TP Three.js 2

Licence Informatique 3ème année

Année 2023-2024
durée : 3h

L'objectif de ce second TP est d'expérimenter les interactions entre un utilisateur et les objets3D qui se trouvent dans votre scène, via le gestionnnaire d'événements de votre navigateur.
Dans un premier temps, récupérez l'archive jointe à cet énoncé et extrayez les dossiers et fichiers qui s'y trouvent dans un sous-dossier nommé TP2 qui se trouvera dans le dossier où vous réalisez vos TP d'informatique graphique. Après extraction vous disposez :

- d'un dossier `js` qui contient le fichier `mesObjets.js`. Ce dernier contient la fonction `cubeColor(c)` qui permet de créer un cube de côté `c`, centré à l'origine, dont les 6 faces sont de couleurs différentes (correction du dernier exercice du TP précédent) ;
- un fichier `tp2_threejs.html` qui permet l'affichage du cube défini dans le fichier `mesObjets.js`.

Ce TP devra être rendu à la fin de la séance, sous la forme d'une archive nommée `TPIG2-XXX`, où `XXX` sera remplacé par votre nom de famille. Elle devra contenir le dossier TP2 et les dossiers et fichiers qui y seront contenus et sera à envoyer via un site tel que `wetransfer`[1] vers l'adresse suivante : `christophe.renaud@univ-littoral.fr`.

Pour tous les TPs Three.js, vous pourrez consulter la documentation officielle en ligne, à l'adresse :
`https://threejs.org/docs/index.html`

Exercice 1

Ajoutez à votre fichier `tp2_threejs.html` un gestionnaire d'événements permettant de traiter les événements clavier (de type `keydown`). Cette fonction devra permettre :

- de déplacer le cube d'une unité vers l'arrière en cas d'appui sur la touche `-` ;
- de déplacer le cube d'une unité vers l'avant en cas d'appui sur la touche `+` ;
- de tourner le cube de 0.1 radian vers la droite autour de l'axe Oy lors de l'appui sur la touche `ArrowRight` ;
- de tourner le cube de 0.1 radian vers la gauche autour de l'axe Oy lors de l'appui sur la touche `ArrowLeft` ;
- de tourner le cube de 0.1 radian vers le bas autour de l'axe Ox lors de l'appui sur la touche `ArrowDown` ;
- de tourner le cube de 0.1 radian vers le haut autour de l'axe Ox lors de l'appui sur la touche `ArrowUp`.

Vous vous référerez aux schémas des rotations présentées dans l'énoncé du TP1 pour trouver le signe de l'angle à appliquer. On précise que l'attribut de l'événement à utiliser pour tester la touche appuyée est `event.key`.

Exercice 2

On souhaite créer un objet 3D permettant de visualiser le repère dans lequel se trouve le cube. Ce repère sera un objet composé de lignes colorées (rouge pour l'axe Ox, vert pour l'axe Oy et bleu pour l'axe Oz) représentant ses 3 axes (voir figure 2a).

1. Ajoutez, dans le fichier `mesObjets.js`, une fonction nommée `repere(size)` qui permet de créer cet objet 3D. Le paramètre `size` représentera la taille de chaque axe, sachant que le repère devra être centrée en $(0, 0, 0)$. Le code à écrire sera simulaire à celui du cube, avec les différences suivantes :

1. Le dossier contenant des scripts `javascript`, il ne pourra pas être envoyé directment par mail, la passerelle antivirus de l'ULCO empêchant l'envoi de ce type de fichiers.

— la géométrie de l'objet sera constituée par un objet `LineSegments` ; dans ce cas, les sommets qui apparaissent dans le `GeometryBuffer` sont interprétés comme des couples successifs de coordonnées représentant les extrémités de chaque segment qui sera dessiné (cf figure 1) ;

