

Compte-rendu TP3

Les saisons :

Les saisons sont simplement liées au timer du widget, en fonction de celui-ci on change la couleur de la skybox et celle de la couleur ambiante du rendu de Phong. En plus de cela, durant l'hiver, pour simuler l'accumulation de neige, la couleur devient plus blanche, et les normales sont uniformisées. Cette simulation de neige à plus tard été retirée pour laisser place à une accumulation basée sur le système de particule.

Les particules :

Le système de particule est implémenté par des shaders, des textures servent de mémoire.

Les textures : elles sont en 256x256 (taille identique au terrain), dans les coordonnées RGB, on stocke les coordonnées XYZ de chaque particule, et en A on stock une variable pour savoir si on affiche la particule, ce qui permet de faire varier le nombre, donc la densité de la pluie/neige.

Une autre texture sert à mémoriser l'accumulation de la neige sur chaque point du terrain, on utilise une seule coordonnée de celle-ci.

Les shaders : un des shader sert donc à l'animation, un autre au rendu des particules. Le shader d'animation « particleEngine » fait tomber les particules.

Le shader d'animation particleEngine, met à jour la texture de position des particules à chaque frame. Un random est fait sur la position de la particule dans une colonne $1/256^2$ quand elle est générée.

Le shader terrainEffect gère les effets résultants des particules : il vérifie l'impact des particules et met à jour en conséquence la texture d'accumulation. Le flux de l'eau est modélisé par la variation du niveau de l'eau en fonction de la hauteur du terrain et de la hauteur de l'eau à la frame précédente sur les points voisins.

Le rendu est fait en mappant une texture sur le point correspondant à la particule dans les shaders particleRender Fragment et Vertex.

La modification du terrain par accumulation de neige ou d'eau est faite dans le shader de rendu du terrain en utilisant la texture d'accumulation de la neige.

Pour paralléliser les différents calculs liés aux particules il nous faudrait soit utiliser openCL ou sur une machine avec une architecture OpenGL récente (≥ 4.3) on peut utiliser un compute shader pour utiliser toute la puissance hautement multithread des GPU.

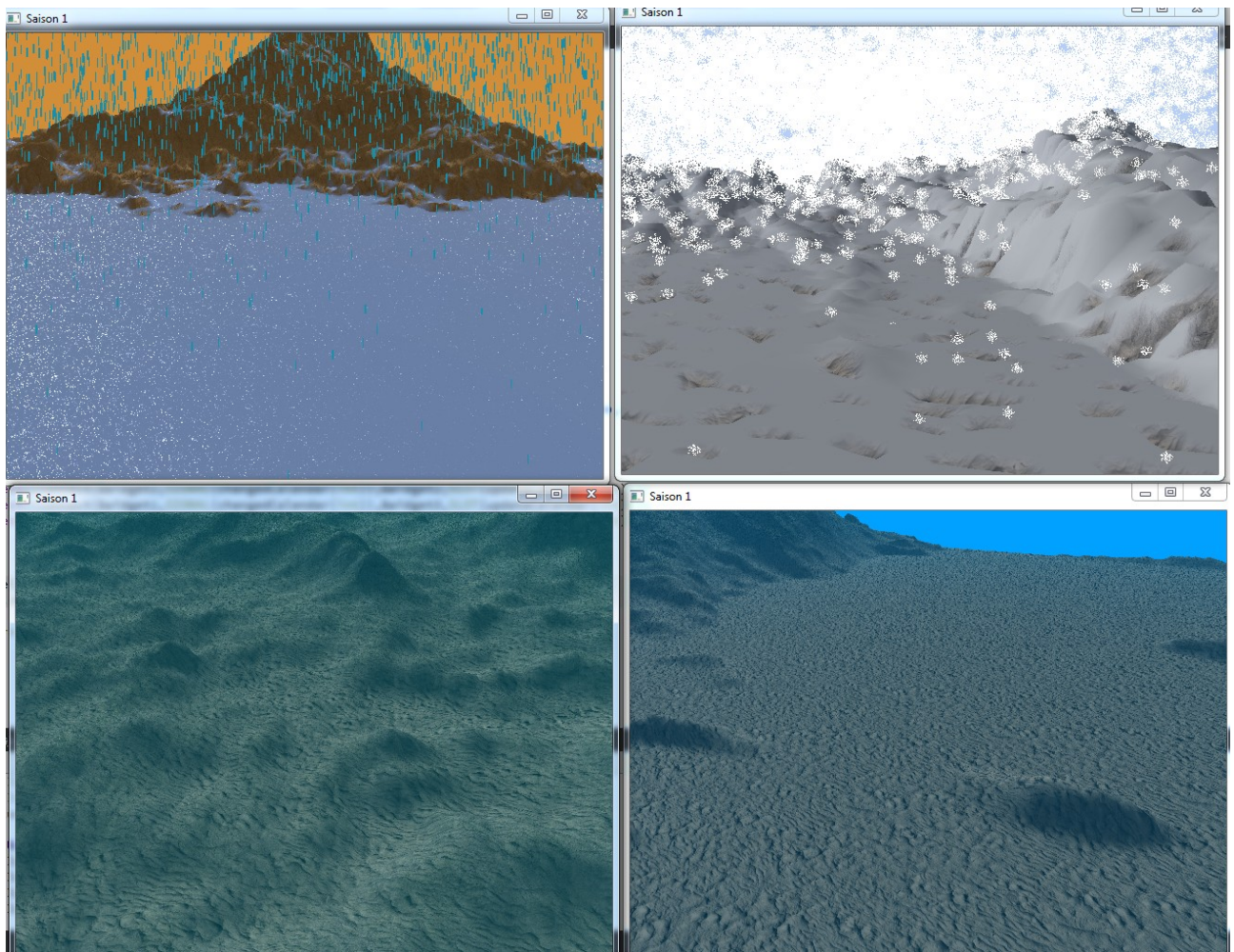


Illustration 1: Les 4 saisons