

Sensibilisation à la programmation multimedia

Christophe Vestri

Le mardi 6 octobre 2025

Plan du cours

- 1^{er} TD: Intro, github, carto/geo, leaflet/mapBox, rest Api
- 2em TD: 2D/3D: Canvas, WebGL et Three.js/babylon.js
- 3em TD: Three/babylon.js + Leaflet.js cartographie
- 4em TD: IA

Objectifs du cours:

- Bases de géolocalisation et de la cartographie
- Initiation multimédia: 2D/3D, carto/géo et infographie
- Expérimenter quelques méthodes et outils web geo/3D
- Réaliser un petit projet (combinera ce qu'on a vu)

Plan Cours 4

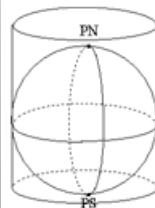
- Rappel dernier cours
- Questions Three.js et réponses
- Exercice:
 - Three.js/babylonJS + Géolocalisation + Rest Api
 - Interactions Device, Leafletjs et Threejs

Géo + Html5 + LeafLet.js

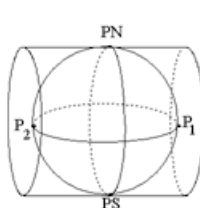
- Repères Géo et carto
- Acces capteur caméra: Géolocalisation, DeviceOrientation, DeviceMotion
- [Leafletjs](http://dmitrybaranovskiy.github.io/leaflet-js/), Mapbox, mapQuest
- Données géolocalisées (REST API)



Représentation cylindrique :

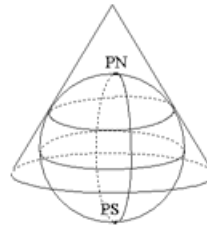


directe

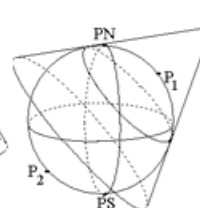


transverse

Représentation conique :



directe



oblique



3D Three.js et Babylon.js

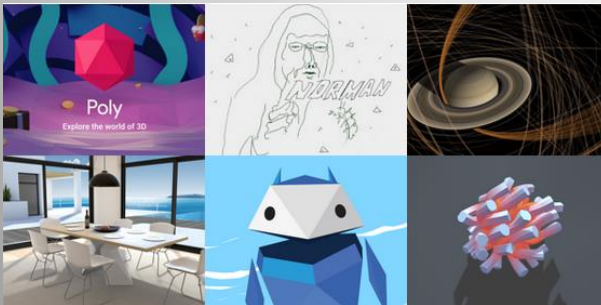
- 3D sur le web



- Libraries 3D de haut niveau de WebGL
- Cross-plateforme et gratuit

Scenes, Cameras, Renderer, Geometry, Materials, Textures, Lights, loaders

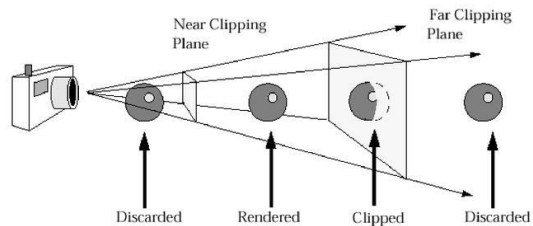
- <https://threejs.org/>
- <https://www.babylonjs.com/>



Rappels avant exercices

3D Clipping

- Objects that are partially within the viewing volume need to be clipped – just like the 2D case

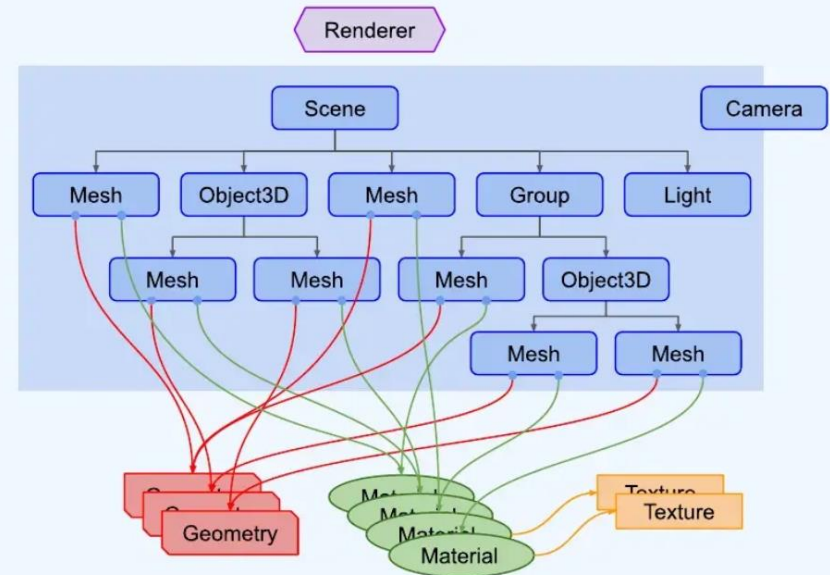


September 15, 2018

Computer Graphics

6

Attention au 3D Clipping



Il vous faut :

- 1 scene
- 1 camera
- 1 light
- Un renderer

Les principaux problèmes

1. Scene mal éclairée (éclairage directif):
 - Solution: éclairage ambiant pour commencer
2. 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?
3. 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 (slide précédent)
 - Utilisez un modèle glTf/obj des exemples de Three.js avant d'utiliser le votre
4. 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

Questions?

- Pb principaux:
 - modèles trop gros/local pour github
 - syntaxe, three avant import
 - CORS lien models ou fichiers json externes
 - scene, lights, model 3D, texture...

Géolocalisation sur terre

- Exercice partie 1: Three.js ou Babylon.js
 - Créez une scène + caméra + light + renderer
 - Créez une sphère de rayon 1 (ou valeurs réelles)
 - Texturez cet objet avec image de planète terre
 - Créer/trouvez une fonction Lat/Lon to cartésien (3D XYZ)
 - Récupérer votre position et afficher un marqueur/objet3D
 - Récupérer les positions de plusieurs pays et afficher le drapeau sur un objet: <https://restcountries.com/v3.1/all>

Chaque fonctionnalité
donne des points

Faites les 2 parties en
parallèle

Géolocalisation sur terre

- Exercice Partie 2: Ajoutez un leaflet à côté
 - LeafletJs -> 3D

Qd on clique sur carte/marqueur -> positionner la terre sur cet endroit (leaflet handlers)
 - 3D ->Leafletjs

Qd on clique sur 1 pays (Raycaster), on recentre la carte
 - Demo

Publiez sur votre Github
pour que je puisse corriger

Exercice bonus

- **Partie 3 (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 ca fonctionne

- **Améliorez le système**

- Autres données
 - <https://ar-js-org.github.io/AR.js-Docs/location-based/>
 - Autre...

**Publiez sur votre Github
pour que je puisse corriger**