

Projet de synthèse d'images

Master Informatique Science de l'Image – 2^{ème} année

Roller Coaster

L'objectif de ce projet est la création d'un « roller coaster » virtuel. Ce projet se réalise par monôme ou binôme.

Le but de ce projet est de réaliser une application 3D permettant de simuler un « roller coaster », des montagnes russes. Il vous permettra de mettre en œuvre un moteur de rendu simple.

1. Le roller coaster : la scène

Un roller coaster c'est une montagne russe. Pour notre application, ce que l'on demande surtout c'est qu'un véhicule, se déplace le long d'une trajectoire préétablie et que l'on puisse regarder la scène à partir de ce véhicule mais également que l'on puisse voir la scène de loin. La scène du roller coaster comporte donc : le véhicule, la « trajectoire », ou chemin, qui doit avoir une représentation visuelle (quelle qu'elle soit) et également un ciel et un environnement (aussi simple soient-il). Enfin, cette scène sera plongée dans 2 ambiances lumineuses : une de « jour » et une de « nuit ».



1.1. Le véhicule

La modélisation du véhicule est laissée à votre entière discrétion. Mais chaque groupe doit avoir sa propre modélisation du véhicule.

Le véhicule sera donc un objet 3D complet et devra être également texturé.

Vous trouverez dans la partie « technique » comment vous pouvez exploiter des bibliothèques de chargement de modèle.

1.2. La trajectoire

La trajectoire doit avoir, en plus d'une représentation abstraite (i.e. mathématique ou informatique) une représentation visuelle. On doit donc « voir » la trajectoire.

Concernant la forme de la trajectoire elle-même, elle devra comporter au minimum une section courbe, c'est à dire que la trajectoire ne peut pas être simplement une ligne brisée. De plus, la trajectoire exploitera l'ensemble de l'espace et sera donc « 3D ». En gros, on ne veut pas d'une trajectoire qui serait contenue dans un plan. Concernant les possibilités mathématiques de cette trajectoire, vous pouvez rechercher des informations sur les courbes de Bézières, ou les courbes splines (il en existe de multiples types).

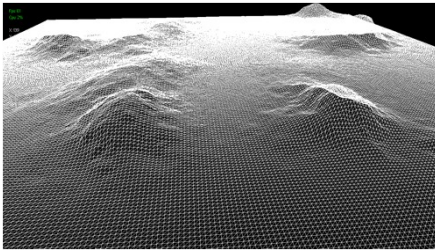
La modélisation 3D, et donc l'aspect visuel, de la trajectoire est, là encore, laissée à votre appréciation. La version la plus minimaliste serait de représenter cette trajectoire par une simple ligne. Évidemment de vrais modèles 3D, probablement répétitifs, seraient appréciés et chaque groupe devra proposer une visualisation originale de sa trajectoire.

1.3. L'environnement

L'environnement constitue le reste de l'univers. C'est la scène que l'on observe depuis le roller coaster. L'environnement contient les éléments suivants :

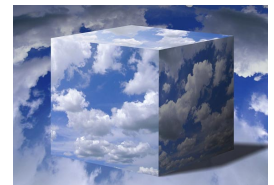
- Un sol ou un décor
- Un ciel
- Deux lumières, une directionnelle (semblable à un soleil) et une autre attachée à l'avant du véhicule et qui éclaire la trajectoire

D'autre part, le monde aura 2 configurations : un éclairage de jour et un éclairage de nuit.



Concernant le sol ou le décor, chaque groupe devra avoir une représentation originale à présenter. Le sol peut varier d'un simple plan à un environnement plus complexe. Mais quoiqu'il en soit, **ce décor devra contenir des textures**. Pour la modélisation du sol, vous pouvez penser à des sols « fractals », des sols générés par « height map » ou tout simplement via le chargement d'un modèle 3D.

Le ciel sera constituée de 2 « skybox ». Une skybox est un cube texturé qui englobe la scène. Traditionnellement, il est dessiné avant tout autre objet, autour de la caméra et en désactivant l'écriture dans le Z-buffer. Cela permet, ensuite de dessiner tout le reste par dessus. Vous implémenterez deux skybox différentes pour les deux ambiances lumineuses de jour et de nuit.