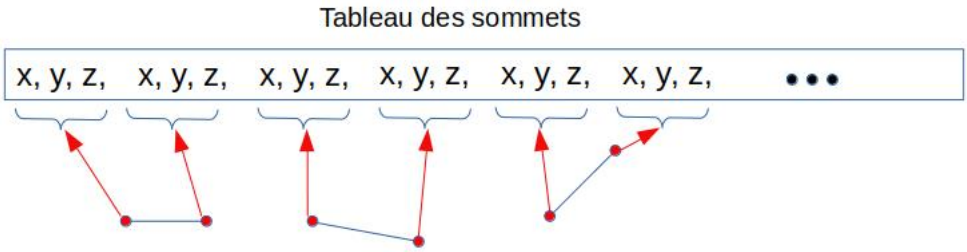
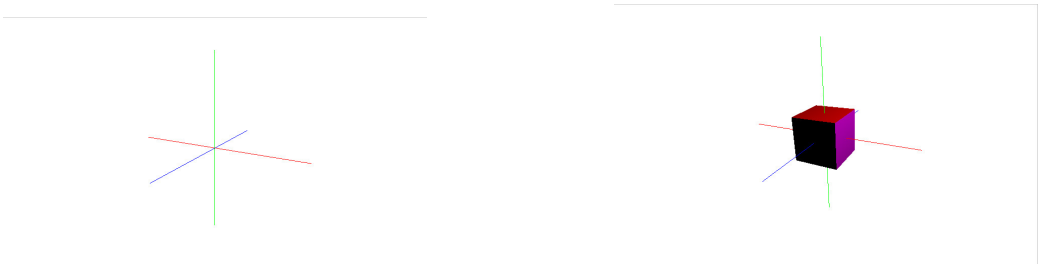


FIGURE 1 – Organisation du `GeometryBuffer` d'un objet de type `LineSegments`.

— le matériau de l'objet sera représenté par un objet `LineBasicMaterial` pour lequel il sera nécessaire que l'attribut `vertexColors` soit mis à `true` (voir documentation) ;
— les lignes ne posséderont pas de normales.

2. Modifiez le fichier `tp2_threejs.html` de telle sorte qu'un repère de taille 4 soit ajouté à la scène et qu'il tourne en même temps que le cube, avec les mêmes mouvements, lors des interactions avec l'utilisateur (voir figure 2b).



(a) Vue du repère coloré.

(b) Vue du repère coloré et du cube qui se déplacent conjointement.

FIGURE 2 – Vue du repère et du cube.

Exercice 3

Lorsque des objets différents sont utilisés pour créer un objet plus complexes (par exemple le cube associé au repère de l'exercice précédent), il est possible de les grouper dans un groupe d'objets, afin de faciliter leur manipulation. Après avoir étudié la documentation de la classe `Group` :

— modifiez le code du fichier `tp2_threejs.html` pour grouper le repère et le cube et appliquer les déplacements prévus directement sur le groupe. Vérifiez le fonctionnement ;
— ajoutez une fonction nommée `repereFleche(size)` à votre fichier `mesObjets.js` qui devra créer un repère similaire à celui de la première question, mais pour lequel des flèches seront présentes sur chaque axe dans le sens positif de cet axe (voir figure 3a). Pour ce faire, vous utiliserez la classe `ArrowHelper` qui permet de créer une flèche et vous regrouperez les 3 flèches nécessaires au sein d'un groupe qui sera retourné par la fonction. Vous modifierez ensuite votre fichier `tp2_threejs.html` de telle sorte qu'il utilise ce nouveau repère (voir figure 3b).

Exercice 4

Lors du démarrage de votre script, la caméra et le renderer sont initialisés avec les dimensions courantes de la fenêtre de votre navigateur. Si vous changez la taille de celle-ci, la modification n'est pas prise en compte et vous vous retrouvez avec un affichage partiel (voir figure 4a). Pour pallier ce problème, il est nécessaire de rajouter une fonction gérant l'événement de redimensionnement de la fenêtre et de remettre à jour les paramètres du **renderer** et de la **caméra**.

1. Ajoutez un gestionnaire d'événements permettant d'être activé lors de la survenue de l'événement `resize` ;

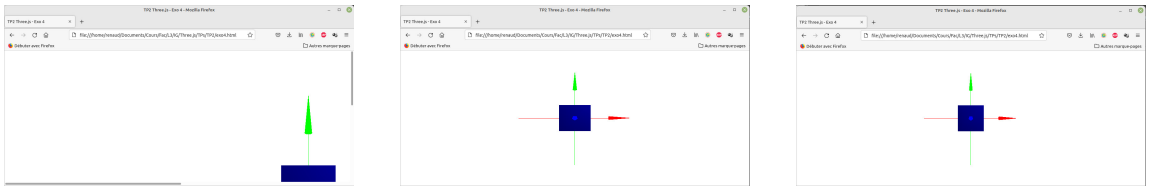
(a) Vue du repère avec les flèches. (b) Vue du groupe composé du repère et du cube.

FIGURE 3 – Vue du nouveau repère et de son association au cube.

