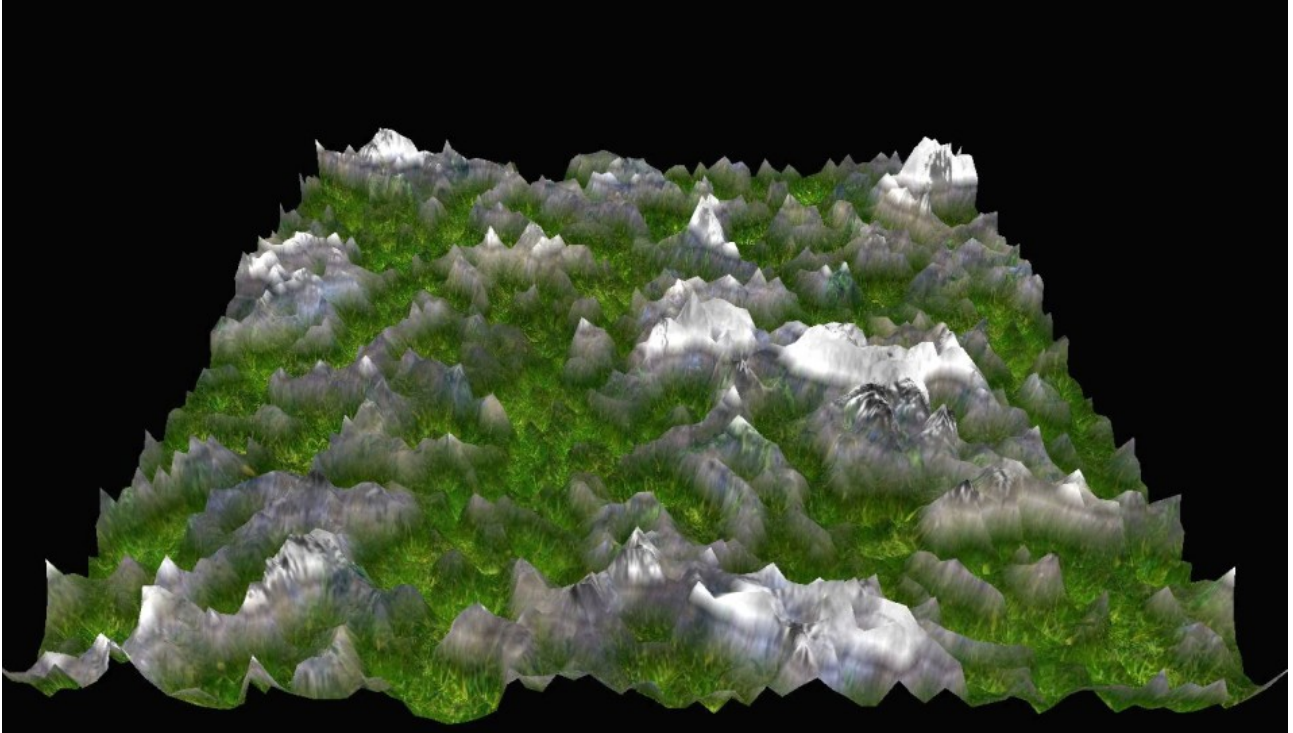


TP 1 & 2
Melvin BARDIN

LIEN DU GIT :

<https://github.com/Kyrial/TP-Application-Interactive>

Resultat finale obtenu :



