

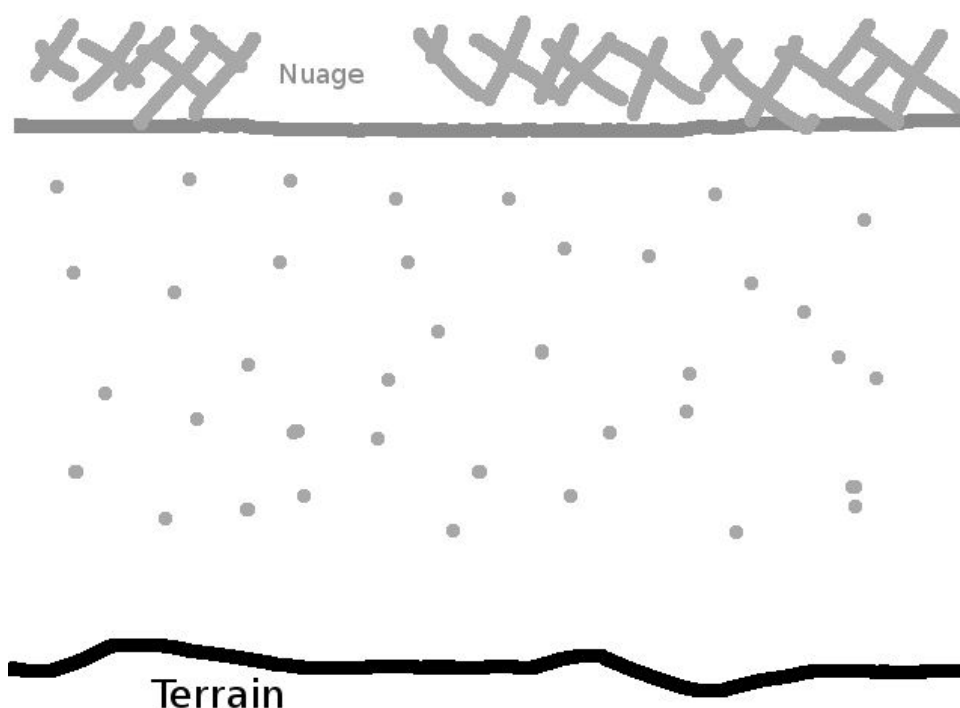
Compte rendu TP3

Arnuaad Sanchez - Thomas Rampin

Pour la synchronisation de la fenêtre avec la saison, nous utilisons un `QTimer()` que nous connectons à la fonction `changeSeason()` d'un objet `SeasonManager` qui gère l'état actuel des saisons d'une fenêtre. Les saisons changent après un certain temps défini.

Ce changement de saison entraîne une évolution des textures qui vont agrémenter notre terrain. En effet, nous changeons les textures pour l'hiver. Nous chargeons les textures correspondant à des montagnes enneigées.

Une fois cette étape réalisée, nous nous sommes penchés sur la génération de particules. De la neige pour l'hiver, et de la pluie pour le printemps ou l'automne.

