**HMIN317-tp1**

**Question 1 :**

La classe *MainWidget* permet la gestion des événements (ici de la souris) et l'initialisation d’openGL (scène, shaders, texture…).

La classe *GeometryEngine* permet de charger dans le programme (shader) et d’afficher le cube.

La classe *fshader* est un fragment shader pour définir la texture / couleur de notre objet.

La classe *vshader* est un vertex shader permet de calculer la position des sommets à l’écran et d’appliquer la texture sur l’objet.

**Question 2 :**

La fonction *initCubeGeometry* créé les sommets du cube dans le vertex buffer objects et les six faces du cube à partir des indices des sommets pour l'index buffer objects, il faut ici bien comprendre la façon d'insérer les faces, on peut constater la présence de triangle *dégénéré (qui n'ont pas d'air)* entre chaque face ce qui permet à la carte graphique de ne pas les prendres en compte et de passer à la face suivante.

La fonction *drawCubeGeometry* affiche le cube à l’aide elle s'occupe de bind nos buffer et d'allouer les informations necessaire pour le shader afin de dessiner / texturer notre cube.

**Question 3 et (Bonus 2):**

Relief créer apartir d'une fonction gaussienne pour déterminer l'altitude z de chaque sommet en rouge les la composante *z* d'un sommet est élevé plus la couleur du sommet est rouge pour arrivé a ce résultat je suis passé par le programme shader fournit et j'ai construit mes couleur comme ce ci

Rouge = 0.1+vertexPos.z/6,

Vert = 0.1,

Bleu 0.1+(3-vertexPos.z)/6



**Bonus 1 (Lumières et materiaux):**

Relief créer apartir d'une fonction gaussienne pour déterminer l'altitude z de chaque sommet en rouge les la composante *z* d'un sommet est élevé plus la couleur du sommet est rouge pour arrivé a ce résultat je suis passé par le programme shader fournit et j'ai construit mes couleur comme ce ci

Rouge = 0.1+vertexPos.z/6,

Vert = 0.1,

Bleu 0.1+(3-vertexPos.z)/6