

Rapport Projet GLTF Viewer

Sujet choisi : Normal Mapping

Jérémy VONG - M2 SIS

Préambule :

Pour le cours de Synthèse Image Avancée, il est demandé de réaliser un projet qui suit le TP réalisé. Ce projet possède plusieurs sujets possible présenté ici :

<https://gltf-viewer-tutorial.gitlab.io/project/>

Lancer le programme :

Le programme se lance avec les commandes ci-dessous dans l'ordre indiqué :

```
source .vscode/bash-init.sh
```

```
cmake_prepare
```

```
cmake_install
```

```
dist/gltf-viewer viewer gltf-sample-models/2.0/[MODEL_NAME]/glTF/[MODEL_NAME].gltf
```

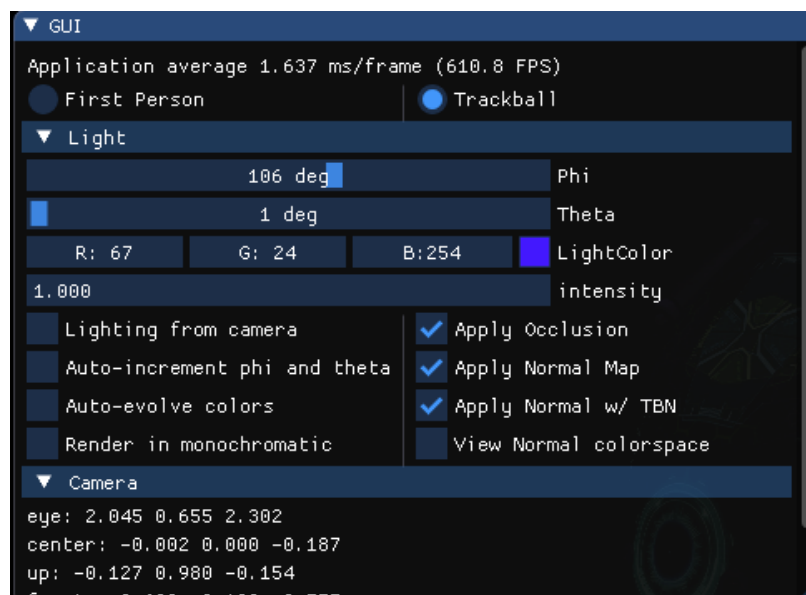
(Pour obtenir les modèles utiliser cette commande :

git clone <https://github.com/KhronosGroup/glTF-Sample-Models> gltf-sample-models)




Spécificités du viewer et utilisation :

Le viewer suit les mêmes spécificités introduites dans le TP, la seule différence est l'usage de MIDDLE-CLICK + LEFT-ALT au lieu de MIDDLE-CLICK + LEFT-CONTROL car le programme ne reconnaît pas la touche CTRL de mon PC portable.

La GUI



Au delà des boutons introduits dans le TP, voici une description des boutons rajoutés.

Auto-increment phi and theta <i>Va automatiquement faire varier Phi et Theta entre leur min et max a des vitesses aléatoires et changeantes lorsqu'ils atteignent un extrema.</i>	Apply Normal Map <i>Applique une simple normal Map sur l'objet</i>		
Auto-evolve colors <i>Va automatiquement faire les couleurs entre 0 et 255 a des vitesses aléatoires et changeantes lorsqu'ils atteignent un extrema. (donne un effet disco)</i>	Apply Normal Map <i>Applique la normal Map sur l'objet avec le calcul de la matrice TBN</i>		
<table border="1"> <tr> <td data-bbox="201 842 612 1037"> Render in monochromatic <i>Binarise le rendu en une couleur</i> </td><td data-bbox="612 842 786 1037">  </td></tr> </table>	Render in monochromatic <i>Binarise le rendu en une couleur</i>		View Normal colorspace <i>Réalise le rendu sur les normals obtenus (enlève donc les textures/metals/couleurs/etc..)</i>
Render in monochromatic <i>Binarise le rendu en une couleur</i>			

