

WebRTC WEBGL et Three.js

Christophe Vestri

Le mardi 12 mars 2019

Plan du cours

- 26 février : Intro Carto/géo, Leaflet
- 5 mars: Capteur et Geoloc/access en HTML5
- 12 mars: WebRTC, WebGL et Three.js
- 19 mars: Aframe, AR.js et VR
- 26 mars : Projets

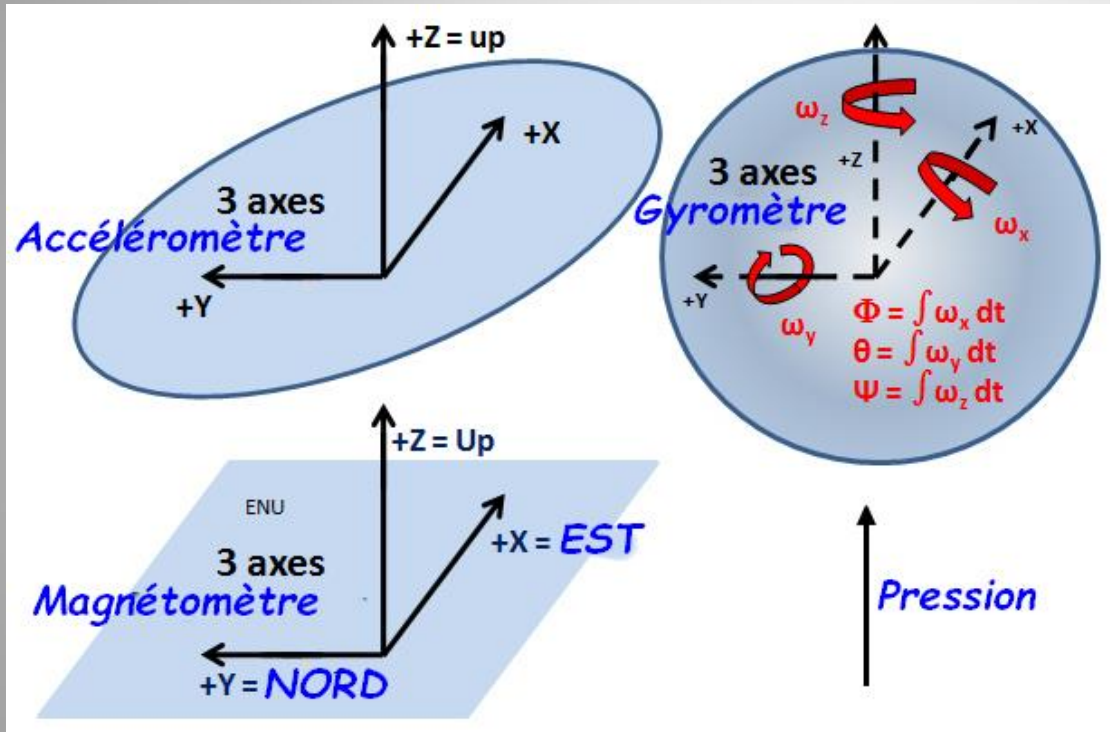
Projet final

- **Projet final**
 - **GéoLocalisation**
 - **Capteurs mouvement/orientation**
 - **UI et scene 3D, interaction**
 - **Exemples:**
 - **Compas 2D/3D: carte 2D + geoloc et directions 3D**
 - **Objets 3D animés avec interaction smartphone**
 - **Réalité augmentée (HtIm5/JS)**

Plan Cours 3

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

Capteurs smartphones



C'est donc un système à 10 capteurs d'attitude qui est embarqué

= 3 accéléromètres
+ 3 gyromètres
+ 3 magnétomètres
+ 1 pression

Html5

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

Pour tester sur un Mobile

- Créer un compte sur <https://www.000webhost.com/>
- Ou tout autre free webhosting site
- Uploader vos fichiers
- Tester avec votre smartphone

