D tout en utilisant les fonctions de la librairie `OpenGL`.

L'application propose en plus de l'affichage 3D une autre fenêtre réservée à l'affichage réduit de la totalité de la scène en 2D, elle se focalise sur la zone visible dans la fenêtre principale de navigation. Le programme permet également une navigation en immersion avec une caméra quaternion. Le code développé est constitué de plusieurs classes. Chacune gère une partie du programme à savoir : les interfaces, la caméra, le traitement des fichiers `CityGML` et les tests automatiques.

### 1.2 Définition des bibliothèques utilisées :

Durant notre projet, nous avons manipulé plusieurs librairies `C++` à savoir :

1- `LibCityGML` qui est une petite bibliothèque `C++` open source. Elle permet l'analyse de fichiers `CityGML` de telle sorte que les données peuvent être facilement exploitées par des applications de rendu 3D (données géométriques sont optimisés pendant l'analyse pour un bon rendu) Par exemple, elle peut être utilisée pour développer des programmes de lecture de fichiers `CityGML` dans de nombreuses applications basées sur la modélisation 3D (`OpenGL`, `OpenSceneGraph`, ...) `libcitygml` est développé par l'équipe 3D du BRGM (le premier établissement public français impliqués dans le domaine des sciences de la Terre pour la gestion durable des ressources et des risques naturelles) pour le projet de recherche `DeepCity3D`

2- `OpenGL` : Initialement développé par Silicon Graphics en début des années 90, `OpenGL` est devenu un standard de graphiques ouvertes le plus largement utilisé dans le monde. C'est est une librairie exploitant la carte graphique d'un

ordinateur permettant ainsi aux développeurs de programmer des applications 2D et 3D. Cela permet en autres de développer des jeux-vidéo. L'avantage principal d'OpenGL est qu'il est multi-plateforme, c'est-à-dire qu'un programme codé avec OpenGL sera compatible avec Windows, Linux et Mac, sous réserve que la gestion de la fenêtre et des inputs soient également multi-plateforme (comme la SDL)

Le programme développé permet l'affichage des objets contenu dans des fichiers de type cityGML. En effet CityGML signifie : City Geography Markup Language . C'est un schéma basé sur XML pour la modélisation et l'échange de modèles 3D géoréférencées de villes et du paysage.

### 1.3 Description de la démarche suivie :

Nous avons commencé notre travail par la compréhension et le développement de la demande. La délimitation du périmètre de sujet est une étape très importante pour bien déterminer les grandes lignes et les fonctionnalités principales qu'il faut développer tout au long du projet.

Vu que nous n'avons jamais manipulé la bibliothèque LibCityGML, une recherche bibliographique profonde s'est avérée indispensable. Nous nous sommes alors penché sur la découverte de cette librairie et nous avons détecté les fonctions qui nous paraissaient importantes et qui peuvent nous être utiles dans le projet.

Ensuite, nous avons modélisé les classes que nous allons développer. Dans un premier temps, nous avons commencé par la création des interfaces avec les différents boutons et fenêtres : bouton de chargement du fichier, fenêtre d'affichage des objets... Ensuite nous avons développé la classe de la caméra qui marchait très bien et donnait de bons résultats, mais après réflexion et pour améliorer notre code et utiliser de nouvelles fonctions, nous avons conçu une autre caméra en utilisant les quaternions.

D'autre part, la gestion du fichier cityGML se fait principalement dans une classe dédiée qui s'appelle : citygmlhandler. Cette classe permet l'analyse du fichier cityGML chargé via l'interface à savoir : lire son contenu, déterminer l'enveloppe des objets, récupérer les objets, appliquer la texture, déduire les coordonnées de la position de la caméra... et enfin pour dessiner les objets, nous nous sommes servis des fonctions d'OpenGL.

### 1.4 Le code :

Le programme développé est constitué de cinq classes principales et d'autres classes réservées aux tests unitaires.

#### **La classe mqcam :**

Cette classe implémente toutes les méthodes qui contrôlent les mouvements de la caméra quaternion. Elle permet de définir sa vitesse de translation, sa vitesse de rotation, ses coordonnées dans la fenêtre d'affichage ainsi que les

coordonnées du point cible, c'est-à-dire le point de la scène qui se trouve au centre de la fenêtre.

Dans cette classe les fonctionnalités des touches clavier et des clics souris sont précisées pour la bonne gestion des mouvements de la caméra.

**La classe citygmlhandler :**

Les fonctions de cette classe permettent de :

- charger un fichier cityGML
- parser le fichier chargé
- Traiter le fichier (récupérer les objets...)
- Dessiner les objets
- Choisir le niveau de détail

**La classe glwidget :**

C'est la classe qui dessine, avec les fonctions d'OpenGL, les objets cityGML, elle les redimensionne, elle leur applique la texture et les affiche dans l'interface QT.

**La classe mainwindow :**

Elle s'occupe de la gestion du contenu des interfaces et définit le comportement des différents boutons. La classe permet aussi de récupérer les coordonnées de l'enveloppe du fichier cityGML chargé et puis elle les utilise pour donner aux fonctions les bonnes paramètres en entrée. De plus, la classe personnalise l'affichage des fenêtres de telle sorte que les objets dessinés apparaissent tous dans la bonne partie de l'interface.

**La classe main :**

La classe principale qui inclut toutes les autres classes et nous permet d'exécuter le programme complet.

**Les tests :**

Pour vérifier le bon fonctionnement de quelques parties du programme, nous avons mis en place des classes réservés aux tests unitaire. Par exemple la classe camtest qui s'assure que la vitesse de la caméra n'a pas dépassé la vitesse maximale.