Sensibilisation à la programmation multimedia

Christophe Vestri

Le mardi 8 octobre 2024

Plan du cours

- 17 septembre : Intro, github, Capteur/Geoloc en HTML5
- 24 septembre: carto/geo, leaflet/mapBox, rest Api
- 8 octobre: Three.js et Babylon.sj
- 15 octobre: Three.js/Babylon.js + cartographie

Objectifs du cours

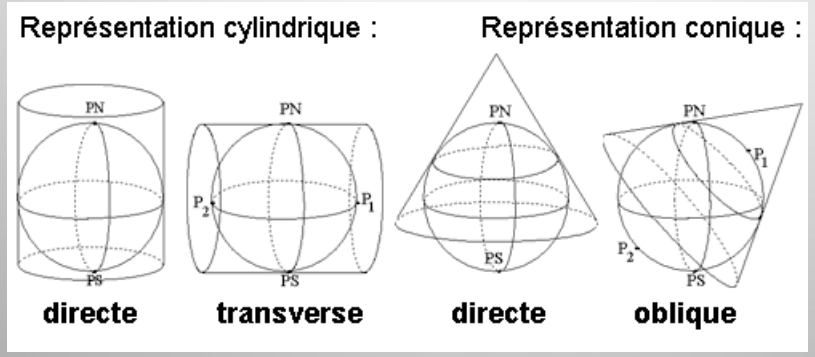
- Bases de géolocalisation et de la cartographie
- Expérimenter outils multimédia/carto/geo/3D
- Evaluation:
  - 25% par rendu de TD
  - N'oubliez pas de commiter, pusher sur Github
  - **Vérifiez le fonctionnement**

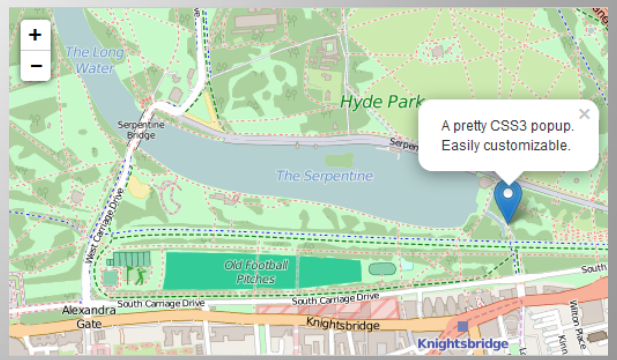
Plan Cours 3

- Rappel dernier cours
- WebGL
- Three.js (exercice – 2h)
- Babylon.js (exercice – 1h30)

Géo + Html5 + LeafLet.js

- Repères Géo et carto
- Acces capteur caméra: Géolocalisation, DeviceOrientation, DeviceMotion
- Leafletjs, Mapbox, mapQuest
- Données géolocalisées (REST API)





Retour exercices

- **Vérifiez bien la fonctionnalité de votre code avec smartphone**
- **Evitez de recopier, développez**
  - https://france-geojson.gregoiredavid.fr/repo/departements.geojson
  - **ChatGPT?**
  - **Pas à jour (2018)**
  - **Pas la consigne -> requête serveur**

Graphique en HTML

- Canvas 2D
- SVG: Scalable Vector Graphics
- CSS3D: pour des effets de rendu 3D
- WebGL: pour de la 3D basique
- Three.js ou Babylon.js pour de la vrai 3D

WebGL

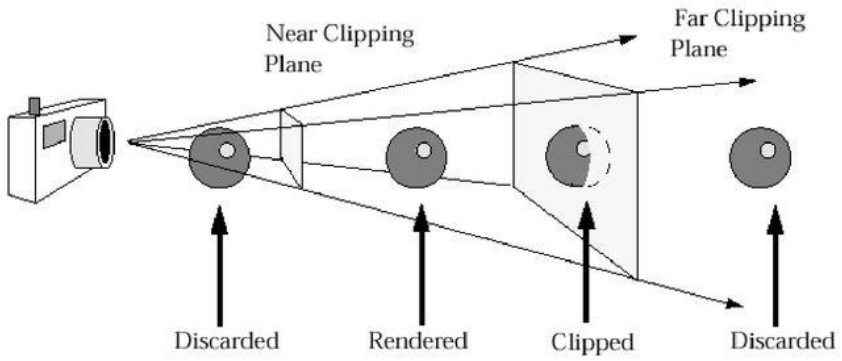
- Qu'est-ce que WebGL
  - Cross plateforme et libre de droits
  - OpenGL ES (OpenGL simplifié pour l'embarqué) dans le Web (HTML5)
  - Bonne intégration Html et mécanisme d'évènements
  - DOM API pour affichage 2D et 3D
  - Langage de type script (pas de compilation)
  - Accélérations matérielles et GPU (GLSL)