Enfin la scène comporte deux lumières différentes : une directionnelle, semblable à un soleil. Et une autre de type spot, située à l'avant du véhicule, qui éclaire la trajectoire située devant le véhicule. Dans l'ambiance « jour », seule la lumière directionnelle est active. Dans l'ambiance nuit, seule la lumière spot est active. Enfin vous avez la possibilité d'ajouter d'autres lumières, actives ou non durant les 2 ambiances lumineuses (jour/nuit).

1.4. Les ambiances lumineuses

Comme précisé ci-dessus, la scène sera plongée dans deux ambiances lumineuses distinctes, une de « jour » et une de « nuit ». Il s'agit en fait de proposer deux représentations originales pour deux ambiances, dont une plutôt lumineuse (ambiance de « jour ») et une plutôt sombre (ambiance de « nuit »). Par contre, la contrainte est toujours de proposer, pour l'ambiance de jour, au moins un éclairage directionnel et pour l'ambiance de nuit, au moins un éclairage de type spot. Sachant que vous pouvez rajouter d'autres éclairages.

2. L'application du « roller coaster »

L'application elle-même devra se représenter sous la forme d'une fenêtre OpenGL qui visualise la scène. Idéalement, cette fenêtre devrait recouvrir l'ensemble de l'écran (donc proposer une application « full screen »). Enfin, l'application permettra de visualiser la trajectoire et la scène du roller coaster suivant 2 points de vue, un attaché directement au véhicule (à la place d'un visiteur), et un autre surplombant la scène et la trajectoire afin d'offrir une visualisation globale.

Les interactions entre l'utilisateur et l'application seront simples. On veut au minimum les interactions suivantes :

- Lorsque l'on est dans le point de vue du véhicule, ou du point de vue « global », pouvoir regarder, grâce à la souris, tout autour de la scène,
- A l'appui de la touche 'c', passer de la caméra du point de vue véhicule au point de vue « global »,
- A l'appui de la touche 'Escape' sortir de l'application,
- A l'appui de la touche 'n', changer « à la volée » l'ambiance jour en nuit et vice-versa.
- A l'appui de la touche 's', arrêter ou redémarrer le véhicule dans sa progression le long de la trajectoire

D'autre part, vous pouvez imaginer d'autres contrôles comme, par exemple, la capacité lorsque l'on est dans le point de vue global, d'orbiter autour de la scène, ou de faire accélérer ou ralentir le véhicule, ou encore de faire la physique du déplacement le long de la trajectoire (prise en compte de la gravité).

3. Travail à réaliser

Vous devrez réaliser une application en C/C++ utilisant de l'OpenGL 3.3 ou plus ! Vous pouvez exploiter les bibliothèques vues en TD (SDL, GLEW, glimac...) ainsi que d'autres bibliothèques, par exemple de chargement de modèle 3D (ASSIMP). Si vous souhaitez utiliser une autre bibliothèque, merci de vérifier auprès de moi si cela est possible. **Votre application devra fonctionner sous linux**, et en particulier, tourner sur les ordinateurs de l'université (du moins ceux disposant d'une carte graphique, exploitant OpenGL 3.3 au moins). Vous pouvez utiliser cmake pour la compilation.

Vous devrez ensuite réaliser les spécifications suivantes (dans l'ordre de « difficulté ») :

1. Affichage de l'environnement, du véhicule du point de vue du véhicule
2. Affichage de la trajectoire du point de vue du véhicule
3. Affichage de l'ensemble du point de vue global
4. Rajouter les textures
5. Déplacement du véhicule le long de la trajectoire
6. Mise en place de la skybox et de l'ambiance lumineuse « de jour »
7. Mise en place de l'ambiance lumineuse « de nuit » et possibilité de switcher entre les 2 ambiances

Vous devrez envoyer à biri@u-pem.fr, dans un unique fichier, les éléments suivants :

- Le code source de votre programme
- Un très court rapport (quelques pages), contenant :
 - le nom et prénom des membres du groupe
 - comment on compile le projet
 - comment on lance puis utilise l'application
 - quels sont les éléments demandés qui fonctionnent et ceux qui ne fonctionnent pas
 - d'éventuelles extensions par rapport à ce qui a été demandé.
 - des captures d'écran