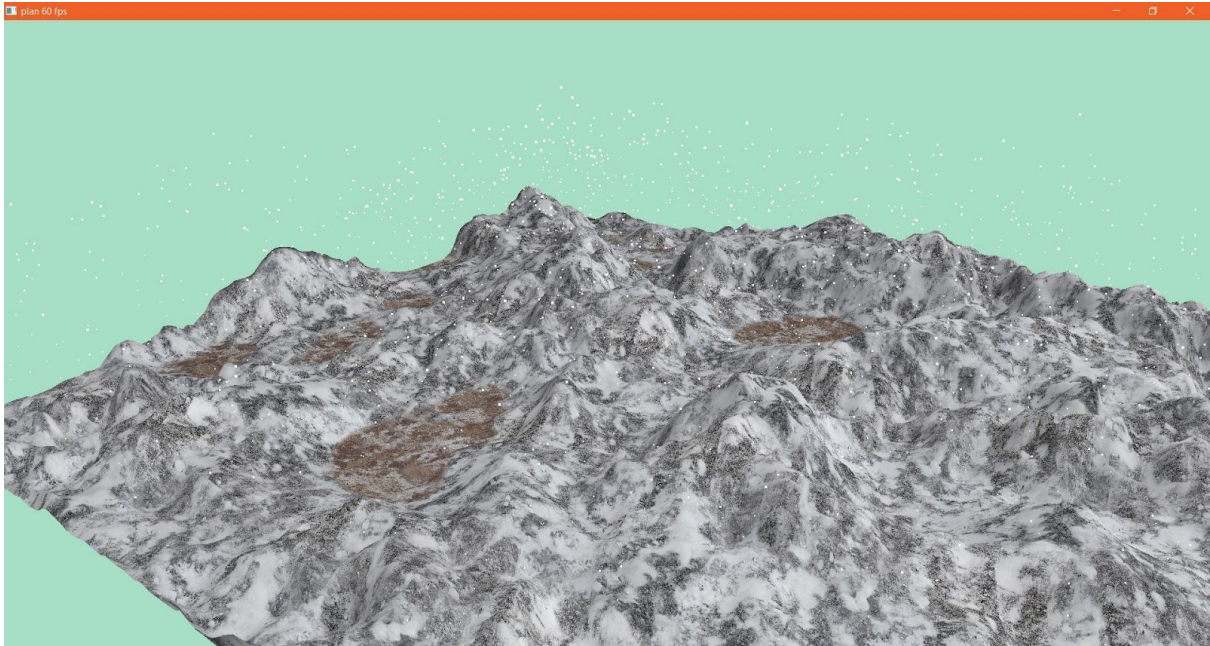


Nous avons simulé un nuage au dessus de notre terrain. Cela permet d'avoir un système de particules qui se rapproche de la réalité. Nous utilisons un système aléatoire qui permet une génération des particules homogène dans l'espace du nuage. Nous donnons à nos particules un vecteur vitesse qui permet une chute de celles ci plus réaliste, avec une oscillation. Nous calculons notre vecteur vitesse grâce à des coordonnées sphériques.

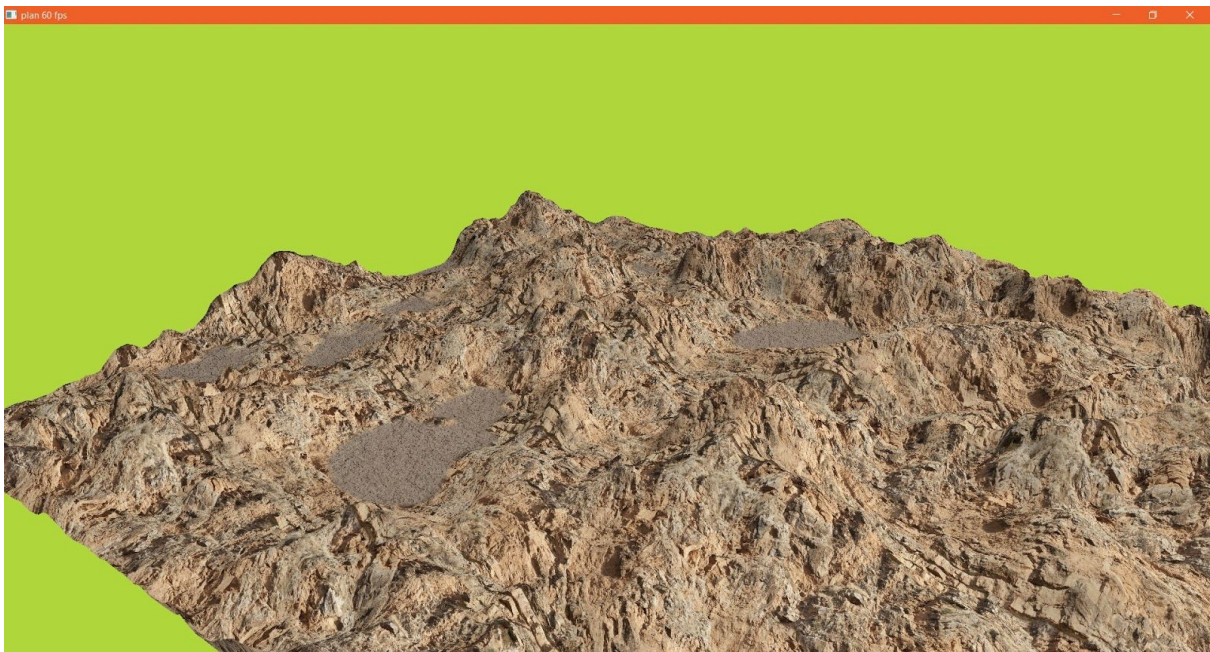
La position de départ, des particules, est calculée du côté CPU, tout comme la vie. Cependant, nous gérons la disparition des particules dans les shaders. Nous utilisons des shaders différents pour le plan et les particules, on passe d'un shader à l'autre suivant ce que l'on veut afficher.

