

# IFT3355 : Infographie

## Travail Pratique 1 : Un affichage 3D défectueux

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

Disponible : Lundi 13 septembre 2021  
Remise : Lundi 27 septembre 2021 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*). 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*. La version que nous utilisons pour ce TP est décrite par :

[https://www.khronos.org/registry/OpenGL/specs/es/3.0/GLSL\\_ES\\_Specification\\_3.00.pdf](https://www.khronos.org/registry/OpenGL/specs/es/3.0/GLSL_ES_Specification_3.00.pdf).

Le template fourni une scène simple avec une affichage jaunâtre, fonctionnant correctement et affichant un armadillo ainsi que trois contrôleurs rouges défectueux. Votre tâche consistera à réparer les contrôleurs 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 `TP1.html` est le launcher du TP. Ouvrez ce fichier à l'aide de votre navigateur favori afin d'exécuter le TP.
- Le fichier `TP1.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 aux contrôleurs. C'est dans ces fichiers que vous aurez à faire la majorité du travail.
- Le répertoire `js` contient les librairies *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 `TP1.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 `TP1.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. Étudiez alors le template de manière à avoir une compréhension globale de la manière dont le programme fonctionne.

#### 1. Partie 1 : Contrôleurs (70 points)

##### (a) Déplacer le contrôleur (10 pts)

La variable `remotePosition` (représentant la position du centre du contrôleur dans le repère monde) déclarée dans le fichier `TP1.js` est modifiée à l'aide des entrées clavier et passée au vertex shader du contrôleur (`remote.vs.glsl`) à l'aide d'une variable de type `uniform`. Modifiez le shader des contrôleurs de manière à changer leur position en fonction des entrées clavier.

Les boutons Q et A sur le clavier changent la hauteur du premier contrôleur (de gauche vers la droite), W et D changent le deuxième, E et D la troisième.

*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 des contrôleurs (`remote.fs.glsl`) de manière à ce qu'il change sa couleur en fonction sa hauteur. 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.

##### (c) Perturbations avec du son (30 pts)

Ajoutez un fichier `test.mp3` de votre choix dans le répertoire principal et appuyez sur le bouton P sur le clavier pour commencer la musique. (Attendez l'apparition du personnage sur l'écran avant de jouer la musique.) Alternativement, vous pouvez tester votre programme en utilisant le microphone comme entrée en appuyant le bouton M sur le clavier. Pour la remise, le fichier `test.mp3` doit être présent.

Perturbes la géométrie et les pixels du personnage avec l'information donnée par le FFT (Fast Fourier Transform) sur une musique de votre choix en temps réel avec le vertex et pixel shader. Le FFT d'un signal audio donne l'amplitude de toutes les fréquences dans le son (vous pouvez choisir les fréquences à utiliser). Un exemple d'accès au FFT est donné dans le shader `armadillo.vs.glsl`, où la valeur `x` du `vec2` correspond à la fréquence choisie. La plus basse fréquence correspond à la valeur 0.0 et la plus haute fréquence correspond à 1.0.

Préparez trois perturbations différentes, les trois contrôleurs vont contrôler leur intensité par la suite. La première perturbation doit réagir aux fréquences basses et déformer la géométrie dans la direction de la normale. La deuxième et la troisième doit réagir aux fréquences moyennes et hautes et changer les couleurs du personnage. C'est à vous de décider comment les couleurs doivent changer.

*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;
```

(d) **Intensité des perturbations** (10 pts)

La hauteur des contrôleurs doit affecter l'intensité des perturbations sur le personnage. Chaque contrôleur affecte le personnage d'une façon différente et prend des différentes fréquences du FFT, mais l'effet global affiché sur le personnage doit être une combinaison des trois contrôleurs. Modifiez les shader du personnage de manière à affecter la géométrie et la couleur des pixels avec des effets de votre choix selon la hauteur des contrôleurs.

2. **Partie 2 : Creative License** (30 pts)

Il est maintenant l'heure de tester différents affichages pour chacun des contrôleurs. Modifiez votre affichage afin de distordre le personnage de trois manières différentes en utilisant le FFT, le temps et du bruit. Amusez-vous avec ces distorsions ! Des points bonus seront attribués pour récompenser des pistes d'exploration particulièrement remarquables. Notez que chaque perturbation doit utiliser le son, le temps et du bruit pour pouvoir obtenir la totalité des points.

Vous pouvez utiliser et vous inspirer avec ces fonctions de bruit :

<https://thebookofshaders.com/11/>

<https://gist.github.com/patriciogonzalezvivo/670c22f3966e662d2f83>

Si vous copiez du code de l'internet, veuillez le noter avec des commentaires. (0 tolérance au plagiat !)

## 5 Remise

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