

Canvas WEBGL et Three.js

Christophe Vestri

Le mardi 24 mars 2020

Plan du cours

- 10 février : Intro Capteur et Geoloc/access en HTML5
- 17 mars: Carto/géo, Leaflet
- 23 mars: Canvas, WebGL et Three.js
- 30 mars: Aframe, AR.js et VR
- 6 avril : WebRTC: Projets

Projet final

- **Projet final**
 - GéoLocalisation
 - Capteurs mouvement/orientation
 - UI et scene 3D, interaction
- **Présentation avec confinement**
 - Lien web pour que tout le monde teste
 - Présentation sous Meet (ou autre mais prévenez moi bien avant et testez)

Plan Cours 3

- Rappel dernier cours
- Canvas et SVG
- CSS3D: exercice
- WebGL et Three.js
 - Exercice: ThreeJs et Device Events
- Projet final

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



Solution exercice 3

- **Utiliser Leaflet**

```
L.geoJSON(geojsonfeature).addTo(map);
```

<https://leafletjs.com/examples/geojson/>

Ensuite requête html avec format Geojson

```
function reqListener () { L.geoJson(JSON.parse(this.response)).addTo(map); }
```

```
var xmlhttp = new XMLHttpRequest();
```

```
xmlhttp.addEventListener("load", reqListener);
```

<https://codepen.io/micka-l-appel/pen/XWbqwpj>

```
xmlhttp.open("GET", "http://monapi")
```

```
xmlhttp.send()
```

<http://plnkr.co/edit/ODe622?preview>

https://www.w3schools.com/js/js_json_http.asp

WebRTC

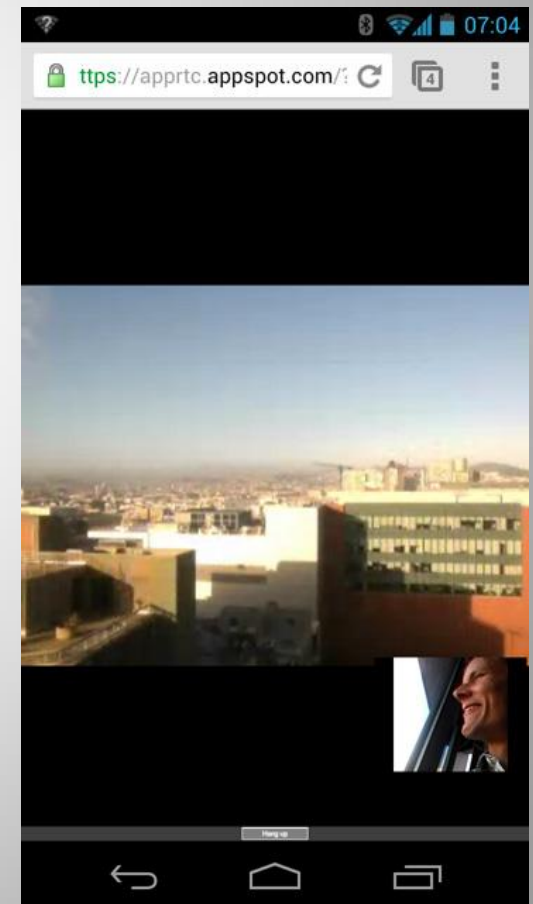


- **Qu'est-ce que c'est?**
 - **Real-Time Communications (RTC) à travers une simple API**
 - **3 taches:**
 - **Acquisition audio et video (Mediastream)**
 - **Communication audio et video (RTCPeerConnection)**
 - **Communication d'autres données (RTCDataChannel)**
 - **<https://webrtc.org/>**

WebRTC



- Chrome, Chrome for Android
- Firefox
- Opera



Graphique en HTML

- **Canvas 2D**
- **SVG: Scalable Vector Graphics**
 - Ex1=30min
- **CSS3D: pour des effets de rendu 3D**
 - Ex2=45min
- **WebGL: pour de la 3D basique**
- **Three.js: pour de la 3D plus poussée**
 - Ex3=1h

CANVAS HTML

- **Element Html pour dessiner**

```
<canvas id="mycanvas" width="500" height="300"></canvas>  