WebRTC

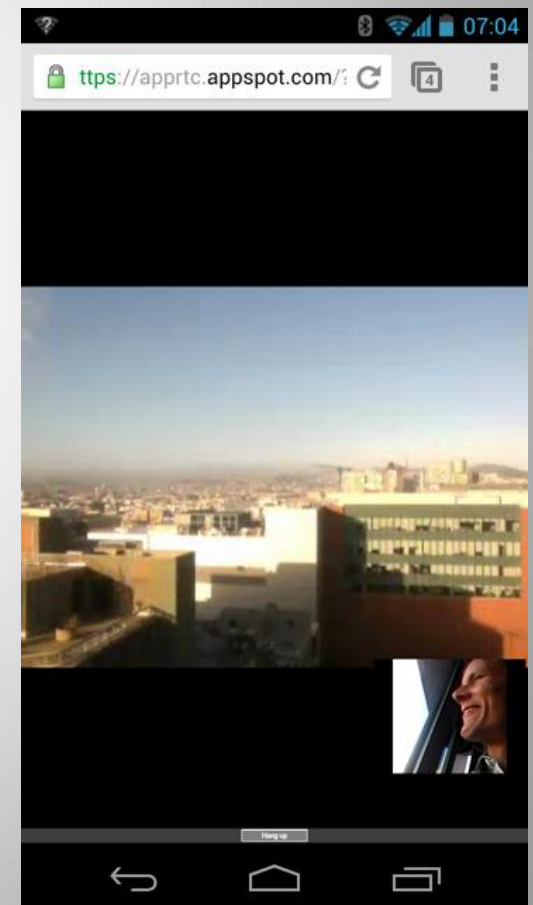


- **Qu'est-ce que c'est?**
 - **Real-Time Communications (RTC) à travers une simple API**
 - **3 taches:**
 - **Acquisition audio et video (Mediastream)**
 - **Communication audio et video (RTCPeerConnection)**
 - **Communication d'autres données (RTCDataChannel)**
 - **<https://webrtc.org/>**

WebRTC



- Chrome, Chrome for Android
- Firefox
- Opera



Example 1

- <https://github.com/webRTC/samples/blob/gh-pages/src/content/getusermedia/audio/js/main.js>

```
11 // Put variables in global scope to make them available to the browser console.
12 var audio = document.querySelector('audio');
13
14 var constraints = window.constraints = {
15     audio: true,
16     video: false
17 };
18
19 function handleSuccess(stream) {
20     var audioTracks = stream.getAudioTracks();
21     console.log('Got stream with constraints:', constraints);
22     console.log('Using audio device: ' + audioTracks[0].label);
23     stream.oninactive = function() {
24         console.log('Stream ended');
25     };
26     window.stream = stream; // make variable available to browser console
27     audio.srcObject = stream;
28 }
29
30 function handleError(error) {
31     console.log('navigator.getUserMedia error: ', error);
32 }
33
34 navigator.mediaDevices.getUserMedia(constraints).
35     then(handleSuccess).catch(handleError);
```

Example 2

- <https://webrtc.github.io/samples/>
- <https://appr.tc/>
- <https://appear.in/3dvtech>

Graphique en HTML

- **Canvas et SVG**
 - context 2D. Exploré la dernière fois
- **SVG: Scalable Vector Graphics**
- **CSS3D: pour des effets de rendu 3D (Ex1)**
- **WebGL: pour de la 3D basique**
- **Three.js: pour de la 3D plus poussée (Ex2)**

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](http://edutechwiki.unige.ch/fr/Tutoriel_SVG_avec_HTML5)

CSS3D

- https://www.w3schools.com/css/css3_3dtransforms.asp
- <https://drafts.csswg.org/css-transforms/>
- <https://keithclark.co.uk/labs/css-fps/>

A finir

Exercice CSS3D

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

Exercice:

