

HAI911I - Moteurs de jeux – TP1&2

Prise en main de Qt Creator, Git et OpenGL ES 3.0

Noura Faraj ✉ noura.faraj@umontpellier.fr

Objectif

Le but des 2 premiers TPs est de réaliser une application de rendu 3D en utilisant Qt/OpenGL. Dans ce travail, chaque étudiant aura pour but de réaliser différentes tâches :

- Afficher une scène simple (une caméra, un objet)
- 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

Framework

Pour nos différents TP, nous allons utiliser le Framework Qt. Il s'agit d'un Framework spécialisé dans les UI. Nous nous servirons 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. Lorsque votre espace GIT est choisi, alors merci de m'envoyer par mail votre : nom, prénom, groupe sélectionné.

Vous devrez rendre un compte rendu des TP1 et TP2, ainsi que votre code source (moodle et git).

L'application Qt

Afin de vous familiariser avec Qt et de simplifier votre développement, nous vous recommandons 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

Expliquer le fonctionnement les méthodes de dessin et de transformation appliquées aux objets. Quelles sont les mécanismes et fonctions permettant de transmettre à l'application les mises à jour à partir des entrées utilisateur.

Question 2

En vous inspirant des méthodes présentes, 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 permettant de générer la géométrie du plan et de le dessiner à l'aide de shaders.

Conseil - dessinez les triangles dans un même plan ($z=0$). Modifier ensuite la caméra pour toujours garder la surface visible.

Ensuite créer une fonction pour calculer les 16*16 sommets et faces. Plaquez une texture sur la surface.

Question 4

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

Créer un mode de déplacement libre de la caméra permettant d'utiliser le clavier pour avancer, reculer, et déplacer la caméra de gauche à droite.

Question 5

Modifier votre TP précédent pour lire une carte d'altitude (height map) et afficher un terrain dont l'altitude en chaque point est donnée par la carte fournie.

Appliquer des textures en fonction de l'altitude des sommets (grass en bas, rock ensuite et snowrocks pour les sommets les plus élevés).

Question 6

Proposer un autre mode d'affichage pour regarder le terrain sous un angle de 45 degrés, et le faire tourner autour de son origine avec une vitesse constante à l'aide d'un timer. Permettre de passer d'un mode de caméra (libre) à un autre (orbital) en appuyant sur c.

Question 7 (bonus)

Modifier le constructeur de la classe `MainWidget` pour qu'il prenne en paramètre la fréquence de mise à jour (en frames par seconde).

Modifier votre programme principal pour afficher votre terrain dans quatre fenêtres différentes, avec des fréquences de mise à jour différentes (1 FPS, 10 FPS, 100 FPS, 1000 FPS). Qu'observez-vous ?

Utiliser les flèches UP et DOWN pour modifier les vitesses de rotations de votre terrain (vitesse identique dans toutes les fenêtres). Qu'observez-vous ?