WebGL

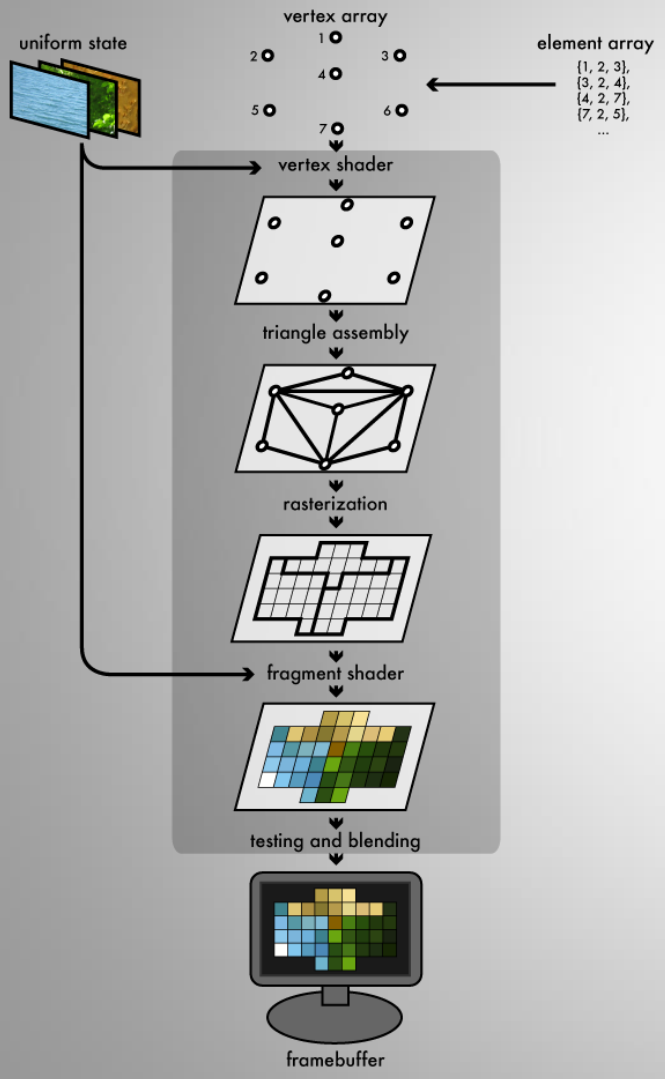
- **Computer graphics**

3D Clipping

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



WebGL Pipeline



Données brutes

Tessellation
Transformation des objets complexs

Fonction prédéfinie
Transformation et éclairage

Fonction programmable
Vertex Shader

Backface Culling
Élimination des faces cachées

Clipping
Élimination des parties cachés

Viewport Transform
Élimination des parties hors champs

Triangle Setup
Transformation des triangles en pixels

Fonction prédéfinie
Texture, filtre et transparence

Fonction programmable
Pixel Shader

Fog Blending
Calcul de brouillard

Visibility Testing
Test de visibilité

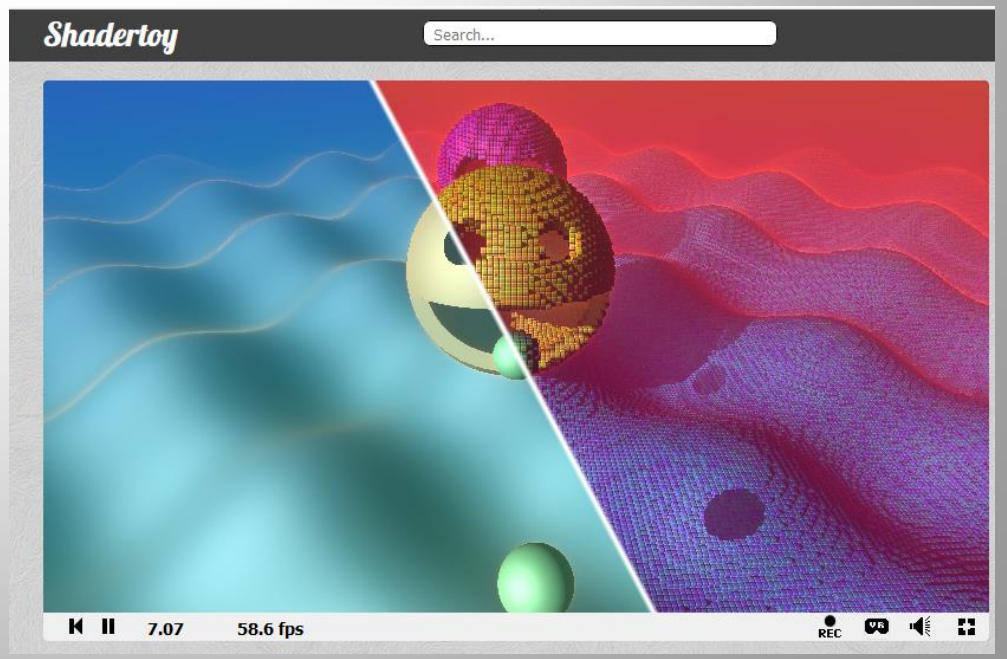
Frame Buffer
Écriture du pixel visible

WebGL

- **Low-level API**
  - **GLSL OpenGL Shading Langage**
  - **Machine d'état: OpenGL Context**
  - **Calcul de matrices et transformations**
  - **Buffers de vertex: positions, normals, color, texture**
  - **Depth buffer, Blending, transparency**
  - **Lighting, Cameras…**
  - **https://developer.mozilla.org/fr/docs/Web/API/WebGL_API**
  - **https://webglfundamentals.org/webgl/lessons/fr/**

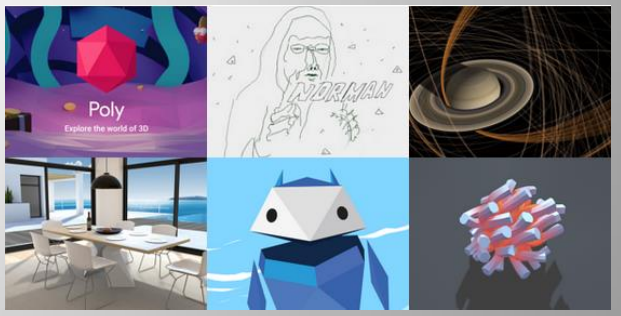
WebGL Exemples

- https://www.khronos.org/webgl/wiki/Tutorial
- https://webglfundamentals.org/
- https://www.shadertoy.com/



Three.js

- **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/**



Fonctionalités

THREE.JS

- Scenes, Cameras, Renderer,
- Geometry, Materials, Textures
- Lights, Shadows
- Shaders, Particles, LOD
- Loaders: Json compatible Blender, 3D max, Wavefront OBJ, Autodesk FBX
- Animation, Trackballcontrols, Math Utilities
- https://davidlyons.dev/threejs-intro/

Outils de debug

- En local (besoin pour charger modèles 3D):
  - Avoir python (miniconda ou autre)
  - Se placer dans le répertoire html
  - python3 -m http.server
  - http://localhost:8000/ firefox ou chrome
