

# **WEBVR et Aframe**

**Christophe Vestri**

Le mardi 20 mars 2018

# Objectifs du cours

- Bases de Cartographie et géographie
- Outils de Cartographie, Géographie en Html5 et JS
- Expérimenter quelques méthodes et outils
- un peu de VR
- Réaliser un projet en RA/VR/Carto
- Evaluation:
  - Présence (20%)
  - Participation en classe (40%)
  - Projet (40%)

# Plan du cours

- 20 février : Intro Carto/géo Leaflet
- 6 mars: Capteur et Geoloc/access en JS et Unity
- 13 mars: WebRTC, WebGL et Three.js
- 20 mars: WebVR, Aframe et AR.js
- 27 mars : Projets

# Plan Cours 4

- Rappel dernier cours
- Réalité Virtuelle
- Outils pour la VR
  - WebVR
  - Aframe
  - AR.js
  - VR in Unity
- Projet final

# WebGL

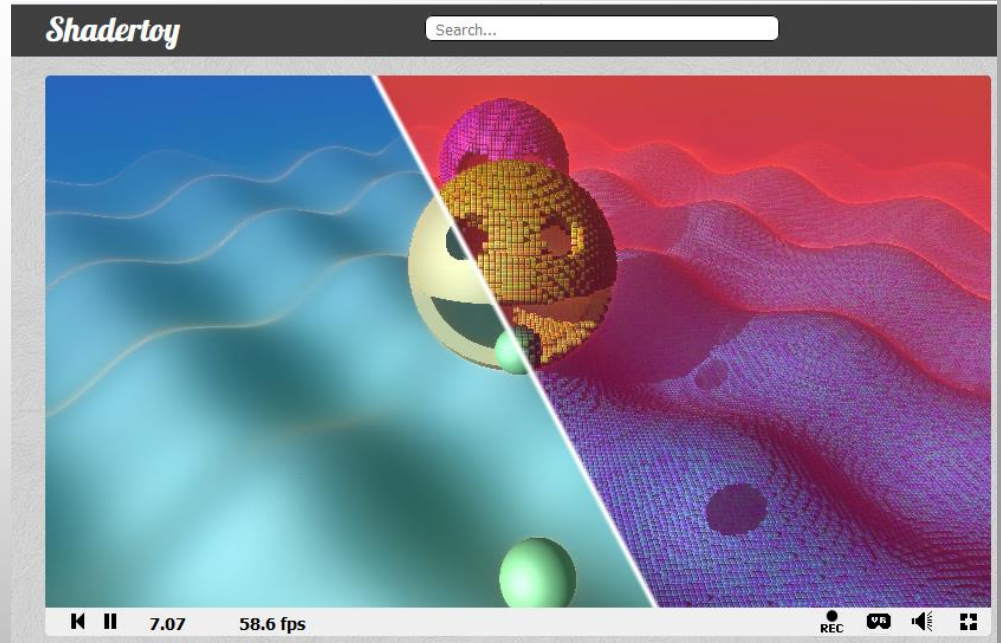


- **Qu'est-ce que WebGL**
  - Cross plateforme et libre de droits
  - OpenGL ES (OpenGL simplifié pour l'embarqué) dans le Web (HTML5)
  - Bonne intégration Html et mécanisme d'évènements
  - DOM API pour affichage 2D et 3D
  - Langage de type script (pas de compilation)
  - Accélérations matérielles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

# WebGL Examples

- [https://developer.mozilla.org/fr/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/fr/docs/Web/API/WebGL_API)
- <https://webglfundamentals.org/>
- <https://www.shadertoy.com/>