Question 1

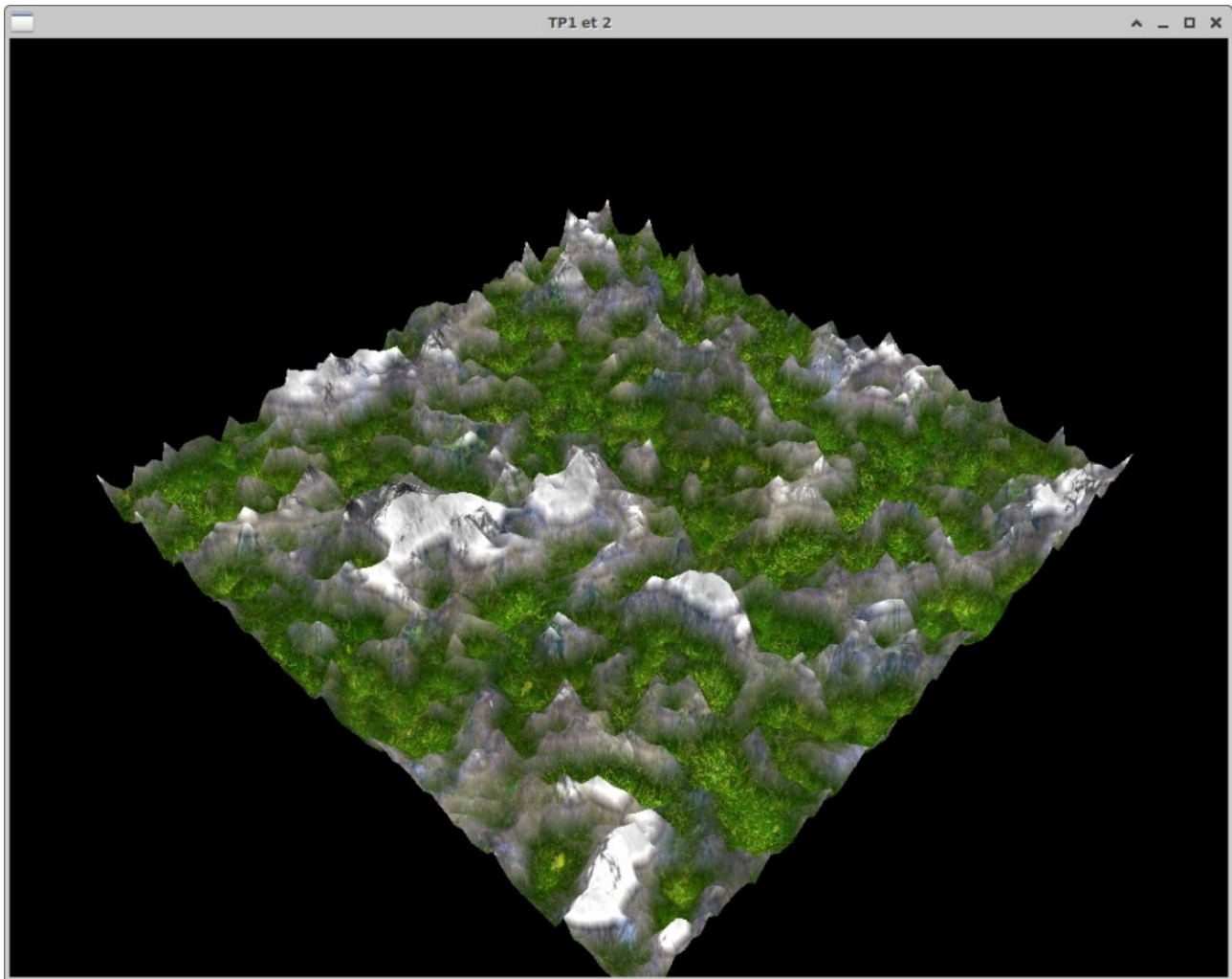
Expliquer le fonctionnement des méthodes de dessin et de transformation appliquées aux objets. Quelles sont les mécanismes et fonctions permettant de transmettre à l'application les mises à jour à partir des entrées utilisateur.

Les mécanisme nécessaire permettant de transmettre à l'application les mise à jour a partir des entrés utilisateur sont provoqué dans la mainLoop. A chaque passage dans la mainLoop, les événement sont interceptés, la vitesse de boucle diffère en fonction des machines, c'est pour cela qu'il faut fixé les modification en fonction du temps écoulé entre chaque boucle.

Les méthode de dessins : création (ou importation) des mesh, bind et envoie dans le Vertex shader. Dans ce vertex shader j'ai modifié la hauteur en fonction de la texture de la carte des hauteurs. J'ai ensuite envoyé la hauteur dans le fragment shader pour qu'il applique une texture en fonction de la hauteur.

Transformation appliqué aux objets : c'est la matrice MVP qui gère cela.

La carte tourne en appuyant sur C, et s'arrête lorsqu'on réappuie.



Question 7 (bonus)

Utilisation de `QelapsedTimer` pour avoir une animation homogène peu importe le nombre de image par seconde et du `deltaTime`

2 fenêtre s'ouvre, l'une avec un rafraîchissement de l'image de 20milliseconde, l'autre avec 500milliseconde.

On peut remarqué que sur la première, l'animation est fluide tandis que la deuxième l'image est saccadé, cependant grâce à `QelapsedTimer`, malgré que le nombre d'image par seconde est différente, l'animation va à la même vitesse.

Ainsi, le nombre d'image par seconde n'influence pas la vitesse de l'animation.