2. Complétez la fonction appelée lors de cet événement pour quelle mette à jour les dimensions de la zone d'affichage du renderer (méthode `setSize`). Testez. Vous notez que lors du redimensionnement, la scène est bien dessinée au centre de la fenêtre, quelle que soit sa taille. Par contre, les objets sont déformés par le redimensionnement (voir figure 4b). Ceci vient du fait que le rapport d'aspect initial avec lequel la caméra perspective a été initialisée n'a pas été modifié ...
3. Complétez votre fonction avec les deux lignes suivantes (en supposant que votre caméra porte le nom `camera`) :

```
camera.aspect = window.innerWidth / window.innerHeight;
camera.updateProjectionMatrix();
```

La première ligne permet de recalculer le rapport d'aspect, tandis que la deuxième permet de mettre à jour la matrice de projection perspective, afin que ces projections soient conformes au rapport d'aspect. Testez (voir figure 4c).



(a) Affichage après redimensionnement sans précautions. (b) Affichage après redimensionnement du renderer. (c) Affichage après réinitialisation de la caméra.

FIGURE 4 – Illustration des problèmes et solutions liées au redimensionnement de la fenêtre du navigateur.

Exercice 5

Lorsque vous utilisez la classe `Group` pour regrouper divers objets, le groupe constitué dispose d'un attribut nommé `children` qui est un tableau qui regroupe les différents objets assemblés. Pour votre objet actuel, on peut représenter son architecture informatique comme celle apparaissant sur la figure 5a : un groupe qui contient deux « fils », le second (le repère) étant lui même composé de 3 fils (les flèches représentant le repère). Cette architecture permet alors d'accéder aux différents composants d'un objet complexe, afin d'appliquer des traitements différents sur ceux-ci.

(a) Architecture du groupe composé d'un cube et d'un repère. (b) Architecture du groupe composé de deux cubes et leur repère associé.

FIGURE 5 – Architectures des groupes construits.

Remarque : un même objet ne peut appartenir qu'à un seul groupe ; en effet chaque objet stocke en interne un lien vers son parent, ce qui interdit d'avoir plusieurs parents. Lorsque vous voulez utiliser un objet similaire dans des groupes différents, il faut donc les recréer.

1. Ajoutez dans votre fichier `mesObjets.js` une fonction nommée `cubeRepere(c)` qui permet de construire et retourner le groupe qui correspond à votre objet actuel, composé d'un cube de côté `c` et d'un repère avec flèches (dont la dimension sera `c+3` pour que les flèches dépassent bien du cube). Vous remplacerez le code actuel utilisé dans votre script principal par un appel à cette fonction pour construire le cube et son repère ;
2. Modifiez le script principal de telle sorte qu'il construise deux cubes colorés (et leur repère respectif) placés respectivement en $x = 2$ et $x = -2$. Ces deux cubes devront être regroupés au sein d'un même objet qui sera celui manipulé par la fonction de gestion des événements clavier (voir figure 5b).
3. Lorsque vous utilisez les flèches gauche et droite du clavier pour faire tourner les cubes autour de leur axe Oy, vous notez que le comportement du nouveau groupe a un peu changé, puisque les deux cubes tournent désormais autour d'un centre commun qui est le repère global. Ce n'est pas le cas pour la rotation autour de l'axe Ox puisque les deux cubes partagent cet axe. De même, le zoom agit encore correctement, puisque on fait se déplacer l'intégralité du groupe. Pour pouvoir retrouver la rotation de chaque cube autour de son propre axec Oy, il est nécessaire d'accéder aux deux objets contenus dans l'objet global et de leur appliquer cette rotation à chacun. Modifiez votre fonction de gestion des événements clavier en conséquence et vérifier le nouveau fonctionnement.

Exercice 6

On souhaite à présent pouvoir appliquer les rotations sur les deux cubes, soit de manière disjointe (l'un des cubes tourne, l'autre pas), soit sur les deux, soit sur aucun. Pour cette question, on va supposer que l'appui sur la touche `1` permet d'activer/de désactiver les rotations sur le cube de gauche et que l'appui sur la touche `2` permet d'activer/de désactiver les rotations sur le cube de droite. Après avoir créé, dans votre script principal, un tableau de booléens à 2 entrées, nommé `rotation` et dont les entrées seront initialisées à `false`, modifiez votre gestionnaire d'événements clavier de telle sorte que :

1. un appui sur la touche `1` inverse la valeur booléenne contenue dans la première case du tableau `rotation` ;
2. un appui sur la touche `2` inverse la valeur booléenne contenue dans la seconde case du tableau `rotation` ;
3. les rotations ne s'appliquent sur chaque cube que si l'entrée correspondante dans le tableau `rotation` est vraie.