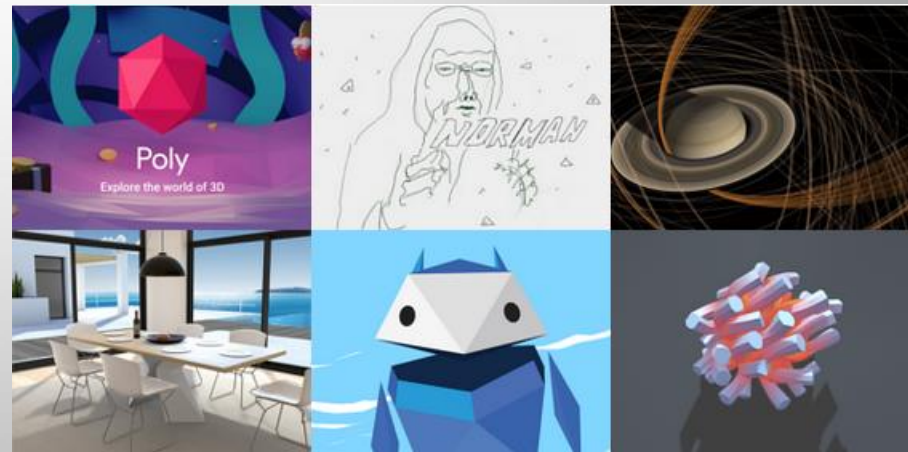
Trouver un exemple simple



# Three.js

# THREEJS

- 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/>



# 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



# Exercice – Three.js

- Créez une scene + caméra + light + renderer
- Créez un objet
- Texturez cet objet
- Téléchargez un objet
- Animez l'objet (mouvement + déformation)
- Ajoutez Fog/pluie ou particules

# Réalité Virtuelle

# Continuum réalité-virtualité



Environnement  
réel



Réalité  
augmentée



Réalité  
virtuelle



Environnement  
virtuel



# Réalité Virtuelle

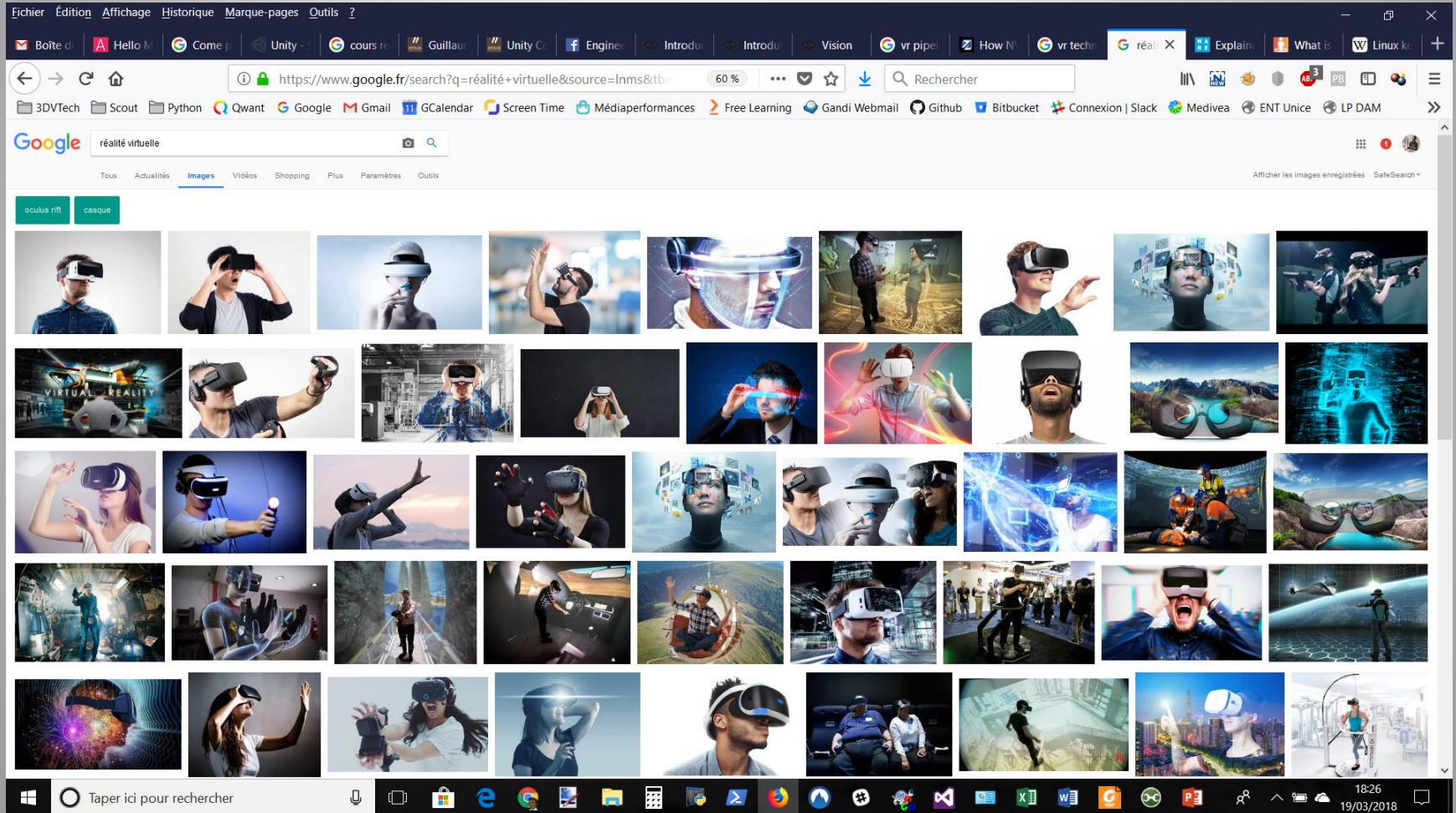
- **Objectif : Immersion**
- **Définition:** Virtual reality (VR) is a computer-simulated environment that can simulate physical presence in places in the real world or imagined worlds. Virtual reality can recreate sensory experiences, including virtual taste, sight, smell, sound, touch, etc.
- **La réalité virtuelle permet de s'extraire de la réalité physique pour changer virtuellement de temps, de lieu, et/ou de type d'interaction avec le monde simulé**

