

Mondes Virtuels - Labo 2

Automne 2025

Exercice 1

Modifiez le monde virtuel que vous avez déjà créé de façon à obtenir un musée de même taille que précédemment mais contenant un hall (20 m sur 10 m) et 3 salles (coté nord). Chaque salle communique avec le hall. Placez une sphère de 1m de rayon qui repose sur le sol.

Chacun de ces objets est fait dans un matériau de type `THREE.MeshBasicMaterial` pour lesquels vous définirez l'attribut de couleur de la façon suivante (exemple pour une sphère) :

```
const geo = new THREE.SphereGeometry(1);
const mat = new THREE.MeshBasicMaterial();
mat.color = 0xCEF6EF ;
const mesh = new THREE.Mesh(geo,mat);
scene.add(mesh);
```

Les couleurs seront choisies parmi celles d'une palette ne contenant qu'un nombre réduit de couleurs. Ceci afin de donner une atmosphère particulière à votre monde. Pour créer votre palette vous pouvez utiliser des sites Web tels que <https://mycolor.space> et <https://colorlisa.com>

Pour ce travail vous fixerez votre palette à au plus 6 couleurs que vous donnerez dans votre rapport.

Exercice 2

L'aspect de la scène ainsi obtenu n'est pas réaliste. Ceci pour une raison essentielle : l'absence de lumière et des interactions entre la lumière et les matériaux associés aux objets de la scène. Pour remédier à cela on choisit d'utiliser des matériaux sensibles à la lumière. Pour cela on change de type de matériau.

Plutôt que d'utiliser les anciens matériaux (tels que `THREE.MeshLambertMaterial` ou `MeshPhongMaterial`) on utilisera le matériau `MeshStandardMaterial` (ultérieurement `MeshPhysicalMaterial`).

Ces nouveaux matériaux permettent :

- d'avoir un éclairage physiquement correct : on s'intéresse ici à reproduire correctement des effets liés à la propagation de la lumière, par exemple les atténuations de l'intensité lumineuse dépendant de la distance parcourue par la lumière
- d'avoir un rendu basé physique : le calcul de la façon dont la lumière interagit avec la matière est décrit par des lois dérivées de la physique

Ce mode de calcul est activé en modifiant le **renderer** de la façon suivante :

```
function createRenderer(){
  const renderer = new WebGLRenderer();
  // Ici code identique à celui existant déjà
  renderer.physicallyCorrectLights = true;
  return renderer;
}
```

Question : mettez en oeuvre ce changement de matériau.

Contrairement aux matériaux non sensibles à la lumière, des sources lumineuses sont ici nécessaires pour que les objets apparaissent et ne soient pas noirs.

Dans le monde réel les objets sont éclairés de deux façons différentes :

- par **éclairage direct** : la lumière incidente à une surface provient directement des sources lumineuses
- par **éclairage indirect** : la lumière incidente à une surface provient de sources secondaires : la surface est éclairé par d'autres surfaces éclairées elles-mêmes directement ou indirectement (par exemple des réflexions multiples).

En THREE.js l'éclairage direct est mis en oeuvre par les sources lumineuses suivantes :

- **THREE.DirectionalLight** : imiter le soleil
- **THREE.PointLight** : imiter une ampoule
- **THREE.SpotLight** : imiter un spot
- **THREE.RectLight** : imiter un néon ou une fenêtre laissant passer beaucoup de lumière.

Question Cherchez le paramétrage de ces sources lumineuses dans la documentation de THREE.js (<https://threejs.org/docs/>).

Question dans un premier temps vous n'utiliserez qu'une source directionnelle. Animez cette source en faisant dépendre ses paramètres du temps.

Exercice 3

On cherche dans cet exercice à étudier les effets de la lumière sur l'aspect des objets. Pour cela on met en oeuvre un "rideau" de sphères, c'est à dire un ensemble de $n*n$ sphères placées dans un même plan vertical de façon à ce que les centres des sphères soient espacés de façon régulière.