```

- **Context 2D (dessin) ou 3D (WebGL)**

- **Acces en javascript (dans le DOM)**

```
var canvas = document.getElementById('mycanvas');  
var myimg = document.getElementById('scream');  
var ctx = canvas.getContext('2d');  
ctx.drawImage(myimg, 10, 120);  
ctx.fillStyle = 'green';  
ctx.fillRect(30, 30, 100, 100);
```



https://developer.mozilla.org/fr/docs/Tutoriel_canvas

<https://www.alsacreations.com/tuto/lire/1484-introduction.html>

SVG

- **Format graphique d'image XML**
- **Image sans perte**
- **Manipulé en javascript (dans le DOM)**
- **Manipulé par CSS**
- https://www.w3schools.com/graphics/svg_intro.asp
- <http://edutechwiki.unige.ch/fr/Tutoriel SVG avec HTML5>
- SVG on canvas: <https://github.com/canvg/canvg>

Exercice 1 (30min)

Christophe.VESTRI@univ-cotedazur.fr

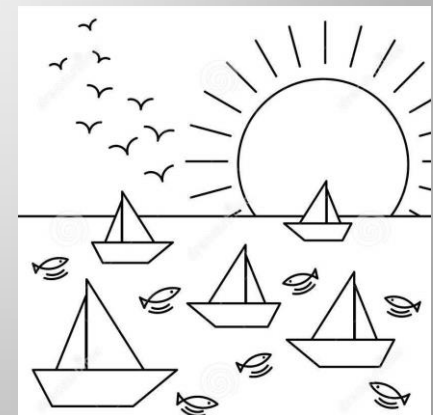
- **Dessiner dans un canvas et un svg (séparés)**

- **Canvas:**

- Choisir et afficher une image avec ciel, mer, bateaux
 - Dessiner bateaux + soleil: Rectangle + triangle + ronds

- **SVG:**

- Dessiner ciel, mer et soleil
 - Quand on passe curseur sur la voile, elle change de couleur
 - Quand on clique sur Soleil ciel gris, sinon bleu



CSS 3D Transform

- Tous les elements (graphiques) peuvent être transformés:
 - Shift, rotation, perspective....
- https://www.w3schools.com/css/css3_3dtransforms.asp
- <https://drafts.csswg.org/css-transforms/>
- <https://keithclark.co.uk/labs/css-fps/>

Exercice 2 CSS3D (45min)

- <https://desandro.github.io/3dtransforms/>

Exercice:

- **Construire 1 dé 6 faces**
- **Lancer le dé aléatoirement avec 1 bouton**
- **Bonus:**
 - **utiliser des images ou dessiner points**
 - **Le dès doit tourner**

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

WebGL - 3D Canvas graphics - OTHER

Usage % of all users

Global 93.75%

Method of generating dynamic 3D graphics using JavaScript, accelerated through hardware

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63		10.2				
			64		10.3				4
1 11	1 16	58	65	11	11.2	all	64	1 11.8	6.2
	1 17	59	66	11.1	11.3				
		60	67	TP					
		61	68						