# Objectif immersion





# Réalité Virtuelle





# Applications



# Objectif immersion





# Principaux éléments de la VR

- **Monde Virtuel**
  - Généralement en 3D avec rendu infographie
- **Immersion**
  - Perception de présence monde non physique
- **Retour Sensoriel**
  - Maximum de sens: visuel, auditif, haptique...
- **Interactivité**
  - Augmente la sensation d'immersion

# Principaux éléments de la VR

LOW  
IMMERSION



Engagement

To achieve the first stage of immersion, the user must accept an investment of time, and the experience must match the user's preferences and offer an entry level of control that is acceptable.



**Barrier 1:**  
Accessibility

- Experienter preference is met e.g. if the content is about sport and the user doesn't like sport they are unlikely to engage with the content.
- Controls and expertise level are intuitive and easy to learn.

**Barrier 2:**  
Time

- Willingness to invest time. If the experienter is unwilling to invest time in the experience then they are unlikely to become engaged.
- Availability of time. The experienter must have time available to partake in the experience.

Engrossment

To achieve the second stage of immersion, the user must accept their surroundings (environment) and successfully ignore/remove external distractions.



**Barrier 3:**  
Depth of World

- The world looks visually believable.
- The experienter is set believable tasks.
- The plot / narrative is believable.

**Barrier 4:**  
Distractions

- Removal of all exterior distractions.

HIGH  
IMMERSION

Total Immersion

To achieve the final stage of immersion, the user must achieve a psychological sense of being in a virtual environment (presence) and a state of 'flow' whereby they are so involved in the activity that nothing else seems to matter.



**Barrier 5:**  
Empathy

- The growth of attachment / addiction.
- The embodiment of a character.

**Barrier 6:**  
Awareness

- A lack of awareness of time (flow).
- A loss of awareness of the real world (presence).

