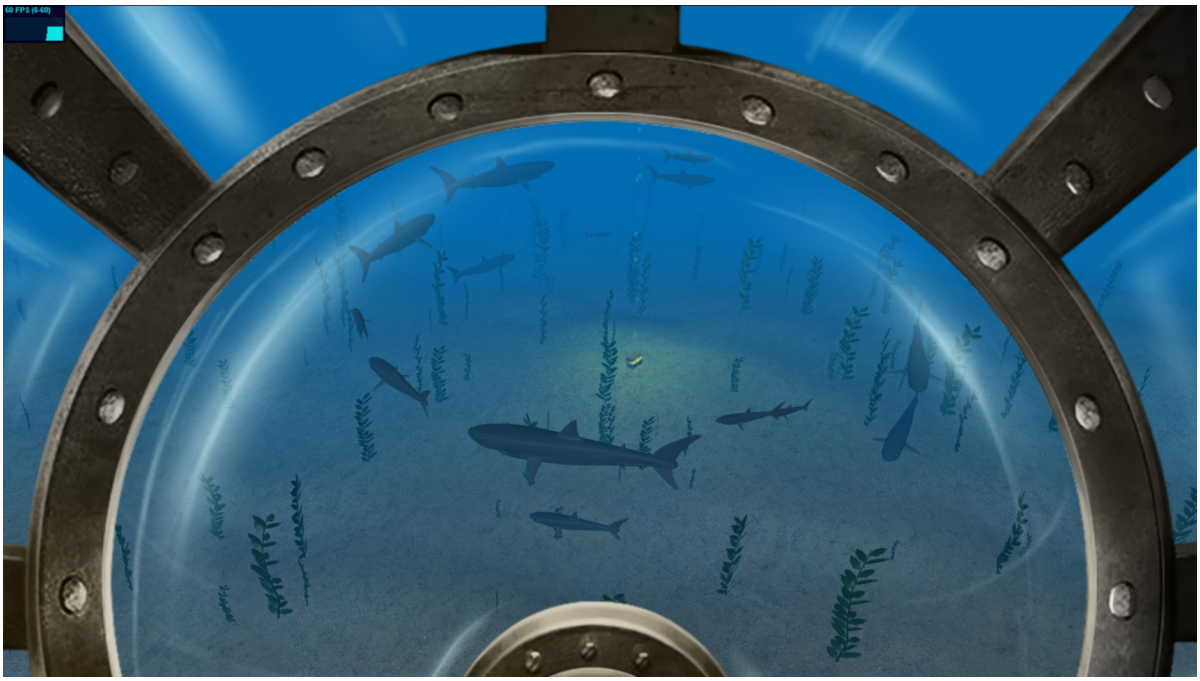
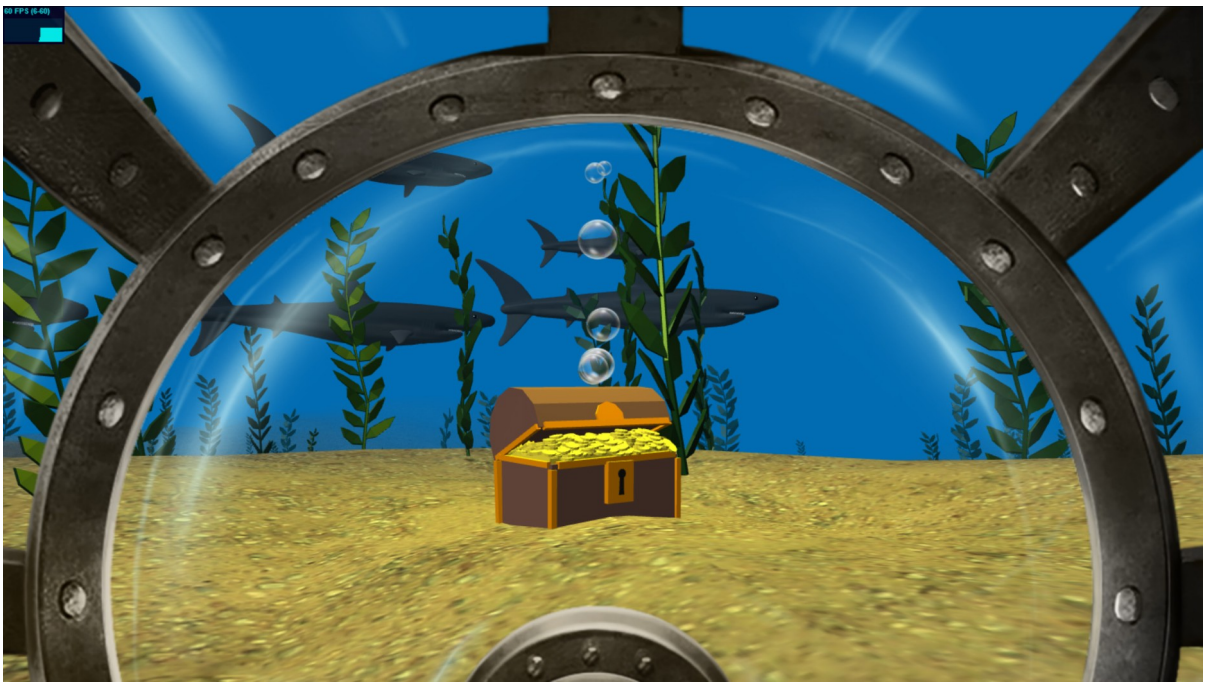
R6.A.05 – Développement avancé
TP noté

