

R6.A.05 – Développement avancé TP 1

Animation et rendu dans un navigateur web avec WebGL

Le but de ce TP est de voir comment afficher de la 3D interactive dans un navigateur web avec WebGL, en utilisant la librairie **ThreeJS** disponible sur <https://threejs.org/>

Installation

Vous allez installer la bibliothèque **ThreeJS** sur votre machine.

Téléchargez la bibliothèque sur sa page Github et décompressez-là sur votre disque :

<https://github.com/mrdoob/three.js/archive/master.zip>

Premier programme

1. Ecrivez un fichier `index.html` dans lequel vous mettrez le code des diapos 13 à 15 du cours sur ThreeJS. Dans ce code, assurez-vous que le chemin d'accès au fichier `three.module.js` est correct par rapport à l'emplacement de votre fichier `index.html`. De même pour le chemin d'accès au répertoire `jsm` qui contient les librairies des « *addons* » de ThreeJS. Par exemple :

```
<script type="importmap">
{
  "imports": {
    "three": "./three.js-master/build/three.module.js",
    "three/addons/": "./three.js-master/examples/jsm/"
  }
}
</script>
```

Faites un double-clic sur `index.html` pour le charger dans un navigateur. Vous devriez normalement voir dans une page web un cube vert qui tourne. Si ce n'est pas le cas, appuyez sur la touche F12 (sur Firefox, Chrome) pour afficher la console JavaScript et voir la sortie d'erreurs.

2. Ajoutez une source de lumière ambiante au moyen de la classe **AmbientLight** (voir cours diapo 31). Vous ne devriez voir pour le moment aucune différence, car dans l'exemple du cours le matériau utilisé pour le cube est de type **MeshBasicMaterial**, dont la couleur n'est pas affectée par les sources de lumière.
3. Ajoutez une source de lumière directionnelle au moyen de la classe **DirectionalLight** (voir cours diapo 32).
4. Ajoutez au cube un matériau correspondant au modèle d'illumination de Phong (voir cours diapo 34).
5. Ajoutez à la scène un système d'axes *xyz* au moyen de la classe **AxesHelper** (voir cours diapo 43).
6. Ajoutez à la scène une grille dans le plan horizontal *xoz* au moyen de la classe **GridHelper** (voir cours diapo 45).
7. Ajoutez un mode de contrôle de la caméra permettant de la faire tourner autour de l'origine au moyen de la classe **OrbitControls** (voir cours diapo 28). Le clic gauche permet de faire tourner la caméra, la molette de zoomer et le clic droit de translater.
8. Faites en sorte qu'en appuyant sur la touche 'a' du clavier on puisse faire apparaître/disparaître le système d'axe *xyz*, et qu'en appuyant sur 'g' on puisse faire apparaître/disparaître la grille (voir cours diapos 7 et 46).
9. Ajoutez à la scène un plan horizontal à $y=0$ au moyen de la classe **PlaneGeometry** (voir cours diapo 26). Par défaut, ce plan est vertical. Pour qu'il soit horizontal, appliquez lui une rotation de $-\text{Math.PI}/2$ sur l'axe *x* (voir cours diapo 18).
10. Ajoutez au plan un matériau correspondant au modèle d'illumination de Phong (voir cours diapo 34).
11. Plaquez une texture sur le plan avec l'image `sol.jpg` à télécharger sur Ametice (voir cours diapo 36).
Si la texture n'apparaît pas, regardez les messages d'erreur dans la console du navigateur. Il se peut que le fichier d'image ne puisse pas être chargé pour des raisons de « CORS » (*Cross-Origin Resource Sharing*), un mécanisme des navigateurs web qui empêche l'accès à des fichiers locaux pour des raisons de sécurité (voir diapos 41-42).
Pour installer un serveur local si vous avez Python, ouvrez un terminal, placez vous avec la commande **cd** dans le répertoire qui contient ThreeJS et votre fichier `index.html`, et tapez :
python -m http.server

Vous pouvez alors voir votre site servi par ce serveur dans un navigateur web, sans problème de CORS, à l'adresse `http://localhost:8000/`
12. Ajoutez à la scène d'autres cubes, des sphères, des cylindres, des cônes, des tores (voir cours diapos 24 à 26)
13. Ajoutez du brouillard à la scène (voir cours diapo 50). Testez différentes valeurs de couleur de brouillard, de distances de début et de fin.