

GMINT317 - Moteurs de jeux – TP1

Prise en main de Qt Creator, Git et OpenGL ES 2.0

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

<https://github.com/master-imagina/hmin317-tp1>

Objectif

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 :

- Afficher une scène simple (une caméra, un cube)
- Apprendre à gérer les évènements (clavier, souris)
- Créer une surface et afficher ses triangles
- Contrôler les déplacements de caméra
- Avoir « commité » sur le serveur son projet en fin de séance

Bonus :

- Jouer avec la lumière
- Colorer le terrain

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.

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

Votre première application

Afin de vous familiariser avec Qt et de simplifier votre développement, je vous recommande d'utiliser l'IDE Qt Creator. Pour vos premiers pas, avec OpenGL ES, votre travail sera d'étudier l'affichage d'un cube en OpenGL avec Qt et de vous en inspirer pour afficher un terrain.

Question 1

A quoi servent les classes MainWindow et GeometryEngine?

A quoi servent les fichiers fshader.glsl et vshader.glsl ?

Question 2

Expliquer le fonctionnement des deux méthodes

```
void GeometryEngine::initCubeGeometry()
```

et

```
void GeometryEngine::drawCubeGeometry(QOpenGLShaderProgram *program)
```

Question 3

En vous inspirant des deux méthodes précédente, écrivez les méthodes permettant d'initialiser et d'afficher une surface plane (16*16 sommets) composée de triangles.

Pour cela, vous devrez écrire deux nouvelles méthodes

```
void GeometryEngine::initPlaneGeometry()
```

et

```
void GeometryEngine::drawPlaneGeometry(QOpenGLShaderProgram *program)
```

Conseil – commencez par mettre à plat le cube en dessinant les six faces dans un même plan ($z=0$). Modifier ensuite la caméra pour toujours garder les six faces visibles.

Ensuite créer une fonction pour calculer les 16*16 sommets et faces.

Attention à modifier aussi les coordonnées de texture.

Question 4

Modifier l'altitude (z) des sommets pour réaliser un relief.

Déplacer la caméra à hauteur fixe au-dessus du terrain.

Utiliser le clavier pour avancer, reculer, et déplacer la camera de gauche à droite.

Compte rendu

Répondre aux questions.

Expliquer votre raisonnement .

Décrire les problèmes rencontrés et les solutions trouvées.

Expliquer comment vous vous y prendriez pour les parties bonus.

Bonus

- Jouer avec la lumière

Créer une lumière de type soleil.

Créer des lumières localisées.

Jouer avec les matériaux, les modèles de lumières.

- 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).