

Module ReV - Projet

Il faudra rendre pour la fin du semestre dans un dépôt sur le git de l'ENIB le code du projet ainsi qu'un document de 10 à 15 pages.

Il s'agit de proposer une visite d'un musée virtuel. Le musée est un bâtiment de 30m*30m. Il est constitué, dans sa partie sud, d'un hall de 15m*30m et de 10m de haut. La partie nord est elle constituée de trois salles, chacune fermée par une porte coulissante. Ces trois salles ont une hauteur de 5m. Au-dessus de ces trois salles se trouve une mezzanine de 15m*30m. Il est possible d'accéder à cette mezzanine.

Chaque salle accueille 8 à 10 tableaux répartis dans les salles, la mezzanine selon leurs thèmes (à vous de définir la nature des tableaux ou photographies selon vos envies). Dans le hall et sur la mezzanine on trouvera également des statues/mobiles articulées animées.

Question 0 : étudiez le code fourni ainsi que la documentation associée au moteur 3d babylonjs.

Question 1 : proposez une modélisation du musée (sols, plafonds, cloisons). Ces différents éléments n'ont pas de matériaux particuliers. Par contre il faudra tenir compte des collisions et de la gravité. Un escalier permet de rejoindre la mezzanine à partir du hall.

Question 2 : habillez les différents éléments au moyens de matériaux (avec ou sans texture).

Question 3 : « accrochez » aux murs les tableaux, selon les contraintes données ci-dessus.

Questions 4 : installez les statues/mobiles dans le hall et sur la mezzanine.

Question 5 : certaines ouvertures sont dotées de portes (classiques ou coulissantes). Quand le visiteur s'approche d'une porte celle-ci s'ouvre. Quand l'utilisateur s'éloigne de la porte (en y passant ou non), la porte se referme.

Question 6 : placez dans la scène des amers (par exemple des sphères semi-transparentes). Quand le visiteur clique sur un de ces amers il se téléporte vers celui-ci.

Question 7 : pour éviter que l'utilisateur ne soit perdu et désorienté par des trajectoires discontinues faites en sorte que la téléportation corresponde à une trajectoire continue depuis la position actuelle jusqu'à celle de l'amer.

Question 8 : quand on s'approche d'un tableau celui-ci donne son nom pour attirer l'attention du visiteur. Attention à faire en sorte que (i) le visiteur ne soit pas surchargé d'informations et que (ii) seuls les tableaux directement accessibles se signalent.

Question 9 : augmentez le signalement en utilisant des effets lumineux et sonores.

Question 10 : quand le visiteur regarde suffisamment longtemps un tableau celui-ci se décrit.

Annexe 1 : sélection d'objets à la souris

```
window.addEventListener("click",function(event){
    var pickResult = scene.pick(event.clientX, event.clientY) ;
    if(pickResult.hit){ // Si on a cliqué sur un mesh et pas dans le vide
        console.log(pickResult.distance) ; // distance de la caméra au point désigné
        console.log(pickResult.pickedPoint) ; // coordonnées du point désigné
        console.log(pickResult.pickedMesh) ; // référence à l'objet désigné
        console.log(pickResult.pickedMesh.name) ; nom de l'objet désigné
    }
})
```

Annexe 2 : rendu sonore

Son 2d

Création d'un son :

On peut alors invoquer les méthodes **play**, **pause**, **stop** .

On peut associer des options à la restitution d'un son :

```
var options = {
    loop : true,                // joue le son en boucle
    autoPlay : true,           // joue le son après son chargement
    volume : 1.0,              // volume du son entre 0 et 1
    playbackRate : 1.0,        // vitesse de restitution (entre 0 et 1)
    spatialSound : false,      // true si son 3d, false sinon
    maxDistance : 100          // portée de la source sonore
} ;
```

```
var options = {loop:true, autoPlay:true} ;
var sound1 = new BABYLON.Sound("sound1","sound1.mp3",null,options) ;
var sound2 = new BABYLON.Sound("sound2","sound2.mp3",null,options) ;
sound.setVolume(0.5) ;
```

On peut vérifier à tout instant l'état de la restitution d'un son :

```
sound.isPlaying // true si le son est en train d'être joué
sound.isPaused // true si le son est en pause
```

On peut modifier l'état de la restitution d'un son :

```
sound.play \\jouer un son
sound.pause \\mettre un son en pause
sound.stop \\arrêter temporairement de jouer un son
```

Son 3d

Pour créer un son spatialisé seules les options sont modifiées :

```
var sound = new BABYLON.Sound("sound","sound_file",scene,function (){sound.play();},{spatialSound:true}) ;
sound.setPosition(new BABYLON.Vector3(0.0,0.0,20.0))
```

On peut associer un son à un maillage :

```
sound.attachToMesh(myMesh)
```