

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* crée 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 prendre en compte et de passer à la face suivante.

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

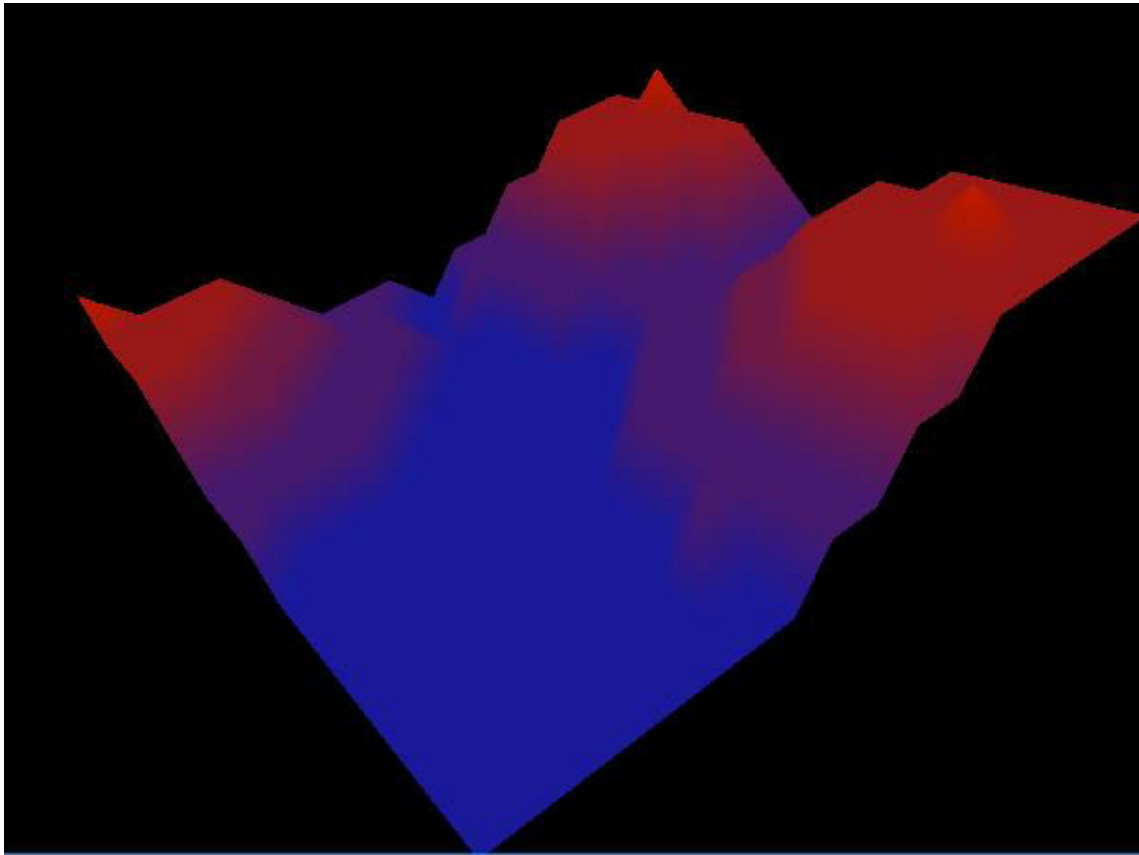
Question 3 et (Bonus 2):

J'ai créé le relief à partir d'une fonction gaussienne pour déterminer l'altitude z de chaque sommet, plus la composante z d'un sommet est élevée plus la couleur du sommet est rouge pour arriver à ce résultat je suis passé par le programme shader fourni et j'ai construit mes couleurs comme ce ci

Rouge = $0.1 + \text{vertexPos.z}/6$,

Vert = 0.1,

Bleu $0.1 + (3 - \text{vertexPos.z})/6$



Bonus 1 (Lumières et matériaux):

Pour générer de la lumière et des matériaux j'ai trouvé deux façons différentes, la première on passe par OpenGL qui propose huit lumières avec différents types de lampe (Lampes directionnelles, Lampes positionnelles, spots...). Et la deuxième que j'ai un peu testé, on la fait soit même grace au shader.

J'ai donc créé une lumière d'ambiance plutôt foncé et deux lumières diffuses avec spéculaires, orangé et bleuté voici le résultat j'ai aussi ajouté un matériel gris clair sur l'objet. Les lumières d'OpenGL sont surrement plus jolies et avec plus d'options mais ayant testé cette technique l'année passé je voulais essayer moi-même.

