Decroos Kevin

Moteurs de Jeux
Compte Rendu de TP3 et 4

https://github.com/KevinCDec/tp_moteurs_de_jeux

Synchroniser les fenêtres entre elles

Le système de synchronisation fonctionne comme prévu avec des signaux.

Il a tout simplement fallu utiliser le système de signal et de slots. J'ai donc créé une classe calendar disposant d'un timer et d'un événement sur celui-ci qui envoie un signal. J'ai du ensuite adapter les mainwidgets pour recevoir le signal et prendre les conséquences (écrire un message). Cela a du ensuite être connecté dans le main.

Simuler les changements de saisons

Le changement des saisons a été effectué grâce à des instructions pour les shaders, en mettant une couleur en particulier par un vecteur en produit des textures. Le vecteur est envoyé depuis paintGL() qui « set » une variable de vshader.

Intégrer un quadtree

Le quadtree a semblé être un exercice intéressant et je disposais d'idées, mais pour une raison qui m'est absolument inconnue malgré mes tentatives, je n'ai pas réussi à transmettre de VertexData correctement autrement que par les méthodes déjà utilisées. Mon objectif était de les faire créer par une classe quadtree avec les indices, mais je n'ai jamais réussi à afficher cela, même avec un simple carré et en repassant par geometryengine directement. Faute de pouvoir tester quoique ce soit, je n'ai pas pu avancer.

L'idée était d'utiliser quadtree de manière récursive (un arbre normal), et de spécifier les niveaux de détails par une profondeur demandée dans les étapes de la construction des triangles. Le niveau de détail serait donc géré dès l'initialisation.

Les dossiers du tp3 et 4 représentent une version des saisons qui fonctionne (tp3), plus des tentatives sur les bases du quadtree.

Suite à cet échec, rien d'autre n'a été accompli. Ma compréhension de ce qui touche à OpenGL et de ce qu'on peut/doit faire avec est beaucoup trop floue pour pouvoir avancer à partir de ces problèmes.