

UNIVERSITE DU QUEBEC A CHICOUTIMI

6GEN715 – INFOGRAPHIE

Travail 2 – partie 2

Programmation GLSL et interactions avec l'utilisateur

Date de remise de l'anneau de l'étape g:

Pour le groupe du jeudi : 17 octobre 2018

Pour le groupe du lundi : 21 octobre 2018

1- OBJECTIFS

- Familiariser l'étudiant au développement de « shaders » dans le langage GLSL.
 - Familiariser l'étudiant aux mécanismes permettant des interactions avec l'utilisateur dans le langage Javascript.
-

2- MODALITÉ PARTICULIÈRE

Ce travail doit être réalisé individuellement.

3- TRAVAIL À RÉALISER

a) Cliquez sur le lien suivant et déplacez votre souris dans le coin supérieur gauche de la page HTML qui sera montrée. La position de la souris devrait y être affichée.

- display-mouse-position.html

Étudiez attentivement le programme Javascript qu'il contient.

b) Téléchargez ensuite les fichiers suivants :

- rotationsouris.html
- rotationsouris-a-corriger.js

c) Le code du fichier Javascript a été volontairement « saboté ». Lorsque vous cliquez le bouton gauche de la souris et que vous déplacez ensuite cette dernière, le cube affiché tourne mais de manière quelque peu chaotique.

- d) Étudiez le code présent dans le fichier HTML ainsi celui présent dans le fichier Javascript. Notamment :
- Observez la présence ligne « `<div id = "display"> ***</div>` » dans le fichier HTML. Étudiez (dans le fichier Javascript) comment les valeurs d'angle sont affichées.
 - Étudiez le fonctionnement des fonctions de rappel associées aux événements de la souris (dans le fichier Javascript).
- e) Sachant que **seulement 3 erreurs** ont été insérées dans le fichier Javascript et que ces erreurs ne touchent que des signes d'opération (+ et -), corrigez le fichier pour que le cube tourne « docilement » sous le contrôle de la souris. Il est à noter que des signes peuvent avoir été modifiés (par exemple, un + peut avoir été remplacé par un -, ou vice versa) ou avoir été ajoutés. **Deux erreurs se trouvent dans la fonction *render()* et une autre dans une fonction de rappel gérant la souris.**
- f) Une fois les trois erreurs corrigées, modifiez le « vertex shader » et le programme Javascript de sorte que la matrice de rotation provienne du programme Javascript (via une variable « uniform » nommée *matricemodelisation*). Vous devrez donc **retirer les définitions de rx, ry et rz** du « vertex shader » (vous n'en aurez plus besoin). Votre programme Javascript devra donc composer une matrice de rotation combinant une rotation autour de l'axe X et une rotation autour de l'axe Y. Ainsi, dans le « vertex shader », les lignes suivantes devraient être présentes :

```
tmp_position    =  matricemodelisation * vPosition;
tmp_position.z  = -tmp_position.z;
gl_Position     =  tmp_position;
```

- g) Finalement, modifiez une fois de plus le programme Javascript pour qu'il affiche l'anneau que vous avez réalisé dans la partie 1. Votre anneau devra pouvoir tourner sous le contrôle de la souris. Pour chaque élément de votre anneau, vous devrez composer une matrice de modélisation un peu plus complexe. Celle-ci devra être le résultat de la multiplication de la matrice de rotation développée au point (e) par une matrice fixant la taille, la forme et la localisation de chaque élément. **Cette matrice devra être composée dans le programme Javascript et ce, pour chaque élément de l'anneau.**

Note : les lignes suivantes (utilisées dans la partie 1) **ne devra plus apparaître dans le vertex shader**. Vous devez composer, dans votre programme Javascript, une matrice réalisant la même tâche.

```
newPosition.x = vScale.x * vPosition.x + vDisplacement.x;
newPosition.y = vScale.y * vPosition.y + vDisplacement.y;
newPosition.z = vScale.z * vPosition.z + vDisplacement.z;
newPosition.w = vPosition.w;
```

Au final, vous aurez développé un programme rudimentaire permettant l’affichage de molécules. Évidemment, il ne sera pas aussi raffiné que ceux montrés dans les exemples suivants :

- <http://glmol.com/>
- <http://web.chemdoodle.com/demos/molgrabber-3d>
- http://www.uqac.ca/daudet/Cours/Infographie/DOCUMENTS/repertoire435/mrdo-ob-threejs/examples/#css3d_molecules

Cependant, vous pourrez bientôt modifier votre programme pour atteindre le même niveau de qualité. Pour ce faire, nous devons avoir vu le modèle de Phong et comment dessiner des sphères et des cylindres.

L’exemple suivant présente un exemple de « visualisateur » d’objets 3D plus élaboré :

- <http://www.finalmesh.com/webgl/viewer/>

4- RAPPORT

S.V.P. Nommez votre fichier ZIP en indiquant qu’il s’agit de la partie 2 (Exemple : travail2partie2-Audet-Daniel.zip.pgp)

Le fichier d'archive (ZIP) doit contenir tous les fichiers requis **pour visualiser l’anneau demandé à l’étape g.**

Veuillez inclure le dossier « Common » dans votre fichier d’archive de telle sorte que l’extraction du contenu de votre fichier ZIP permettent de visualiser tous les fichiers HTML en double-cliquant sur ceux-ci (vos fichiers HTML doivent donc contenir un chemin correct pour qu’un navigateur puisse retrouver les fichiers Javascript présents dans le dossier « Common »).

Cryptez le fichier d'archive ZIP et transmettez le fichier résultant (PGP) sur la plateforme Moodle du cours.

[Procédure de cryptage à utiliser pour remettre les travaux](#)

[Accès à Moodle \(UQAC\)](#)