Compte rendu TP2

Question 1 :

Modifier votre TP précédent pour lire une carte d'altitude (height map) et afficher un terrain dont l'altitude en chaque point est donnée par la carte.

Pour cela, on commence par charger une image dans un objet Qimage pour pouvoir lire les valeurs de ses pixels en niveaux de gris dans la classe GeometryEngine. On fait alors varier la valeur sur l'axe Z en fonction de la valeur en niveaux de gris des pixels de l'image.
M'étant alors aperçu que l'on chargeait deux fois la même image, une fois pour récupérer ses valeurs et une autre fois dans MainWidget pour texturer le terrain, j'ai décidé de charger une seule fois cette image en mémoire dans la classe MainWidget et de transmettre l'objet Qimage à GeometryEngine.
Problème rencontré : la texture ne correspondait pas aux valeurs de mon affichage (j'avais des pics à des endroits sombres et des creux à des endroits clairs).
Solution trouvée : enlever la fonction "mirrored" appliquée sur la texture pour l'afficher dans le sens correspondant à celui de mes valeurs.

Question 2 :

Modifiez votre fonction d'affichage pour regarder le terrain sous un angle de 45 degrés, et le faire tourner autour de son origine à une vitesse constante.

Dans la classe MainWidget, on commence par positionner la caméra au dessus de l'objet sur l'axe z. Puis, on applique une rotation de 45 degrés sur l'axe x pour que la caméra soit bien positionnée.
Il suffit ensuite de faire tourner l'objet sur lui-même. Pour cela, j'ai décidé de le faire tourner sur l'axe z en lui spécifiant une vitesse de rotation constante.
Problème rencontré : le terrain s'arrête de tourner au bout d'un certain temps.
Solution trouvée : enlever la friction dans la fonction timerEvent.

Question 3 :

Comment est contrôlée la mise à jour du terrain dans la classe MainWidget ? A quoi sert la classe QTimer ? Comment fonctionne-t-elle ?

L'appel de la fonction "update" dans la fonction "timerEvent" permet d'indiquer au programme qu'il doit rappeler la fonction "paintGL" et ainsi rafraichir l'affichage et redessiner le terrain.
La classe Qtimer permet de répéter une action au fil du temps, dans notre cas pour adapter la rotation du terrain à intervalle de temps régulier. Elle fonctionne en émettant un signal tous les "x" temps – valeur indiquée par la fonction "setInterval(int msec)" – et démarre avec la fonction "start" puis s'arrête à l'aide de la fonction "stop".

Modifier le constructeur de la classe MainWidget pour qu'il prenne en paramètre la fréquence de mise à jour (en frames par seconde).

Modifier votre programme principal pour afficher votre terrain dans quatre fenêtres différentes, avec des fréquences de mise à jour différentes (1 FPS, 10 FPS, 100 FPS, 1000 FPS). Qu'observez vous ?

L'image à 1 FPS n'est pas fluide, on n'a pas l'impression de voir une animation.
L'image à 10 FPS n'est pas fluide, on voit bien une animation mais très saccadée.
Les images à 100 et 1000 FPS sont fluides, on observe bien la rotation du terrain mais on ne distingue pas de différence entre les deux.
On note cependant que les rotations sont décalées dans le temps. Cela s'explique par le fait que les fenêtres sont lancées les unes après les autres et qu'aucune synchronisation n'est effectuée.
Or, ce décalage ne pose pas de problème car : "Le but de ce TP est d'afficher une scène dans plusieurs fenêtres et de contrôler **séparément** l'affichage de chaque fenêtre."

Utiliser les flèches UP et DOWN pour modifier les vitesses de rotations de votre terrain. Qu'observez vous ?

Avec une vitesse suffisamment faible, même les images à 1 et 10 FPS paraissent fluides.
Si la vitesse passe en négatif, on inverse le sens de rotation.
Si la vitesse est trop importante, on perd la fluidité de l'animation des images à 100 et 1000 FPS.

