
SENSIBILISATION A LA PROGRAMMATION MULTIMEDIA

Christophe Vestri

TD 3

Outils de debug :

- Serveur web local:
 - Avoir python (miniconda ou autre)
 - Se placer dans le répertoire html et lancer : `python3 -m http.server`
 - <http://localhost:8000/> firefox ou chrome
- Debug F12 ou sous smartphone android
 - Chrome sur smartphone et page à déboguer
 - Connecté à un smartphone: <chrome://inspect/>
 - <https://developers.google.com/web/tools/chrome-devtools/javascript>

Les principaux problèmes que vous pouvez rencontrer :

- Scène mal éclairée (éclairage directif) :
 - Solution: éclairage ambiant pour commencer
- Objet géométrique non visible
 - Choisissez une position de caméra, placer l'objet devant
 - Faites 1 dessin sur papier pour être sûr de ce que vous faites
 - Problème de clipping?
- Mon modèle 3D ne s'affiche pas:
 - Vérifiez la console de votre navigateur (les erreurs...)
 - Enlevez la texture, mettez un matériau simple
 - Vérifiez l'échelle de votre objet et les positions (voir 2)
 - Utilisez un serveur local (pour Three.js)
 - Utilisez un modèle glTF des exemples de Three.js avant d'utiliser le votre
- Mon objet ne bouge pas
 - Vérifiez que vous appelez bien : `renderer.setAnimationLoop(animate);` ou `engine.runRenderLoop(renderLoop);`
 - Il doit y avoir une variable (angle/position/scale) qui varie, testez avec un breakpoint

Three.js

- <https://threejs.org/>
- <https://davidlyons.dev/threejs-intro>
- <https://threejs.org/examples/>
- Modèles à télécharger : <https://github.com/mrdoob/three.js/tree/master/examples/models>
- lire <https://threejs.org/docs/#manual/en/introduction/Loading-3D-models> puis les exemples avec [Loader](#) ([3DMLoader...](#)) ou [ObjectLoader](#) (json)

Babylon.js

<https://www.babylonjs.com/>
<https://doc.babylonjs.com/features>
<https://doc.babylonjs.com/journey>

Exercice part1 : Géolocalisation et 3D (ThreeJs ou BabylonJS)

- Créez une scène + caméra + light + renderer
- Créez une sphère de rayon 1 (où les valeurs réelles)
- Texturez cet objet avec une image de planète terre
- Créez/trouvez une fonction qui convertisse les Lat/Lon en coordonnées cartésien (https://en.wikipedia.org/wiki/Geographic_coordinate_conversion#Coordinate_system_conversion), attention aux axes que vous utilisez (threejs/babylonjs), aux angles et à l'orientation (N/S)
- Récupérez votre position et placez un marqueur ou objet 3D
- Récupérez les positions de plusieurs pays et afficher des marqueurs avec un marqueur marqueur/objet texturé du drapeau du pays <https://restcountries.com/>
- Ajoutez d'autres données si vous le souhaitez



Exercice part2 : Interactions

- LeafletJs -> 3D
 - Qd on clique sur carte/marqueur, repositionner la terre en 3D sur cet endroit
 - <https://leafletjs.com/examples/extending/extending-3-controls.html> (Handlers)
- 3D -> Leafletjs
 - Qd on clique sur 1 pays, on recentre la carte
 - <https://threejs.org/docs/#api/en/core/Raycaster.intersectObject>
 - https://threejs.org/examples/#webgl_interactive_cubes
 - https://doc.babylonjs.com/features/featuresDeepDive/mesh/interactions/picking_collisions
 - <https://playground.babylonjs.com/#KNE00#1327>

Exercice bonus : Objets géolocalisé dans caméra

- Code exemple:
 - Afficher le flux de la caméra
 - Récupérer géolocalisation et orientation
 - Ajouter des objets géolocaliser (three.js ou babylonJS)
 - Vérifier avec smartphone que ça fonctionne
- Améliorez le système
 - Autres données
 - <https://ar-js-org.github.io/AR.js-Docs/location-based/>
 - Autres idées : Compas 2D/3D: carte 2D + geoloc et directions 3D

Noubliez pas : poussez sur github pour que je puisse corriger