**Cahier des charges Math imagerie 3D**

**Java 3D**

Bref historique et descriptif

1. Contraintes matérielles. Et mise en place du système.
2. Etude des paradigmes classiques de programmation de base. Premier programme, exemples simples, primitives de bases, … En gros, un « Hello World ! » de la technologie.
3. Que permet de faire chaque technologie ?

Faire le lien entre la théorie et ce que la technologie est capable de faire. Peut-on gérer un modèle d’illumination qui tient compte des lumières de la scène, etc… Si oui, comment ? Le montrer avec des exemples de codes.

Constitue-t-elle un moteur 3D complet ou faut-il programmer les projections, la facettisation des objets, le rendu, etc…

1. Plaquage de textures sur les objets

Techniques propres à la technologie.

**Ressources labo**

**Liste vue au cours :**

1. Général :
   1. Type d’observateurs : projection parallèle ou en caméra de pino
   2. Le dessin se fait en reliant les points à partir de l’équation: exemple de facettisation d’un cylindre, d’un polyèdre et d’unesphère et d’une surface paramétrée.
   3. Le dessin se fait avec les surface paramétrée : exemple de facettisation d’un cylindre, d’un polyèdre et d’une sphère et d’une surface paramétrée.
2. Affichage
   1. Affichage en fil de fer
   2. Remplissage d’un polygone
   3. Elimination des partie cachées : quel algorithme utilisé
3. Model d’illumination
   1. Lumière ambiante
   2. Sources lumineuse ponctuelles non directionnelle
   3. Réflexion diffuse
   4. Réflexion spéculaire
4. Lissage des images : quel algorithme ?

**Qu’est-ce qu’un moteur 3D :**

Un moteur 3D est une API (i.e. il fournit un ensemble de fonctions) qui permet à son utilisateur, en l'occurrence vous, de représenter des objets dans un monde en trois dimensions. (On s'en serait douté en même temps.) Le moteur 3D est un intermédiaire entre vous et les librairies graphiques de plus bas niveau comme DirectX ou OpenGL, qui permet donc de se passer des aspects les plus complexes de la gestion d'objets 3D. Le moteur gère lui-même les objets dans la scène et vous donne accès à des fonctions pour les manipuler et les afficher de façon optimale.

Cependant, il ne faut pas confondre moteur 3D (ou moteur graphique) et moteur de jeu, voire moteur physique par exemple. En effet, un moteur 3D ne contient pas de fonctions pour gérer les collisions entre les objets ni pour les entrées utilisateurs. Ceci sera fait à l'aide de bibliothèques différentes à intégrer au programme, mais ce n'est en aucun cas Ogre qui gérera ça.

Pour résumer, voici par exemple ce qu'Ogre nous permettra de faire :

* afficher des personnages, bâtiments, paysages en 3D ;
* gérer les lumières et les ombres ;
* utiliser une ou plusieurs caméras ;
* créer des effets de brouillards, flammes, fumées...

En gros, cela concerne tout ce que l'on voit à l'écran, tout ce qui est en rapport avec le *rendu*.