

TP Three JS

Exercice 1

Objectif: Premier pas avec Three.js

Créer un scène avec deux objets de type sphere

- Créez un page html et en y référençant la bibliothèque three.js et votre fichier javascript dans lequel vous implémenterez l'exemple.

```
<script type="text/javascript" src="../../js/three.js"></script>
<script type="text/javascript" src="init.js"></script>
```

- Sur l'événement onload de la balise body appelez la fonction d'initialisation de votre application
- Dans la fonction d'initialisation
 - Créer un moteur de rendu WebGL
 - Définissez la taille de la fenêtre de rendu window.innerWidth X window.innerHeight
 - Ajoutez le moteur de rendu au DOM via

```
document.body.appendChild( renderer.domElement );
```
 - Créez une scène
 - Créez une caméra positionnée en (0,0,20)
 - Créez une lampe de type 'SpotLight', positionnée en (0, 30, 60) et d'intensité 1 puis l'ajouter à la scène
 - Créer deux géométries de type sphere de rayon .2, l'une positionnée en (0,0,0) l'autre en (1,0,0)
 - Créer un matériau de type 'MeshPhongMaterial' avec comme paramètre une couleur rouge
 - Créer deux maillages à l'aide des deux géométries et du matériaux
 - Les ajouter à la scène
 - Faites le rendu de la scène avec la caméra définie préalablement.

Animer la scène

- Ajouter les deux spheres dans un groupe et ajoutez le groupe à la scène
- Écrire la fonction de rafraîchissement (comme vu en cours)

```
var render = function () {
    requestAnimationFrame( render );
    animate();
    renderer.render(scene, camera);
};
requestAnimationFrame( render );
```

- Écrire la fonction animate faisant tourner autour de l'axe Y le groupe des deux sphères d'un pas de rotation deltaTheta prédéfini dans votre code. Vous utiliserez la méthode rotateY() des 'objet3D'.

Exercice 2

Objectif: Visualiser une courbe 3D.

À partir du code de l'exercice 1 vous allez modifier uniquement la partie consistant à ajouter les sphères au groupe.

- Écrire une fonction paramétrique de $[0, 1]$ dans R^3 retournant les valeurs (x, y, z) pour une valeur de paramètre t , i.e. $f(t) = (x(t), y(t), z(t))$.
- Faites varier le paramètre t entre 0 et 1 et pour chaque valeur du paramètre ajoutez une sphere positionnée en $f(t)$ au groupe. Testez comme fonction :
 $x(t) = \cos(5 * t * 2 * PI)$; $y(t) = \cos(7 * t * 2 * PI)$; $z(t) = \cos(13 * t * 2 * PI)$;
- Au cours de l'animation faites dilater les sphères les une après les autres pour donner l'impression qu'une plus grosse sphère se déplace le long de la courbe.

Exercice 3

En utilisant des groupes imbriqués réalisez la marche de la chenille fait avec blender.

Exercice 4

Objectif : À l'aide des touches clavier, pilotez un avatar se déplaçant sur une grille de jeu et évitez les boulets.

Construction de la grille

Disposer suivant une grille de 10×10 en x et y des cubes qui constitueront la surface de jeu.

Les cubes seront :

- de largeur *cubeSize* paramétrable
- de couleur choisie aléatoirement pour les distinguer et percevoir le déplacement de l'avatar.

L'avatar doit être défini par un état décrivant :

- sa position
- sa direction
- une vitesse de déplacement dans la direction courante
- une vitesse de rotation

Les mouvements possibles de l'avatar sont :

- avancer tout droit
- faire une rotation sur la droite (flèche à droite)
- faire une rotation sur la gauche (flèche à gauche)

L'avatar ne sera pas visible mais la caméra prendra la position de l'avatar.

Implémentez le pilotage de l'avatar sur la grille. Si l'avatar sort de la grille envoyez un signal d'alerte.

Épreuve du parcours

Au lieu de se déplacer sur la grille complète, l'avatar doit se déplacer que sur certain bloc sans quoi il tombera dans le vide.

À partir de la grille précédente générer un ensemble de blocs contigus allant d'un point A de départ à un point B d'arrivée diamétralement opposés. Le joueur doit se rendre le plus vite possible du point A au point B sans tomber dans le vide.

Les boulets

Sur le parcours se promènent des boulets qu'il faut absolument éviter faute de se faire pousser dans le vide. Les boulets démarrent du point B et se promène aléatoirement sur le parcours (à chaque bifurcation il choisissent aléatoirement une nouvelle orientation qui peut être l'orientation opposée à leur déplacement).