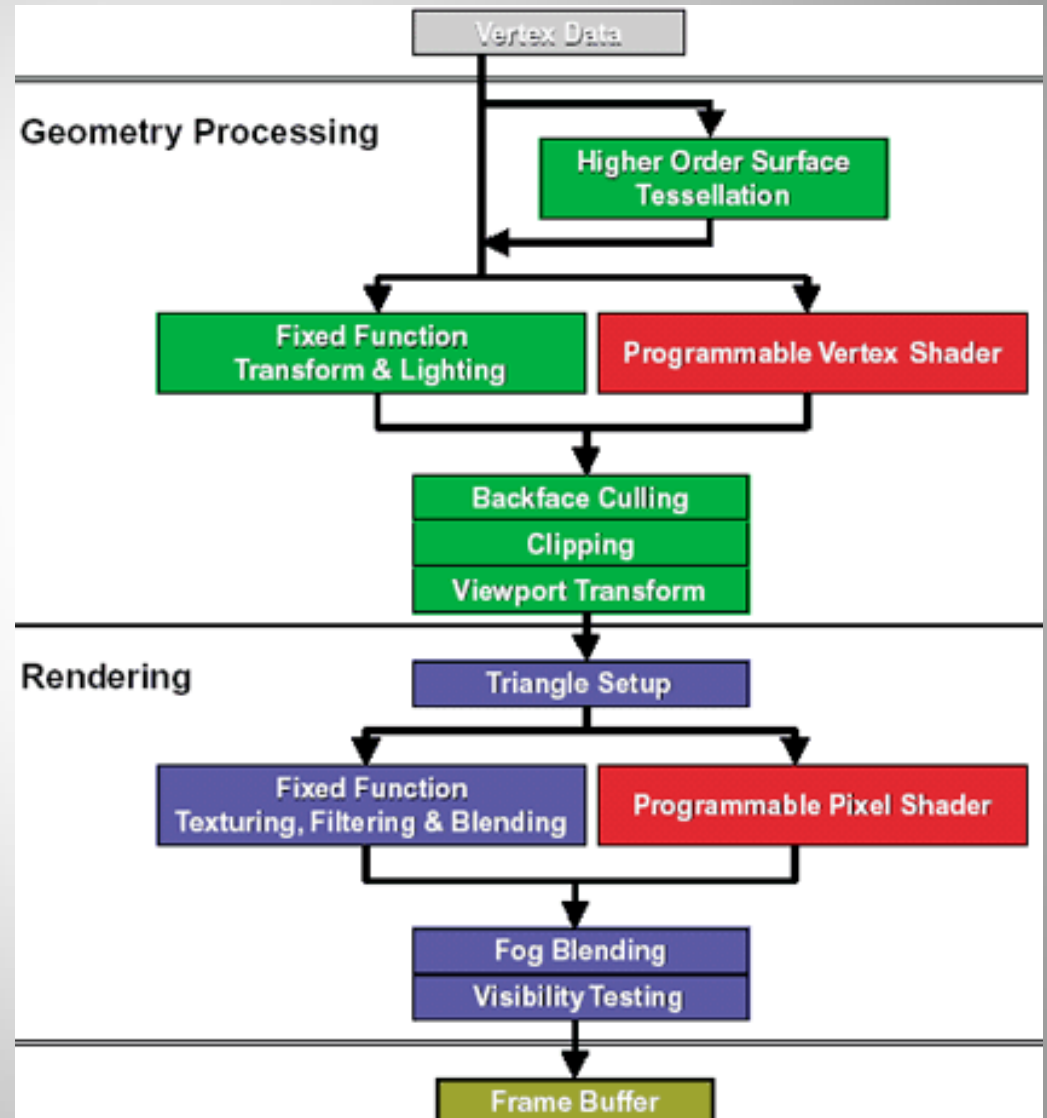
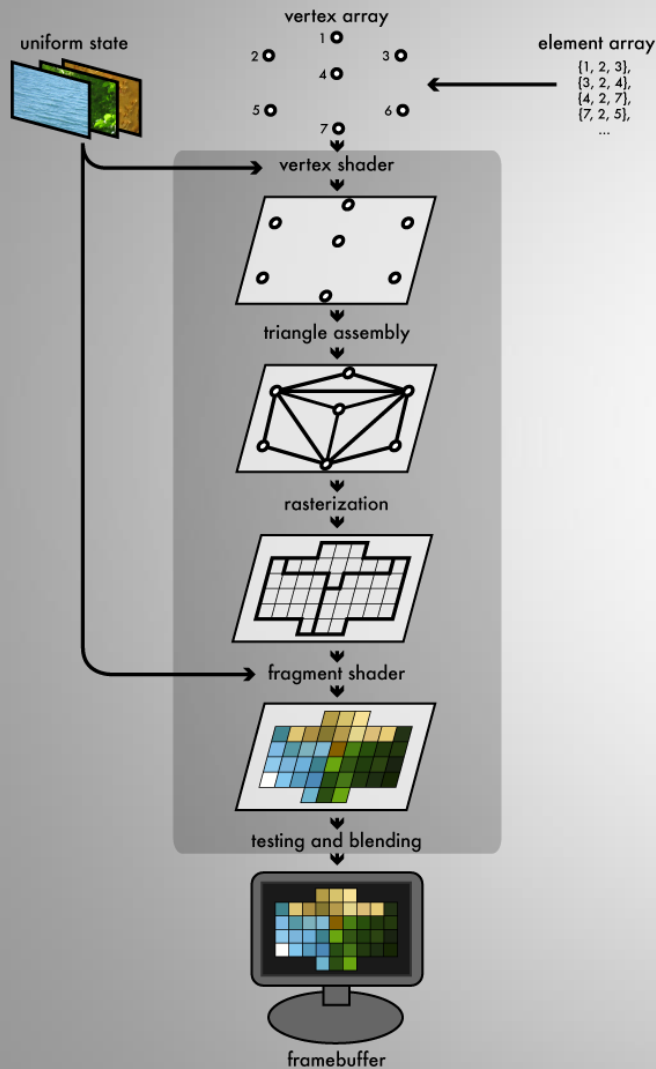
— Blacklist:

