

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.

Le thème du musée : la « belle époque » période de 1880 à 1914 (https://fr.wikipedia.org/wiki/Belle_Epoque) , dans différents domaines (arts, sciences et techniques, architecture, politiques, ...).

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 doivent modéliser aussi bien la forme des objets 3d que l'aspect de leur surface (matériaux, matériaux avec texture). Il faudra également tenir compte des collisions et de la gravité. Un escalier permet de rejoindre la mezzanine à partir du hall.

Question 2 : « accrochez » aux murs les tableaux, selon les contraintes données ci-dessus. Des statues seront également installées dans le hall et sur la mezzanine. Certaines d'entre elles sont constituées d'un maillage triangulaire alors que d'autres le sont par des objets géométriques articulés animés.

Question 3 : 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 finit par se refermer.

Question 4 : 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. Un tableau, une statue peuvent être un amer.

Question 5 : 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 6 : quand le visiteur regarde suffisamment longtemps un tableau (ou le « touche ») celui-ci se décrit (pour certains au moyen d'un texte écrit pour d'autre d'un texte dit).

Question 7 : Un agent virtuel, incarné dans le musée, attend le visiteur et le guide de tableau en tableau en décrivant à bon escient ces tableaux. Cet agent doit se déplacer sans entrer en collision avec son environnement, d'autres agents ou le visiteur.

Question 8 : Proposez un comportement de l'agent en présence d'un visiteur qui ne le suivrait pas

Il doit choisir entre lui imposer son objectif ou suivre les envies du visiteur voire combiner ces deux comportements.

Si vous utilisez des sons : évitez les « bande-sons » musicales, qui prennent beaucoup de ressources et privilégiez des sons courts à combiner et/ou à associer à des événements.

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 3d

```
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)
```