

Compte rendu du TP1 de moteur de jeux

BREUILLÉ Édouard
ROUQUAIROL Célia

28 septembre 2017

1 Question 1

La classe `MainWidget` sert à définir une fenêtre de visualisation, ainsi que tous ses comportements : La disposition de la caméra sur la scène, la gestion des inputs ainsi que l'affichage de la scène.

La classe `GeometryEngine` sert à définir et créer un objet visualisable par le widget en construisant le tableau de points et les reliant dans l'ordre ainsi que de lier une texture correctement à cet objet.

Les fichiers `fshader.glsl` et `vshader.glsl` servent à définir un filtre de couleur à la texture et son rendu 3D (opacité, brillance...)

2 Question 2

La méthode `initCubeGeometry` définit les coordonnées des points d'un cube, le point de la texture auquel il est relié et le tableau d'indice permettant de relier les points dans l'ordre d'un *TriangleStrip*.

La méthode `drawCubeGeometry(QOpenGLShaderProgram *program)` quant à elle, précise les VBOs à utiliser, l'emplacement des textures et dessine ensuite le cube à partir de ces données.

3 Question 3

Ces deux fonctions se trouvent dans le fichier `geometryengine.cpp`. La principale difficulté que nous avons rencontrée est la construction des *TriangleStrips*, qui différait de la méthode que nous avons utilisée l'année précédente en cours de modélisation 3D.

4 Question 4

La modification de l'altitude a été très simple à implémenter : il nous a suffi de choisir un nombre aléatoire entre 2 et (-2) pour que l'écart ne soit pas trop

grand. Le placement de la caméra fut un peu plus complexe : elle se situait dans le dos du terrain. Il nous a donc fallu la replacer devant le terrain et la faire pivoter de 180 afin qu'elle se trouve par défaut face au terrain. La récupération des événements clavier n'a pas posé de problème majeur : lorsqu'une touche est pressée, si elle correspond à une des touches directionnelles, la caméra se déplace sur l'axe correspondant (x : gauche/droite, et y : haut/bas). Nous appliquons également plusieurs effets à la fois sur la caméra : si les touches haut/gauche sont pressées, la caméra se déplacera en diagonale dans cette direction.

5 Jouer avec la lumière

Pour recréer la lumière du Soleil, nous aurions pu utiliser une lumière directionnelle. Pour les lumières localisées (comme une lampe), il existe des lumière de type "point source"

6 Texturer le terrain

Actuellement, le terrain n'est texturé qu'à l'aide d'une simple image. Pour créer des sommets blancs, nous aurions pu utiliser la fonction `glTexEnv(..)` par exemple, et pousser le raisonnement plus loin en éclaircissant les sommets selon leur distance à l'altitude maximale.