# Types de VR

Non immersif



Semi-immersif

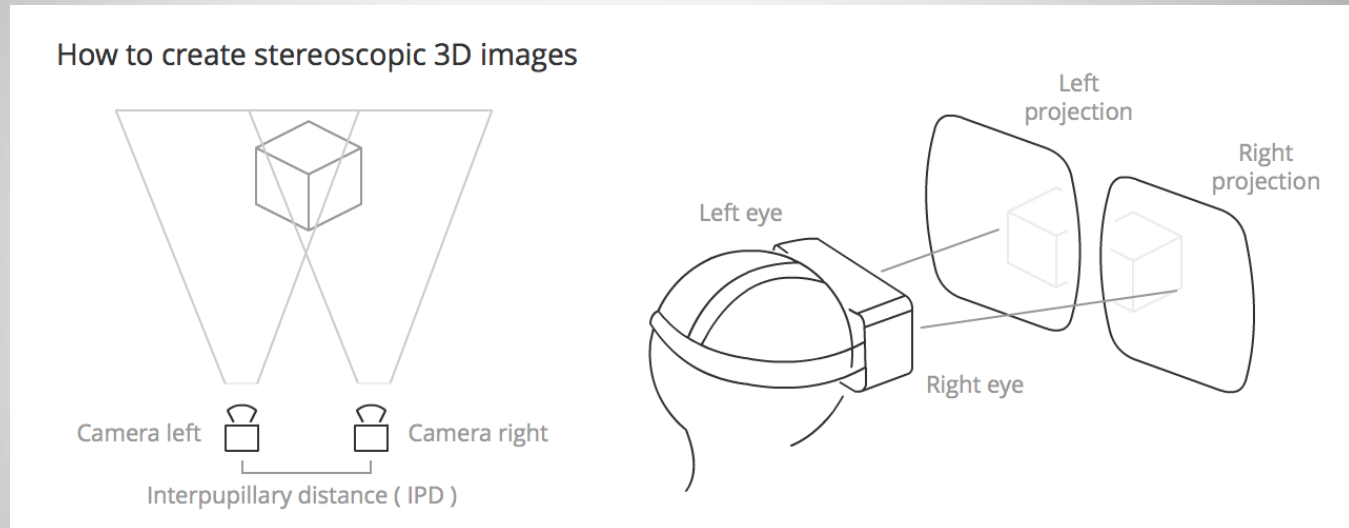


Immersif

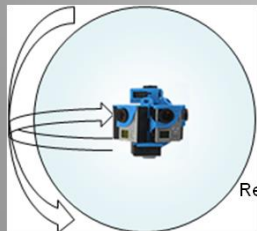


# Comment faire de la VR

- 2 vues



- Infographie ou vidéos



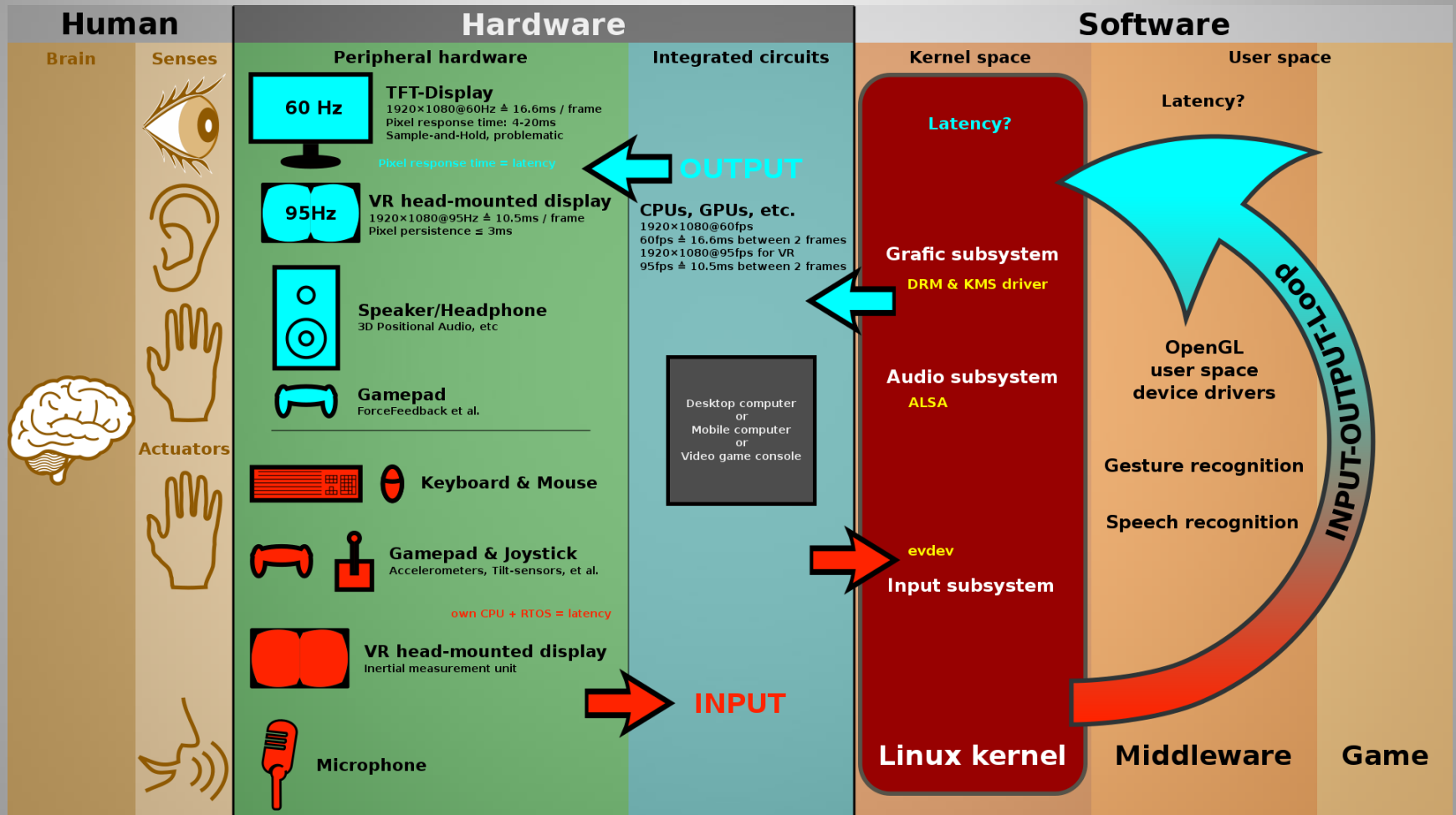
2D panoramic image



# Fonctionnement

- **Un PC/Console/Smartphone**
- **Head-Mounted Display**
  - Lentilles, écrans, processeurs