<https://www.khronos.org/webgl/wiki/BlacklistsAndWhitelists>

WebGL

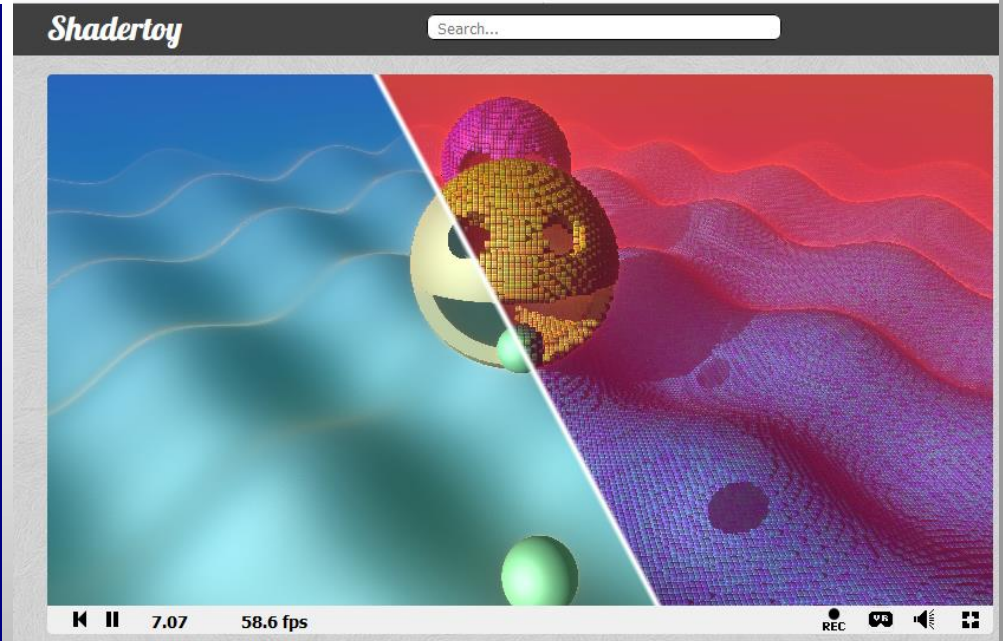
- **Low-level API**
 - GLSL OpenGL Shading Language
 - Machine d'état: OpenGL Context
 - Calcul de matrices et transformations
 - Buffers de vertex: positions, normals, color, texture
 - Depth buffer, Blending, transparency
 - Lighting, Cameras...
 - <http://www.webgltutorials.org/>
 - <https://www.khronos.org/webgl/wiki/Tutorial>

WebGL Pipeline



WebGL Examples

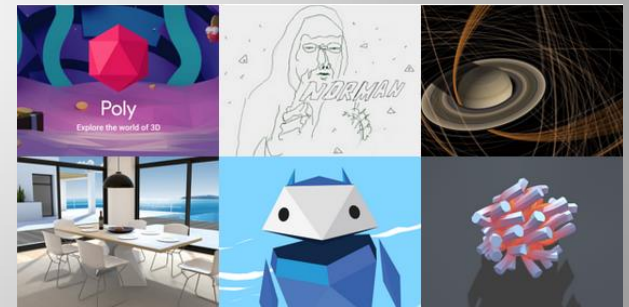
- https://developer.mozilla.org/fr/docs/Web/API/WebGL_API
- <https://webglfundamentals.org/>
- <https://www.shadertoy.com/>



Three.js

The logo for Three.js, featuring the word "THREE" in white serif font and ".js" in a smaller white sans-serif font, all 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/>



