

Compte rendu TP1

Question 1 :

La classe mainwidget fournit un ensemble de fonctions permettant le paramétrage des éléments OpenGL tel que la scène, les textures ou les événements souris par exemple.

La classe geometryengine fournit les méthodes permettant de dessiner un cube.

Question 2 :

La fonction initCubeGeometry() permet de créer un tableau de vertices et un tableau d'indices pour ensuite les bind au buffer pour pouvoir être dessiné plus tard. De plus il donne les position pour les texture à y mettre.

La fonction drawCubeGeometry() va lire les VBO pour pour afficher les vertices en prenant en compte la position des points des faces et la position pour les textures.

Question 3 :

Afin de générer une surface plane de 16*16 points, j'ai utiliser une simple double boucle for pour créer les points. Ensuite pour ajoutes les indices des points dans le bon ordre j'ai suivi le pattern suivant : à chaque début de ligne de « Triangle_Strip » je rajoute la premier point de la ligne et en fin de ligne je rajoute le dernier.

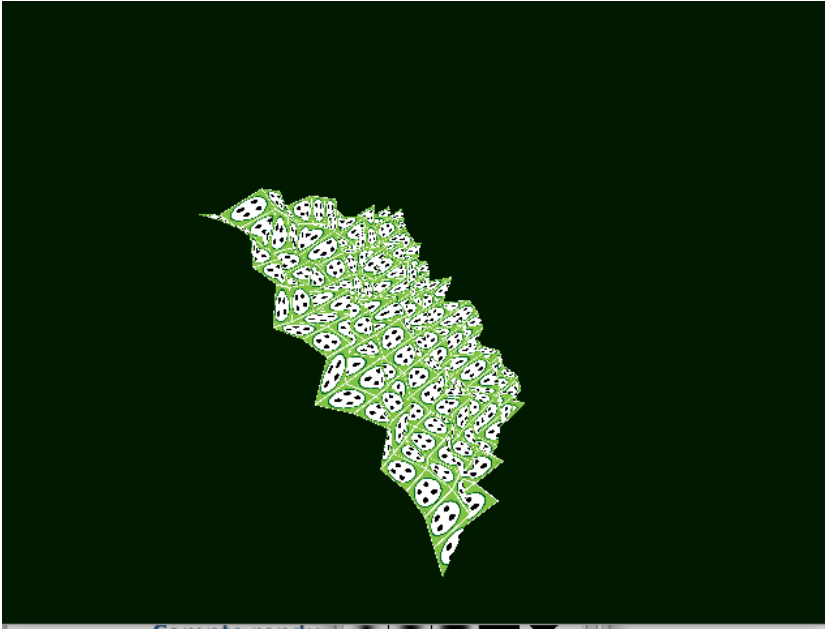
Ensuite de la même manière que drawCubeGeometry() j'ai créer une fonction drawPlaneGeometry() en indiquant le bon nombre d'indices dans le tableau d'indices.

Question 4 :

Pour réaliser un relief j'ai simplement généré un nombre aléatoire pour la composante z de chaque point.

Pour effectuer le mouvement de caméra j'ai enregistré sa position de départ que j'incrémente en fonction des input reçus (flèche du haut, du bas ...), ensuite j'appelle la fonction update() qui va rappeler paintGL(). Dans paintGL() un translate de la matrice est fait en fonction de ses positions.

Résultat :



Problème rencontré :

Qt Creator ne fonctionnant pas il à fallu trouver le moyen d'executer le projet à l'aide de ligne de commandes.

La commande "qmake" pour générer un makefile puis la commande make pour générer un executable.