

Rapport de Projet

Implémentation d'un visioneurs de fichiers gltf

April 4, 2021

Contents

1	Généralités	1
1.1	Limitation des FPS	1
1.2	Modes d'affichage	1
2	Normal Maps	1
2.1	Modèles possédant des tangentes	1
2.2	Modèles sans tangentes	2
2.3	Calculer les tangentes à la volée	2

1 Généralités

1.1 Limitation des FPS

1.2 Modes d'affichage

2 Normal Maps

2.1 Modèles possédant des tangentes

Les modèles 3D peuvent être exportés avec leurs tangentes, ce qui se traduit par un attribut “TANGENTE” supplémentaire pour les primitives des fichiers gltf. Dans ce cas, il suffit de traiter cet attribut comme les autres dans `createVertexArrayObject`.

Une fois que l'on a accès aux tangentes dans le shader, on peut procéder comme dans ce tutoriel.

Un problème rencontré est le manque de standard pour les normal maps. En effet, certains utilisent des normal map dont les bits peuvent être signés

et non-signés, certains représentent le vert comme la direction vers le haut ou vers le bas, voir même certains n'incluent pas de coordonnée verte, qui doit alors être déduite des deux autres (le vecteur normal étant alors supposé normé). Il n'y a pas de moyen d'y remédier automatiquement. Aussi, j'ai rajouté des options pour l'interprétation des normal maps.

2.2 Modèles sans tangentes

On pourrait précalculer les tangentes des modèles. J'ai tenté de faire cela, mais je n'ai pas eu le temps d'obtenir quelque chose de viable. J'avais néanmoins une alternative déjà implémentée qui fonctionne.

2.3 Calculer les tangentes à la volée

Le fragment shader permet de donner accès à dérivée en x et en y des variables, ainsi on peut avoir accès aux dérivées des coordonnées de textures ainsi que de position dans le viewspace. Avec la capacité de différentier des variables, on a la possibilité de déduire les tangentes dans le fragment shader. C'est la méthode présentée par cet article, j'ai adapté le codé présenté, et le résultat est fortement concluant.