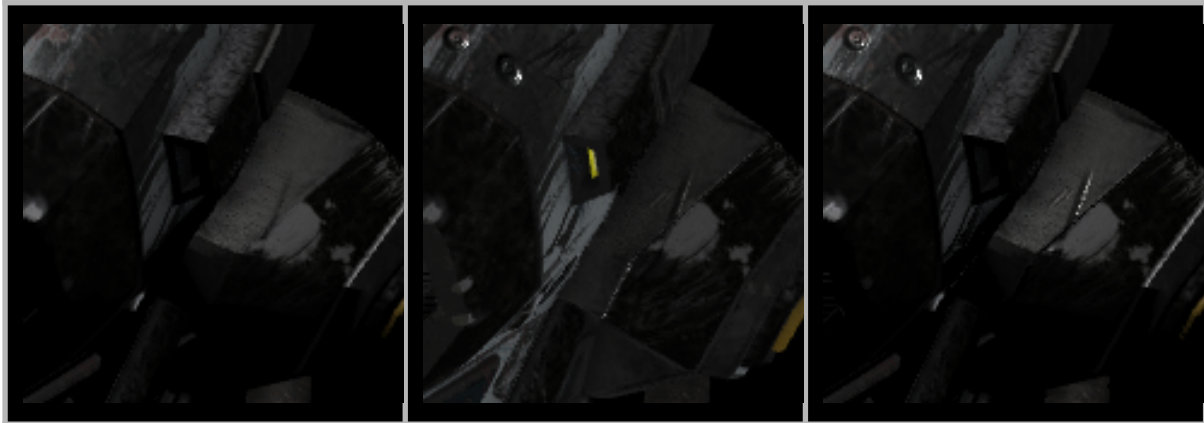
Normal Mapping

Pour ce projet, j'ai choisi le Normal Mapping, car je pense qu'il s'agit de la technique la plus utile pour améliorer le rendu sans pour autant trop alourdir les performances.

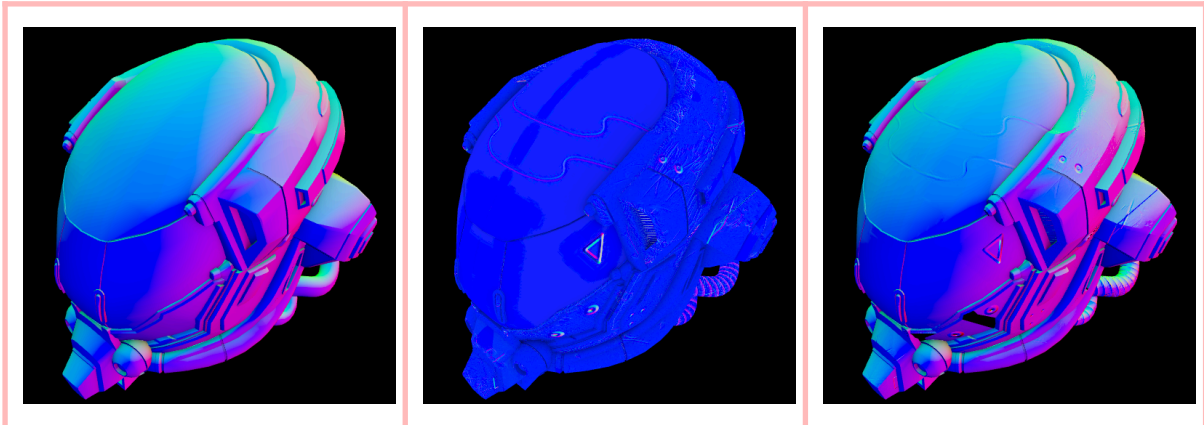
Résultat obtenus :



Avec un zoom voici l'évolution des détails



Un bouton permet aussi d'afficher la vue des normales (avec ou sans occlusions)



On peut remarquer un patch noir dans la dernière image qui apparaît lors de l'ajout du TBN, dont je ne connais pas l'origine et n'ai pas pu le résoudre.

Difficultés rencontrés :

La compréhension même de glTF, dans les spécifications de glTF il y a déjà un attribut Tangent alors que l'énoncé indiquait l'implémentation de ceux-ci.

Le débogage est une étape très compliquée, l'impossibilité de connaître les valeurs passées dans le fragment shader rend la tâche très compliquée pour identifier le problème. Le passage des données du modèle par glTF ne rendant pas la tâche plus facile.

Il est possible que des problèmes résolus dans le projet ne l'ont pas été dans le rendu TP.

Connaissances Acquises :

Grâce à ce sujet, j'ai pu comprendre plus en profondeur le calcul du Normal Mapping. Ma connaissance de base étant simplement visuel de la normal Map, je n'avais jamais eu en tête l'idée de l'espace tangent. Avec ce sujet j'ai pu m'appliquer à l'implémentation de la Normal Map, me heurtant à divers problèmes j'ai pu approfondir ma compréhension de glTF.

J'ai pu comprendre que même si glTF possède dans ces spécifications l'attributs tangent, les modèles en eux-mêmes peuvent ne pas l'avoir informé.