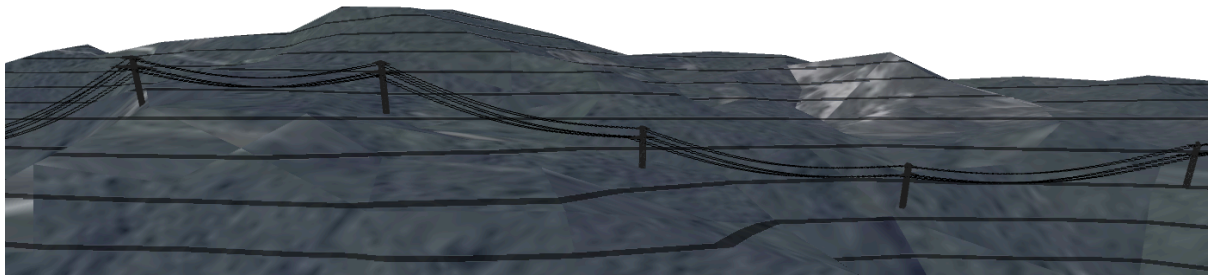


Projet info228 IGSD

Des pylônes dans les collines



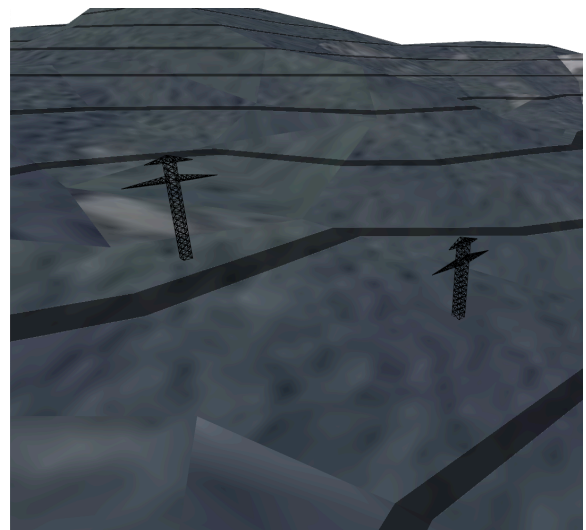
Le but de ce projet est de réaliser une application 3D qui affiche un terrain texturé représentant une vallée entourée de collines et dessiner par dessus des pylônes et lignes électriques (type haute tension).

1 Le terrain

Votre première tâche consistera à charger et afficher le terrain ainsi qu'un repère composé des 3 axes x , y et z . Le modèle fourni est composé de points entre -202.09592 and -179.59933 sur l'axe z . (Et entre -135 et 127 sur l'axe x et entre -158 et 159 sur l'axe y). A vous de placer la caméra et de gérer des déplacements de caméra permettant de placer son point de vue. Vous pourrez utiliser le modèle nommé **HYPERSIMPLE** dans votre espace ecampus. Le modèle **SIMPLIFY** est plus gros et nécessite d'augmenter la mémoire dans les paramètres de processing. Il est aussi plus long à charger et ralentit le développement.

2 Shader

Votre seconde tâche consiste à utiliser un shader qui affiche des lignes de niveau. Pour cela il faudra passer le z du modèle au fragment shader. Le fragment shader pourra ainsi regarder le z interpolé et si le reste de la division entière est compris entre 2 bornes il pourra rendre le pixel noir (ou plus foncé). Si les z étaient en mètres (ce qui n'est pas le cas) on pourrait tester si $z \bmod 100$ est compris entre 0 et 1. Le shader pourra aussi teinter la vallée (tous les pixels dont le z interpolé est en dessous/dessus d'une certaine valeur en vert.



3 Des pylônes

Votre troisième tâche consiste à créer un pylône que vous construirez en gris foncé segment par segment dans un modèle PShape. (L'unique modèle sera ensuite affiché à plusieurs endroits la scène). Pour pouvoir placer les pylônes au bon endroit, vous devrez créer une fonction qui cherchera le z (l'altitude) du modèle pour un x,y donné. Les fonctions `getChildCount()`, `getChild()`, `getVertexCount()` et `getVertex()` permettent de parcourir tous les points du terrain et de chercher celui dont le x,y est le plus proche du x,y demandé (et de retourner le z).

4 Des lignes électriques

Votre quatrième tâche consiste à créer un modèle de lignes électriques qui suit les points d'attache sur les pylônes. Vous créerez une fonction qui place N (ie =25) pylônes entre 2 points en paramètre (ie entre 20,100 et 40,-115) et qui crée les lignes électriques entre. Ces lignes devront fléchir par effet de la gravité. Plus la distance entre 2 pylônes sera grande plus les lignes devront fléchir vers le bas. Les "bras" des pylônes qui supportent les lignes devront rester perpendiculaires à la ligne. Les touches du clavier serviront à afficher/cacher chaque modèle.

5 Bonus : Les éoliennes

Pour les binômes, nous attendons un travail supplémentaire. Vous devrez créer un modèle animé d'éolienne avec des courbes de Bézier, que vous placerez en haut des collines et qui seront reliés à la ville qui se trouve dans la vallée en $x=20, y=100$ par une ligne électrique.

Le projet (fichier pdf du rapport, pde processing, fichier glsl MAIS pas le modèle 3D !) sera à rendre dans un fichier zip à déposer sur ecampus pour le 29 avril 2024. Le rapport devra expliquer les fonctionnalités, la structure du code, le fonctionnement des shaders et des difficultés rencontrées. La semaine du 29/04 vous ferez une soutenance de votre projet devant l'enseignant de votre groupe de TP.

