Cahier des charges du projet Open Scene Graph

Le projet à la base est de montrer la faisabilité d’affichage de toutes sortes d’informations (topographiques, géographiques, environnementales) sur un modèle 3D représentant des environnements urbains, ces informations étant regroupées dans un fichier cityGML. Le but est d’implémenter des fonctions en C++ incluant l’API Openscenegraph qui affichent ces informations.

Nous avons fait le choix d’étudier également Neoaxis, et d’en faire une comparaison avec Openscenegraph, en raison de sa communauté plus dynamique et de sa facilité à mettre en place les méthodes d’affichage voulues du fait que Neoaxis est un SDK.

**Objectifs :**  
Dans le cadre de ce projet, nous allons étudier et implémenter pour chacune des 2 librairies :

* Capacité de créer un HUD (head user display) relatif aux éléments des scènes : dans ce cas il s’agit d’un texte 2D sur l’écran

-          Ajout de point particulier dans un repère local, tel que par exemple l’origine d’un repère

-          Affichage de label permettant de donner une information textuelle sur un élément de la scène (point, surface, volume).

-          Affichage de données numériques relatives aux surfaces, et transformation en texture. Projection sur une surface. Les valeurs du champ scalaire seraient représentées par un code couleur générique, et modifiable selon le type de données.

-          Affichage de données 3D dans l’espace sous la forme d’un champ de vecteurs 3D : affichage d’une grille 3D de flèches dans un espace local à définir. Affichage de différents vecteurs relatifs à leur espace.

-       Etude faisabilité d’une utilisation dans un environnement de réalité virtuelle (affichage en stéréovision, gestion des périphériques de Réalité augmentée)

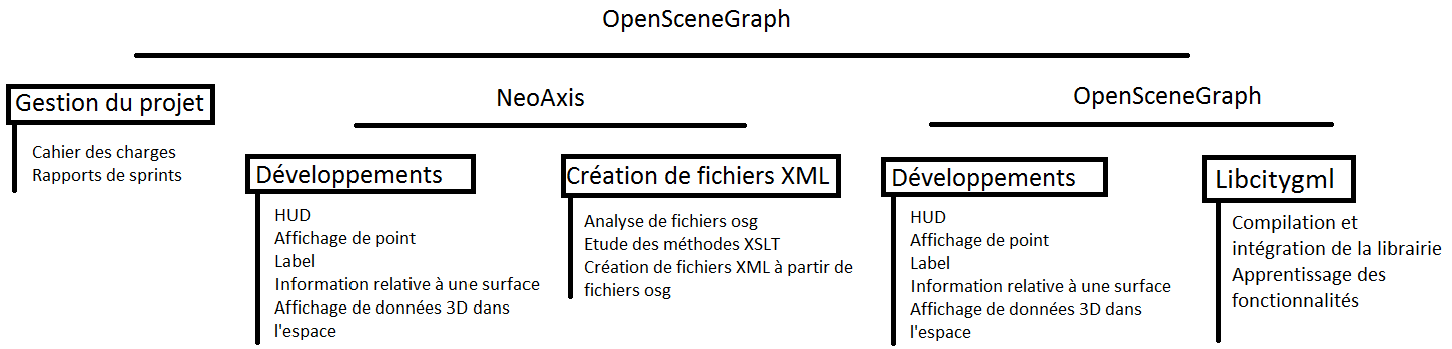
- Lecture et affichage de données issues d’un fichier CityGML, à savoir les données géométriques, avec gestion du niveau de détail, et les métadonnées, avec un choix explicite de la couche à afficher.

- NeoAxis : création d’un module permettant de créer un fichier xml, utilisable par neoaxis, depuis un fichier citygml.

- OSG : Vérification du bon fonctionnement de libcitygml : il doit convertir suffisamment de données pour pouvoir être pertinent au sein du projet c'est-à-dire que l'on devra ajouter des fonctionnalités à libcityGml afin que la librairie supporte une plus large gamme de caractéristiques.

- Étudier la compilation sur Android version 3.x.x (Honeycomb), configuration matérielle : minimum dual core 1Ghz, 1 Go de ram, résolution écran 800x400

Ces différents objectifs nous donnent le WBS suivant :



Planning  
Le travail se décompose en 2 branches parallèles :

-          La partie neoaxis

-          La partie open scene graph

Le développement utilisera les méthodes agiles.  
  
Ce projet est constitué de 4 sprints de 2 semaines :  
- 15-26 octobre : création du HUD + vérification du bon fonctionnement de libcitygml + étude de la structure de fichiers citygml  
-  5 - 16 novembre : label + affichage d’un point particulier + vérification de la stéréovision  
- 19 - 30 novembre : affichage 2D d’un champ scalaire + conversion fichier citygml en xml  
- 3 - 14 décembre : Affichage d’un champ de vecteurs 3D  
  
  
  
Il nous restera 1 semaine pour faire le rapport et la démonstration

Livrables :

A la fin de chaque sprint, nous devrons rendre :

* Le code source montrant l’implémentation d’une démonstration
* Une illustration de la démonstration, telle qu’une capture d’écran par exemple
* La documentation liée à cette implémentation
* Le rapport lié à chaque sprint

Répartition du travail :

Globalement, on aura 3 personnes sur la branche OpenSceneGraph et 2 personnes sur Neo Axis. Néanmoins, nous comptons mener ce projet à l’aide de pratiques agiles, ceci est donc susceptible de changer selon les difficultés rencontrés au cours des différents sprints.

1er Sprint :

* Création HUD sous OSG : Yohann M et Yohann F
* Création HUD sous NeoAxis : Vincent
* Vérification du bon fonctionnement de libcitygml : Rémi et Brice

2ème Sprint :

* Affichage d’un label et d’un point sous Neoaxis : Vincent
* Affichage d’un label sous OSG : Yohann M et F
* Affichage d’un point sous OSG : Rémi et Brice
* Stéréovision : Vincent pour Neoaxis et les autres pour OSG

3ème Sprint :

* Affichage champ scalaire sous Neoaxis : Vincent
* Affichage champ scalaire sous OSG : Yohann M et F
* Conversion citygml en ogrexml : Rémi et Brice

4ème Sprint :

* Affichage champ vecteur 3D sous Neoaxis : Vincent
* Affichage champ vecteur 3D sous OSG : les autres