Maxime TEIXEIRA-RICCI
Master 2 Informatique IMAGINA

COMPTE RENDU DES TRAVAUX PRATIQUES
MOTEUR DE JEU

Dépot Git : https://github.com/maxime-teixeirarricci/HMIN317.git

—

TP1

09-10-2018

Question 1.

A quoi servent les classes MainWidget et GeometryEngine?

Ce sont deux classes qui permettent d'initialiser et d'afficher une scène 3D avec un cube en utilisant OpenGL.

Pour préciser, la classe « MainWidget » s'occupe d'initialiser la scène, notamment les fonctions « paintGL() » et « resizeGL() » permettent de générer la scène et s'occupent de gérer la camera.

Quant à « GeometryEngine », cette classe s'occupe de la gestion des objets 3D et de la construction de ces objets en gérant les triangles, les textures ainsi que les buffers nécessaire à l'affichage.

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

Le v-shader (Vertex Shader) est un shader qui s'applique sur les sommets (vertex). Le shader travaille sur chaque sommets indépendamment des autres pour appliquer des modifications ou non.

Le f-shader (Fragment Shader, ou Pixel Shader), est un type de shader qui calcule la couleur des pixels qui doivent être afficher.

Question 2.

Expliquer le fonctionnement des deux méthodes initCubeGeometry() et drawCubeGeometry().

La fonction initCubeGeometry() permet d'initialiser les sommets et les indices des triangles afin de générer un cube. Cette fonction génère deux tableaux de sommets et d'indices afin de le passer à la fonction drawCubeGeometry().

Ainsi la fonction drawCubeGeometry() récupère les tableaux afin de pouvoir afficher en fonction du mode d'afficher (ici en utilisant TRIANGLES_STRIP).

Question 3.

Pour faciliter la conception de la surface plane, on utilisera plus le mode de TRIANGLES_STRIP, mais plutôt simplement le mode TRIANGLES. Ainsi on peut facilement créer une surface de 15 faces sur 15 faces et dont chaque face contient 2 triangles.

Question 4.

Pour modifier la hauteur des sommets, on choisit arbitrairement que la hauteur est fonction des coordonnées en X et en Y. (Z = X * Y)

TP2

09-10-2018

Question 1.

Pour réaliser une carte d'altitude (ou Height Map), on utilise une image en nuance de gris qui représente l'élévation des sommets en fonction des sommets. Cette image fonctionne comme une image de texture, sauf qu'au lieu de modifier la couleur des faces ou sommets on modifie leurs positions dans l'espace.

Question 2.

Pour modifier la caméra, on utilise la méthode *lookAt()* de la variable *matrix* dans la classe MainWidget. Cette méthode prends en argument la position de l'œil, le point visé par la caméra et sa normal. Pour mettre la caméra a un angle de 45 de degré, plusieurs points sont intéressant. Les points ou X = Z, X = -Z, Y = Z, Y = - Z sont des points ou la caméra sera à 45 degrés de l'origine de la scène. Pour simuler la rotation du terrain on fait bouger la caméra sur un cercle passant par ces points à une distance de 3 (prit arbitrairement). Ce cercle a pour équation :

X = 3 * sin(t)
Y = 3 * cos(t)
Z = 3

Pour modifier donc la position de la caméra, on doit faire varier la variable t. Pour faire ceci, on ajoutera une certaine quantité à la variable t selon la rapidité que l'on souhaite faire tourner la caméra (ici on ajoutera 0,1 à t à chaque rafraîchissement).

Question 3.

La classe QTimer permet de gérer les timers. Cela permet de rafraîchir la scène avec un certain intervalle (en milliseconde). Ainsi, lorsque l'on veut avoir un rafraîchissement de 30 images par seconde, cela signifie que l'on doit rafraîchir toutes les 1/30 * 1000 = 33,33 ms.

Pour prendre en compte la fréquence de mise à jour de la scène, on rajoute un attribut dans la classe MainWidget et dans le constructeur.

Lors de l'instanciation de cette classe, on pourra mettre en argument le nombre d'images par secondes que l'on veut pour la scène.

Ainsi, on peut donc créer de multiple fenêtres avec différents nombre d'images par seconde. Cela aura pour conséquence de donner l'impression de ralenti si on choisi un nombre d'images par seconde se rapprochant de 0.

Appliquer des textures en fonction de l'altitude des sommets.

Afin de texturer la surface plane selon l'altitude des sommets, on va utiliser une texture de dégradé allant du bleu au rouge.

Cependant, au lieu d'utiliser 2 coordonnées pour utiliser la texture, on utilisera une seule coordonnée, celle en X. Cette coordonnée sera en fonction de la hauteur du sommet en question et sera borné entre 0 et 1 (On définira une hauteur minimale, 0, et une hauteur maximale, ici 0,25f).

Il semblerait qu'il est des soucis vis-à-vis des extrêmes, créant des artefacts visuelles. Pour éviter cela, on évitera l'intervalle [0, 1] mais on privilégiera [0.01 , 0,99] plutôt.