Vous allez au cours de ce TP noté afficher une scène sous-marine en 3D dans un navigateur web au moyen de ThreeJS. Téléchargez sur Ametice le fichier données_tp_noté.zip et décompressez-le dans le répertoire de votre projet.
A la fin de la séance, vous ferez une archive ZIP du répertoire de votre projet auquel vous donnerez votre nom (ex : dupont.zip) et vous le remettrez sur Ametice (rubrique « Rendez ici votre TP » sur la page du module R6.A.05)





1. Ajoutez à la scène une grille d'une taille de 200x200 dans le plan horizontal *xoz* au moyen de la classe THREE.GridHelper.

2. Positionnez une caméra initalement en (x=0, y=50, z=50).

3. Ajoutez un mode de contrôle de la caméra permettant de la faire tourner autour de l'origine au moyen de la classe THREE.OrbitControls. Modifiez le point visé par la caméra : controls.target.set(0, 10, 0);

4. Faites en sorte que la zone d'affichage 3D soit modifiée lorsque vous redimensionnez la fenêtre.

5. Le fichier terrain.json code au format JSON les altitudes du relief du fond sous-marin avec les valeurs suivantes :

```
{
        "dimx":101,
        "dimz":101,
        "altitudes" : [24,33,39,47,50,53,60,62,56,69,75,78,80,85,89,93,96,101,103,105,
                       108,113,117,122,126,130,132,135,139,142,145,147,149,151,154,
                       …
        ]
}
```

   Ceci correspond à un terrain codé sous la forme d'une grille régulière, dont dimx est le nombre de points selon l'axe x, dimz le nombre de points selon l'axe z, et le tableau altitudes contient les valeurs d'altitudes selon l'axe y de tous ces points (donc dimx * dimz valeurs), rangées dans un tableau à une dimension.
   Ecrivez le code permettant de charger ce fichier JSON et de créer un terrain 3D long de 200 x 200 unités que vous ajouterez à la scène.

   Aide : pour avoir un terrain ombré, ne lui appliquez pas de materiau de type THREE.MeshBasicMaterial qui ne réagit pas à la lumière, mais un matériau de type THREE.MeshLambertMaterial, et faites appel à la fonction suivante pour calculer les normales du terrain :

```
geometry.computeVertexNormals();
```

   (où geometry est la variable qui designe la géométrie du plan de type THREE.PlaneBufferGeometry que vous déformez)

6. Texturez ce terrain avec l'image sable.jpg que vous répeterez 10 fois selon les deux axes x et z du terrain.

7. Ajoutez une source de lumière ambiante de couleur 0x555555.

8. Ajoutez une source de lumière directionnelle de couleur 0x999999.

9. Ajoutez une source de lumière ponctuelle en (x=0, y=12, z=0) de couleur 0x999999, d'intensité 1 et qui éclaire jusqu'à une distance de 20.

10. Utilisez comme couleur de fond de la scène la couleur 0x016cb2.

11. Pour simuler l'atténuation de la lumière par l'eau, ajoutez un brouillard de couleur 0x016cb2, qui commence à la distance 2, et qui est maximal à la distance 120.

12. Chargez l'objet 3D tresor.obj et son matériau tresor.mtl et positionnez-le en (x=0, y=9, z=0).

13. Ecrivez une fonction creation_algues(nb_algues, terrain) dans laquelle vous chargerez l'objet 3D algue.obj et son matériau algue.mtl et que vous clonerez à nb_algues exemplaires à des positions et rotations (en y) aléatoires sur le terrain. Vous utiliserez du raycasting pour positionner correctement les algues à la surface du terrain (uniquement la position, pas la rotation, les algues devront pointer vers le haut). Vous appliquerez une mise à l'échelle aléatoire entre 1 et 7. Les algues seront ajoutées à un groupe lui-même ajouté à la scène.

14. Affichez les statistiques de nombre d'images par seconde avec stats.min.js.

15. Chargez l'objet 3D requin.obj et son matériau requin.mtl et faites en sorte qu'il décrive des cercles de rayon=30 autour du point (x=0, y=15, z=0).

16. Ecrivez une fonction creation_requins(nb_requins, rayon_min, rayon_max, hauteur_min, hauteur_max) dans laquelle vous clonerez nb_requins auxquels vous appliquerez une mise à l'échelle aléatoire entre 0.2 et 1. Vous les ajouterez à un groupe, lui-même ajouté à la scène. Vous les animerez sur des cercles de rayons aléatoires (entre rayon_min et rayon_max) à des hauteurs aléatoires (entre hauteur_min et hauteur_max) autour de l'origine.
    Pour parcourir les éléments d'un groupe :

```
for( var i=0; i<un_groupe.children.length; i++ )
{
    un_groupe.children[i].position.y += 0.1;          // par exemple
}
```

17. Animez les algues en leur appliquant une rotation en x de :
    0.1 * Math.cos(temps_ecoule_en_secondes).

18. En appuyant sur la touche 'c' du clavier, faites en sorte qu'on puisse faire apparaître/disparaître l'image cockpit.png en plein écran par dessus le rendu 3D (voir TP 5 sur l'affichage de l'image de jumelles).