

Three.js

Christophe Vestri

Le mardi 2 février 2021

Plan du cours

- 6 janvier : Intro, github, Capteur/Geoloc en HTML5
- 13 janvier: carto/geo, leaflet, rest Api
- 26 janvier: 2D/3D: Canvas, WebGL et Three.js
- 2 février: Three.js
- 9 février : Projets, exam ou autres exercices

Plan Cours 3

- Rappel dernier cours
- Questions Three.js et réponses
- Exercices:
 - ThreeJs (semaine dernière)
 - Géolocalisation

Html5

- **Acces capteur caméra:**
- **DeviceOrientation, DeviceMotion**
- **Caméra, Audio, Géolocalisation**
- **touchevents/mouse/...**
- **<https://developers.google.com/web/fundamentals/native-hardware/device-orientation/>**

Leafletjs

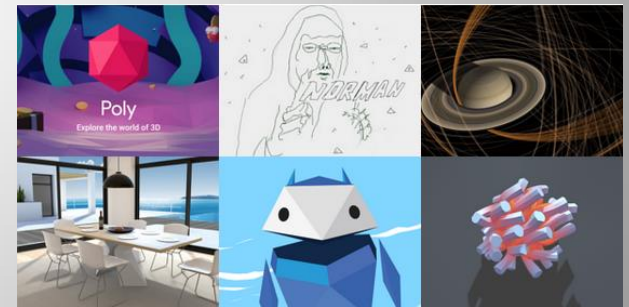
- [leafletjs](https://leafletjs.com/) est une librairie Opensource pour afficher des cartes interactives utiles à la navigation (comme google maps)
- Seulement 33Ko, Tous les browsers
 - Map controls
 - Layers
 - Interaction Features
 - Custom maps



Three.js

The logo for Three.js, featuring the word "THREEJS" in white, uppercase, serif font on a red rectangular background.

- Qu'est-ce que Three.js
 - Couche abstraite et haut niveau de WebGL
 - Librairie javascript pour créer des scènes 3D
 - Cross-plateforme et gratuit
 - Rendus en webGL, CSS3D et SVG
 - <https://threejs.org/>
 - <http://davidscottlyons.com/threejs-intro/>



Questions/Solution Three.js

- Texture
 - Il faut [serveur local](#)
- OrbitControl
 - Attention aux workflow modules -> sinon [JS files link](#)
- Modèles
 - [GLTFLoader](#)
 - Ne pas oublier l'éclairage

Fichiers Locaux/distants

- Avoir python (miniconda ou autre)
- Se placer dans le répertoire html
- `python3 -m http.server`
- <http://localhost:8000/>

<http://duspviz.mit.edu/tutorials/localhost-servers/>

Utile aussi: `chrome.exe --allow-file-access-from-files`

Courses/Examples

- <https://threejs.org/>
- <http://davidscottlyons.com/threejs-intro/>
- <https://threejs.org/examples/>
- <https://codepen.io/rachsmith/post/beginning-with-3d-webgl-pt-1-the-scene>

Three.js

- **Exercice 2 (1h30) :**
 - Créez une scène + caméra + light + renderer
 - Créez un objet générique (sphère ou cube)
 - Texturez cet objet
 - Téléchargez un objet 3D
 - Animez les objets avec les DeviceEvents:
DeviceOrientation, DeviceMotion
 - Ajoutez Fog/pluie ou particules
- **Bonus, mettre un contexte: compas/gyro, système solaire.... ou Physique, animation...**

Three.js

- **Exercice 3 (2h) : Geolocalisation sur terre**
 - Créez une scène + caméra + light + renderer
 - Créez une sphère de rayon 1
 - Texturez cet objet avec image de planète terre
 - Créer une fonction Lat/Lon to cartésien
 - Récupérer votre position et afficher un marqueur
 - Récupérer les positions de plusieurs pays et afficher des marqueurs (couleur différente)
- **Bonus: mettre un modèle 3D à la place du marqueur à notre position, drapeau du pays**