Question proposez une fonction qui permette de créer de tels rideaux. Cette fonction sera paramétrée en particulier par nle nombre de lignes et de colonnes et **r** le rayon de chaque sphère du rideau.?

Les deux attributs de base des matériaux de type `MeshStandardMaterial` sont, outre l'attribut `color` :

- `roughness` : 0 (lisse) à 1 (très rugueux)
- `metalness` : 0 (non métallique) à 1 (métal)

Question : affectez des matériaux différents à chaque sphère d'un rideau. Les paramètres de la sphère en bas à gauche du rideau valent 0 tandis que ceux du haut à droite valent 1. La couleur de tous ces matériaux est la même.

Question : décrivez les effets de la lumière sur les sphères en fonction des paramètres `roughness` et `metalness`.

Question : d'ailleurs pourquoi utiliser des sphères plutôt que des boites ou des rectangles?

Question : Recommencez avec le matériau `THREE.MeshPhysicalMaterial` mais cette fois-ci en faisant varier les paramètres `clearCoat` et `clearCoatRoughness` (ajout d'une couche de vernis à la surface de l'objet).

Exemple de matériau : `THREE.MeshPhysicalMaterial`

color	0x448AFF	couleur
roughness	0.9	aspect rugueux
metalness	0.1	aspect métallique
clearCoat	0.1	intensité de la couche de vernis
clearCoatRoughness	0.5	rugosité de la couche de vernis
reflectivity	0.1	degré de réflectivité

Exercice 4

Dans l'exercice précédent vous avez travaillé avec des éclairages directs. Reste à traiter de l'éclairage indirect.

Question : trouvez dans votre scène une configuration "source directe / objet" qui conduit à un éclairage peu crédible. Argumentez sur la cause de ce manque de crédibilité.

On propose de remplacer les contributions indirectes - qui seraient trop coûteuses à calculer de façon explicite par une source particulière dite ambiante.- Dans le monde réel les objets sont éclairés non seulement par les sources lumineuses (80% *des contributions* le reste est de l'éclairage indirect).

En THREE.js une source ambiante est une source lumineuse qui représente la lumière provenant de tout l'environnement d'un objet. Pour cet objet la lumière provient donc de toutes les directions.

Une source lumineuse ambiante sera spécifiée par la couleur et l'intensité de la lumière qu'elle envoie :

```
const light = new THREE.AmbientLight(0xffeecc,0.4);
scene.add(light) ;
```

Question : ajoutez une source ambiante

Exercice 5

Pour l'instant les sources lumineuses sont invisibles : ce sont uniquement des métaphores abstraites, non incarnées qui permettent de définir les propriétés de la lumière. Dans cet exercice on veut remédier à ce problème et représenter visuellement certaines sources lumineuses.

Question Proposez une représentation d'une ampoule en tenant compte des points suivants :

- La source lumineuse est une source ponctuelle
- L'ampoule est une sphère d'une couleur donnée
- Les positions de la source ponctuelle et celle du centre de l'ampoule doivent être confondues

Mettez en oeuvre cette représentation au moyen d'une fonction qui renvoie une ampoule qui éclaire.

Exercice 6

Question Dans chaque salle utilisez des sources de type `THREE.SpotLight` pour éclairer dans chacune des salles deux tableaux.

Question Panne de courant dans le musée. Eclairez vous pour vous déplacer en utilisant une lampe torche mise en oeuvre par une source lumineuse de type `THREE.SpotLight`.

Exercice 7

Il s'agit ici d'utiliser de textures pour améliorer la crédibilité du monde.

Question Utilisez pour cela les fichiers du répertoire `assets/textures/`.

Question Que se passe-t-il quand le rapport hauteur / largeur de la surface texturée est très différent du rapport nombre de lignes / nombre de colonnes du fichier texture ? Comment remédier à cela ?

Exercice 8

Choisissez un thème à votre musée. Placez dans les salles, de façon pertinente environ 10 tableaux. Adaptez les matériaux et les éclairages à votre thème. Tenez compte des contraintes de palette.