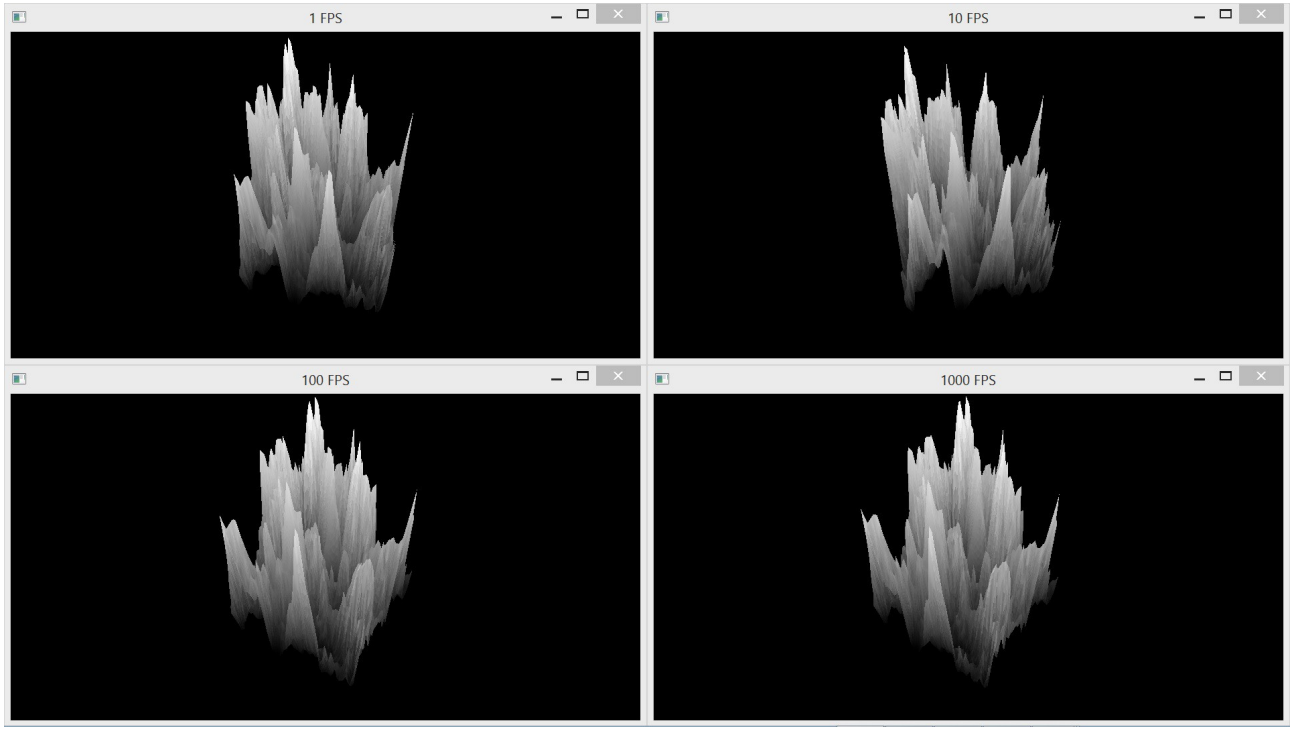

Illustration 1: TP2 - 1, 10, 100 et 1000 FPS

Bonus :

Jouer avec la lumière.

Pour créer une lumière de type soleil, j'ajouterai une seule source lumineuse quelque part dans mon monde. Je ferai en sorte qu'elle éclaire suffisamment l'objet qui m'intéresse dans la scène et qu'on puisse voir les ombres et/ou les reflets.
Pour créer des lumières localisées, j'ajouterai plusieurs sources lumineuses en activant les lumières de 0 à 8 selon mes besoins et adapterai leur couleur ainsi que leurs coordonnées.
Pour jouer sur les matériaux et les modèles de lumières, je jouerai sur la réflexion des matériaux et sur l'aspect ambiant, spéculaire ou diffuse.

Texturer le terrain en utilisant des couleurs.

Pour texturer le terrain à partir de couleurs, je stockerai un tableau de couleurs et un tableau d'indices correspondant chacun à une couleur pour chaque sommet du terrain. Ensuite, au moment de dessiner le terrain, la face correspondrait aux différentes couleurs des sommets convergentes en une couleur commune au centre de la face.
Pour adapter la couleur à l'altitude, chaque sommet aurait ainsi une couleur dépendant directement de sa coordonnée en z.