

IFT3355 : Infographie

Travail Pratique 1 : Un affichage 3D défectueux.

Introduction à *Three.js*, *WebGL* et aux Vertex Shaders

Disponible : Lundi 13 janvier 2020 11 : 00
Remise : Lundi 27 janvier 2020 23 : 59

1 Introduction

Les principaux objectifs de ce TP sont la mise au point d'un environnement de développement graphique, incluant la vérification de compatibilité de votre navigateur web, l'autorisation d'utilisation de fichiers locaux ainsi que la découverte des vertex et fragment shaders. Dans le cadre de cette découverte, vous utiliserez un template mis à votre disposition, incluant des shaders (fichiers `.glsl`).

Votre travail principal consistera à acquérir une compréhension haut niveau de la façon dont fonctionne le programme, à modifier des shaders, et faire communiquer de manière rudimentaire du code *JavaScript* et des shaders. Une partie des détails de ce qui se passe dans le restant du code deviendra plus clair un peu plus tard dans la suite du cours. Plusieurs de ces concepts sont expliqués dans les ressources web listées sur le site web du cours. Vous êtes bien sûr invité à consulter ces ressources dès maintenant.

Pour programmer un shader, vous allez utiliser un langage de programmation appelé *GLSL* (*OpenGL ES Shading Language v1.0*). Notez qu'il y a plusieurs versions de *GLSL*, contenant des fonctionnalités plus avancées, disponibles en *OpenGL*. Lors de vos recherches sur le web, faites donc attention à ce que le code que vous puissiez trouver corresponde à la bonne version de *GLSL*.

Le template fourni une scène simple avec une affichage jaunâtre, fonctionnant correctement et affichant un armadillo ainsi qu'un contrôleur rouge défectueux. Votre tâche consistera à réparer le contrôleur ainsi qu'à perturber de différentes façon le fonctionnement de l'affichage 3D. Vous pouvez faire bouger la caméra autour de la scène à l'aide de la souris.

2 Template

- Le fichier `A1.html` est le launcher du TP. Ouvrez ce fichier à l'aide de votre navigateur favori afin d'exécuter le TP.
- Le fichier `A1.js` contient le code *JavaScript* utilisé pour construire la scène et rendre l'environnement. Vous aurez à faire des changements mineurs afin de répondre aux questions demandées.
- Le répertoire `glsl` contient les vertex et fragment shaders liés au personnage et au contrôleur. C'est dans ces fichiers que vous aurez à faire la majorité du travail.
- Le répertoire `js` contient les bibliothèques *JavaScript* requises. Vous n'avez pas à modifier ces fichiers.
- Le répertoire `obj` contient les géométries chargés dans la scène.
- Le répertoire `images` contient les textures utilisées.

3 Exécution

Comme mentionné ci-dessus, le TP peut être lancé en exécutant le fichier `A1.html` dans n'importe quel navigateur moderne. Toutefois, la plupart des navigateurs empêcheront les pages d'accéder aux fichiers locaux sur votre machine. Si vous exécutez `A1.html`, vous obtiendrez probablement un écran noir et un message d'erreur dans la console, similaire au message suivant :

```
XMLHttpRequest cannot load armadillo.vs.glsl.  
Cross origin requests are only supported for protocol  
schemes : http, data, https.
```

Pour contourner ce problème et dans le cadre des travaux pratiques utilisant *JavaScript*, nous vous conseillons d'utiliser *WebStorm* (<https://www.jetbrains.com/fr-fr/webstorm/>), un IDE pour les langages Web. Bien que payant, vous pouvez bénéficier d'une licence gratuite en tant qu'étudiant.

Si vous ne voulez pas travailler avec un IDE, la page web suivante vous donnera des renseignements sur la façon de pouvoir contourner le problème mentionné plus tôt :

<https://threejs.org/docs/#manual/en/introduction/How-to-run-things-locally>

Enfin, notez qu'après modification du code, vous pourriez être amené à devoir effectuer un "*hard refresh*" de votre navigateur web (CTRL + F5) afin de rafraîchir le cache.

4 Travail à réaliser

Avant toute chose, assurez-vous donc vous puissiez exécuter le template fourni dans le navigateur web de votre choix. Etudiez alors le template de manière à avoir une compréhension globale de la manière dont le programme fonctionne.

1. Partie 1 : Contrôleur (70 points)

(a) Déplacer le contrôleur (15 pts)

La variable `remotePosition` (représentant la position du centre du contrôleur dans le repère monde) déclarée dans le fichier `A1.js` est modifiée à l'aide des entrées clavier et passée au vertex shader du contrôleur (`remove.vs.glsl`) à l'aide d'une variable de type `uniform`. Modifiez le shader du contrôleur de manière à changer sa position en fonction des entrées clavier.

Important : N'utilisez pas les fonctions de *Three.js* ; vous serez noté sur le code présent dans le shader.

(b) Ramener le contrôleur à la vie (20 pts)

Modifiez le fragment shader du contrôleur (`remote.fs.glsl`) de manière à ce qu'il change sa couleur en fonction de la variable `tvChannel`. Notez que vous avez à passer cette variable de type `uniform` au shader depuis le code *JavaScript*, où elle est déjà déclarée et modifiée par les entrées clavier. Vous pourrez modifier le canal et voir la façon dont le contrôleur réagit.

(c) Interférences (35 pts)

Il s'avère qu'un puissant aimant se trouve à l'intérieur du contrôleur, causant des interférences dans l'affichage lorsque celui-ci se trouve à proximité d'un élément de la scène. Modifiez le shader du personnage de manière à affecter la couleur des pixels se trouvant à une distance donnée du contrôleur, renvoyant alors la couleur de votre choix.

Indice 1 : Vous serez probablement amené à faire quelque chose avec la variable `interpolatedNormal`, définie dans le vertex shader de *armadillo* et passée à son fragment shader.

Indice 2 : Dans le vertex shader, vous pourriez vouloir utiliser une des matrices de transformation prédéfinies :

```
uniform mat4 modelMatrix;  
uniform mat4 modelViewMatrix;  
uniform mat4 projectionMatrix;  
uniform mat4 viewMatrix;  
uniform mat3 normalMatrix;
```

2. Partie 2 : Canaux - Creative License (30 pts)

Il est maintenant l'heure de tester différents affichages pour chacun des canaux. Modifiez votre affichage afin de distordre armadillo de trois manières différentes. Amusez-vous avec ces distorsions! Vous pourriez, par exemple, vouloir :

- Perturber le maillage de armadillo à l'aide de vagues.
- Ajouter du bruit.
- Faire exploser le modèle dans la direction des normales de ses faces.
- Colorer armadillo à l'aide de couleurs variant dans le temps et/ou l'espace.

ou toute autre chose d'une complexité similaire ou supérieure. Des points bonus seront attribués pour récompenser des pistes d'exploration particulièrement remarquables.

5 Remise

Créez un repertoire VOTRENOM_TP1 et placez-y les fichiers sources du projet en respectant la hierarchie du template. Placez-y également un fichier README.txt contenant votre nom ainsi que toute information que vous aimeriez fournir au correcteur. Compressez ce repertoire avant de le rendre sur StudiUM.