Concernant les structures de données, nous utilisons un tableau qui contient toutes les particules. Celui ci fait donc une taille `MAX_PARTICULES`. Nous regardons à chaque frame si il y a une place de libre dans ce tableau, et nous mettons à jour les particules de celui ci.

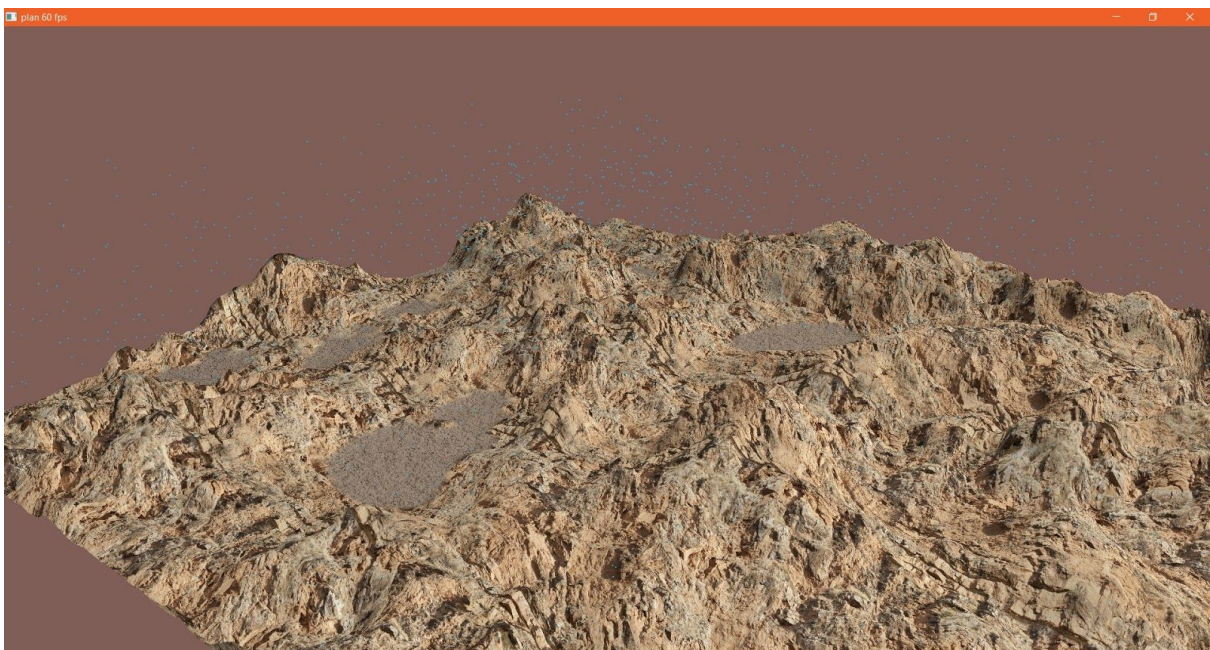
Pour visualiser la transition entre saisons plus clairement nous avons mis en place un système de changement de couleur du fond de l'application. Une couleur est définie pour chaque saison et à l'aide d'une interpolation linéaire entre chaque couleur pour obtenir une transition fluide du fond entre chaque saison.



Plan en hiver avec particules de neiges



Plan au printemps



Plan en Automne avec particules de pluies

On effectue aussi un calcul des normales pour un meilleur éclairage de notre terrain, on utilise la heightmap utilisé pour générer le terrain pour calculer la normal d'un vertex en utilisant les données de la heightmap pour le point au dessus, dessous, à gauche et à droite. On calcule ensuite un vecteur lumière entre la position de notre lumière et celle du vertex. Puis on calcule le produit scalaire du vecteur lumière et de la normale pour l'utiliser dans le fragment shader afin de calculer la couleur du vertex.

Pour les bonus:

- Pour accumuler les particules on peut calculer à l'endroit où elles tombent une modification de la height map. Cela la fera augmenter avec l'accumulation puis réduire lors de la fonte des neiges.
- Pour la localisation des effets climatiques, nous avons mis en place un système qui permet d'avoir un rendu différent de la neige, notamment grâce au côté aléatoire de la génération des particules.
- Pour améliorer la création des particules, on peut implémenter un système de Pool afin de réutiliser des particules mortes au lieu d'en créer de nouvelles afin d'économiser de la mémoire et du temps de calcul.