- Construire 1 dé 6 faces
- Lancer le dé aléatoirement avec 1 bouton

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						

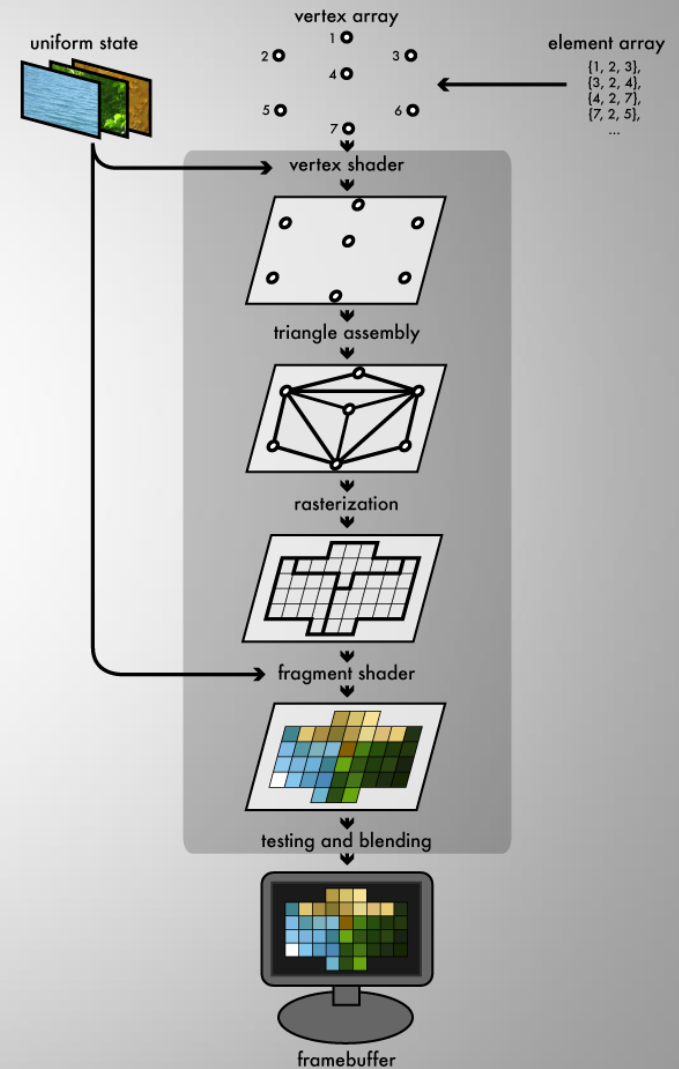
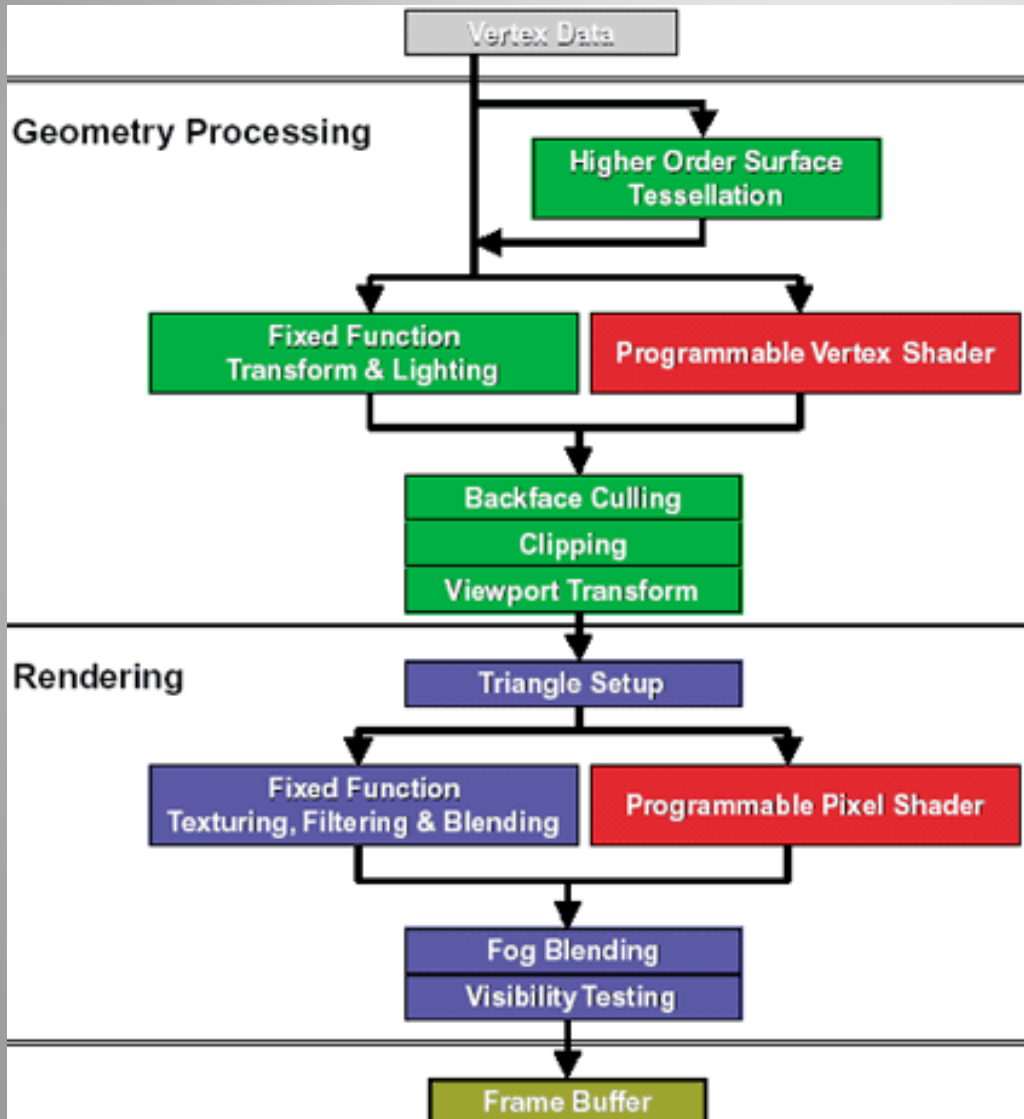
— Blacklist:

<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>

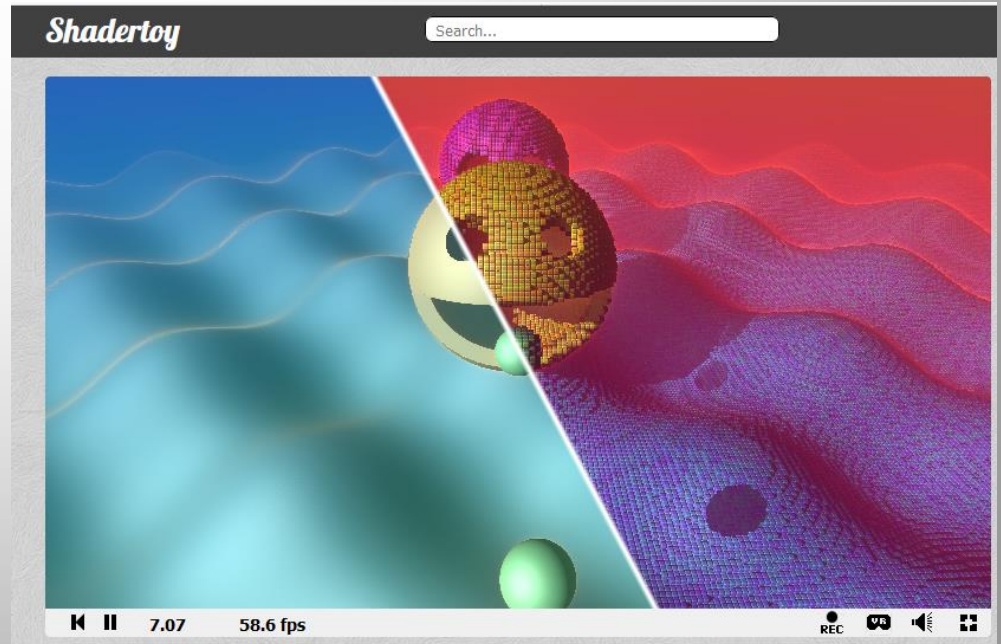
WebGL Pipeline



WebGL Examples

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

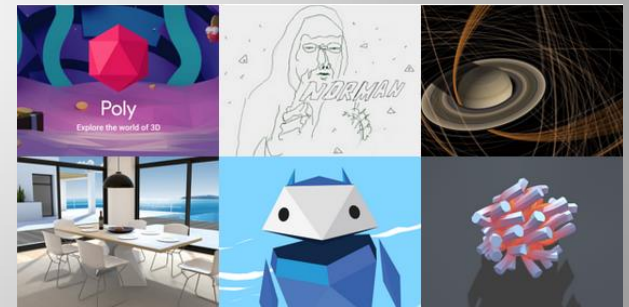
Trouver un exemple simple



Three.js

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



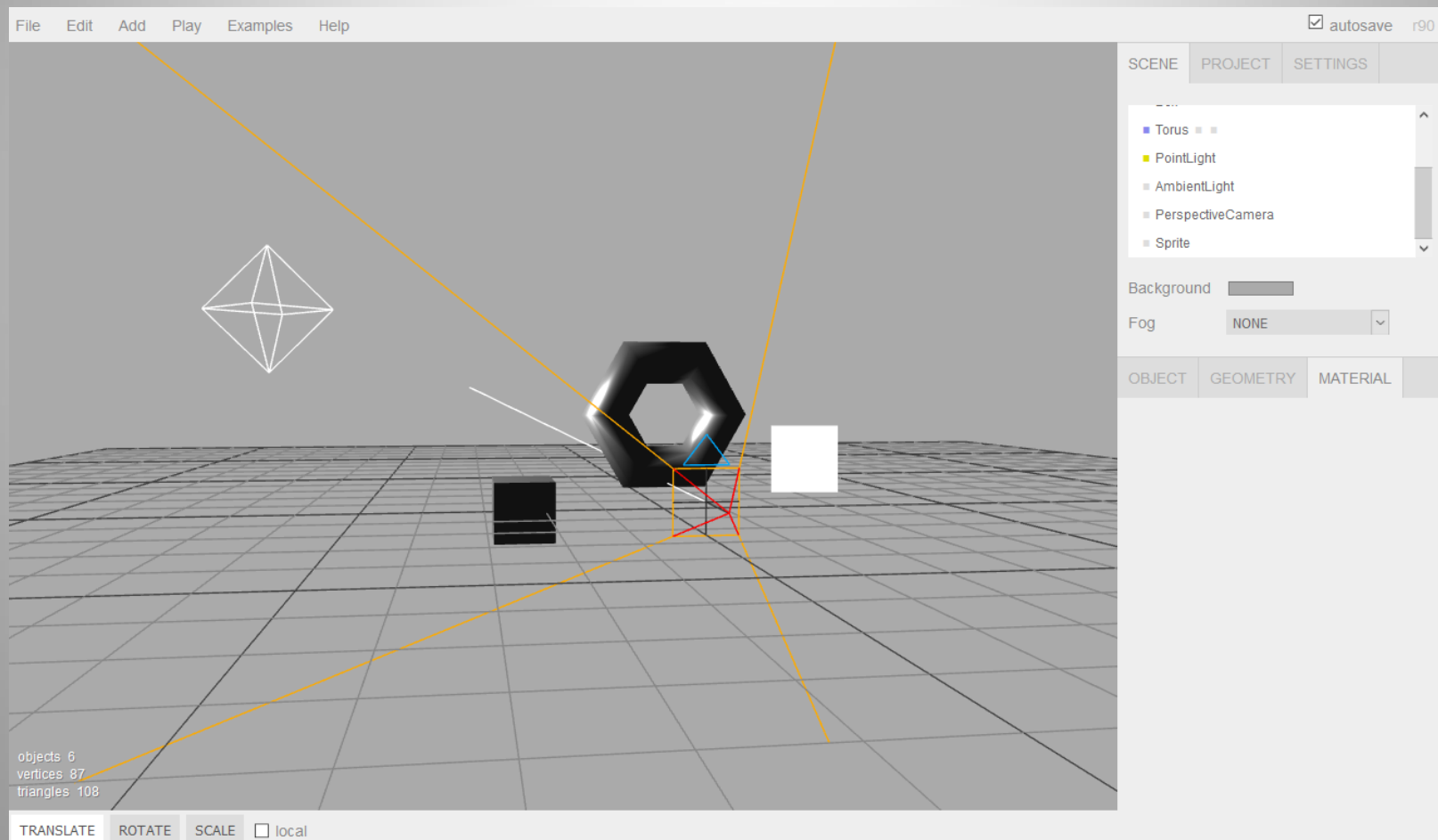
Fonctionnalités

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/>



Courses/Examples

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

Exercice 2 – Three.js

- **Exercice (à rendre):**
 - Créez une scene + caméra + light + renderer
 - Créez un objet générique (sphere 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....**

Rappel

<https://github.com/artmobilis/ArtMobilis-js/wiki/fr-Configuration-framework-nodejs-ionic-android>

- **Chrome:**

- Bloque getUserMedia pour les fichiers locaux
- Lancer avec --disable-web-security pour du debug
- Navigator.getUserMedia plus supporté -> MediaDevices.getUserMedia()
- Il faudrait utiliser adapter.js
- Attention: exemples pas mis à jour -> utilisez Firefox

- **Firefox:**

- Version 40 et +: pb avec les vieilles cartes graphique blacklistées
- Installer version 31 pour du debug (marche sur mon laptop)