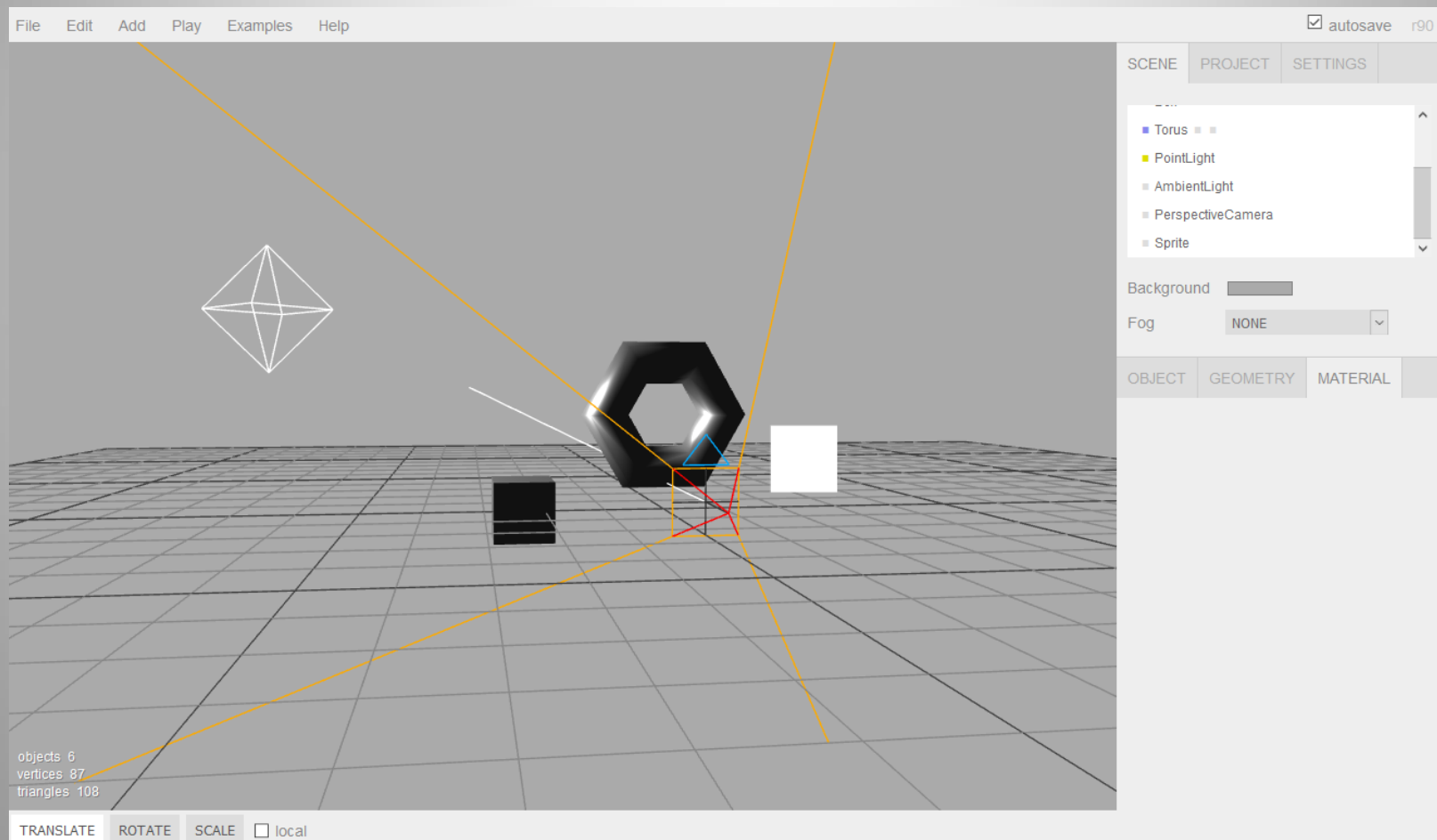
Fonctionnalités

THREEJS

- 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

Threejs Editor

- <https://threejs.org/editor/>



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://classroom.udacity.com/courses/cs291>
- <https://codepen.io/rachsmith/post/beginning-with-3d-webgl-pt-1-the-scene>

Exercice 3 – Three.js

- Exercice 3 (à rendre avec exercice 1 et 2):
 - 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
- Option, mettre un contexte: compas/gyro, système solaire.... ou [Physijs](#) ou autre

Projet final

- **Projet final**
 - GéoLocalisation
 - Capteurs mouvement/orientation
 - UI et scene 3D, interaction
- **Présentation avec confinement**
 - Lien web pour que tout le monde teste
 - Présentation sous Meet (ou autre mais prévenez moi bien avant et testez)