- Smartphone android -> Chrome
- https://developers.google.com/web/tools/chrome-devtools/javascript
  - Simulation de smartphone (F12)
  - Connecté à un smartphone: chrome://inspect/

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 sur 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 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

Exercice 1 – Three.js

- **Exercice 2 (2h)**
  - https://threejs.org/docs/index.html#manual/en/introduction/Installation
  - https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene
- **Objectif: testez les éléments de base d'une scène 3D:**
  - **Créez une scène + caméra + light + renderer**
  - **Créez un objet générique (sphère ou cube)**
  - **Texturez cet objet**
  - **Téléchargez un objet 3D**
  - **Animez un objet avec le smartphone, DeviceEvents, DeviceOrientation et/o DeviceMotion**
  - **Ajoutez ce que vous voulez: Fog/pluie ou particules**
- **bonus: mettre un contexte**

Publiez sur votre Github
pour que je puisse corriger

Babylon.js

- **Qu’est-ce que Babylon.js**
  - **Idem Three.js:** librairie javascript pour créer des scènes 3D, cross-plateforme et gratuit
  - **avec d’autres fonctionalités:** Gaussian Splat Rendering, WebXR, Apple Vision Pro…
  - https://www.babylonjs.com/

  - https://doc.babylonjs.com/journey

Exercice 2 – Babylon.js

- **Exercice 2 (1h30)**
  - **https://doc.babylonjs.com/journey**
- **Objectif: testez les éléments de base d'une scène 3D:**
  - **Créez une scène + caméra + light + renderer**
  - **Créez un objet générique (sphère ou cube)**
  - **Texturez cet objet**
  - **Téléchargez un objet 3D**
  - **Animez un objet avec le smartphone, DeviceEvents, DeviceOrientation et/ou DeviceMotion**
  - **Ajoutez ce que vous voulez: Fog/pluie ou particules**
- **bonus: mettre un contexte**

même chose que Three.js

Publiez sur votre Github pour que je puisse corriger