

GMINT317 - Moteurs de jeux – TP1

Rendu, gestionnaire de version et IHM

Rémi Ronfard  remi.ronfard@inria.fr  <https://team.inria.fr/imagine/team/>

Objectifs

Le but de ce TP est de réaliser votre première application de rendu 3D en utilisant OpenGL. Dans ce travail, chaque étudiant aura pour but de réaliser différentes tâches :

- Apprendre à gérer les événements (clavier, souris)
- Afficher une scène simple (une caméra et un triangle)
- Créer une surface carrée à base de triangles
- Appliquer une carte d'altitude sur cette surface
- Permettre les déplacements de caméra
- Avoir « commité » sur le serveur son projet en fin de séance

Bonus :

- Afficher un brouillard localisé
- Jouer avec la lumière
- Gérer les collisions caméra-terrain
- Texturer le terrain en utilisant des couleurs
- Créer un rendu terrain infini

Framework

Pour nos différents TP, nous allons utiliser le Framework Qt. Il s'agit d'un Framework spécialisé dans les UI. Nous nous servons principalement de ce Framework afin de gérer les IO de notre application.

Gestionnaire de version

La première étape à réaliser est de sélectionner son groupe de travail, et de changer le mot de passe administrateur. (Vous pouvez également vous créer un utilisateur par la suite). Lorsque votre espace GIT est choisi, alors merci de m'envoyer par mail votre : nom, prénom, groupe sélectionné.

Cloner le deuxième TP :

```
git clone http://lange@www.lange.xyz/GMIN317/TP/groupe1/tp2.git
```

Au plus tard la semaine prochaine, vous devrez rendre un compte rendu de ce TP sur votre espace GIT, ainsi que votre code source.

Attention : éviter les messages de commit inutiles, toujours décrire le travail réalisé.

Votre première application

Afin de vous familiariser avec Qt et de simplifier votre développement, je vous recommande d'utiliser l'IDE fourni. Pour vos premiers pas, avec OpenGL, votre travail sera d'étudier l'affichage d'un triangle en OpenGL avec Qt et l'étendre :

<http://qt-project.org/doc/qt-5/qtgui-openglwindow-example.html>

Après le rendu de votre premier triangle, vous allez devoir réaliser une surface plane (16*16 sommets) composée de triangles. Grâce aux IO, vous pourrez afficher soit des triangles pleins, soit des triangles en fil de fer.

Une fois cette surface créée, appliquez des transformations aléatoires aux sommets pour réaliser un relief.

Entrées et sorties de votre première application

Afin de vous guider, les interactions avec OpenGL seront réalisées avec les événements Qt, cette stratégie sera utilisée pour déplacer la caméra :

- QMouseEvent
- QKeyEvent

Ces IO permettront de déplacer la caméra dans l'espace.

Nous utiliserons ensuite une carte d'altitude (entrée fichier de l'application) grâce aux routines élémentaires de chargement d'une image dans Qt. Pour chaque sommet de notre maillage de terrain, nous appliquerons un déplacement au sommet en corrélation avec la valeur du pixel correspondant dans la carte d'altitude.

Compte rendu

Présenter vos fonctionnalités (ex : fonction RGB vers height map, ...)

Expliquer votre démarche de développement.

Présenter votre structure de données.

Expliquer comment vous vous y prendriez pour les parties bonus.

Bonus

- Afficher un brouillard localisé

En deux étapes, réaliser un brouillard global puis un brouillard localisé.

Utiliser la fonction **glFog()** pour afficher un brouillard.

Afficher un brouillard localisé dans les vallées.

- Jouer avec la lumière

Créer une lumière de type soleil.

Créer des lumières localisées.

Jouer avec les matériaux, les modèles de lumières.

- Gérer les collisions caméra-terrain

Empêcher la caméra de traverser la surface du sol.

- Texturer le terrain en utilisant des couleurs

Utiliser la coloration de sommets pour texturer votre terrain.

Appliquer des textures en fonction de l'altitude des sommets (blanc pour les sommets très élevés par exemple).

- Créer un rendu terrain infini

Réfléchir à une stratégie multi-résolution du terrain pour permettre un rendu infini.
Générer aléatoirement un rendu cohérent.