  - Magnétomètres, accéléromètres, gyroscopes
- **Des capteurs d'entrées**
  - Joystick, manettes, baguettes, gants
  - Plateformes de mouvements

# Systeme de VR





# Rendu vidéo



## Virtual Reality by numbers

< 20ms

motion-to-photon latency

120°  
HORIZONTAL  
FIELD OF VIEW

135°  
VERTICAL  
FIELD OF VIEW

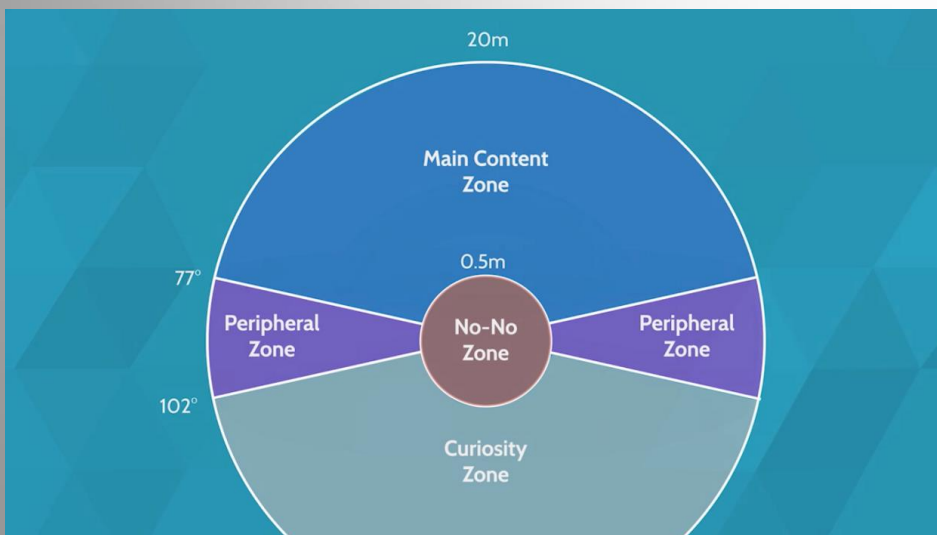
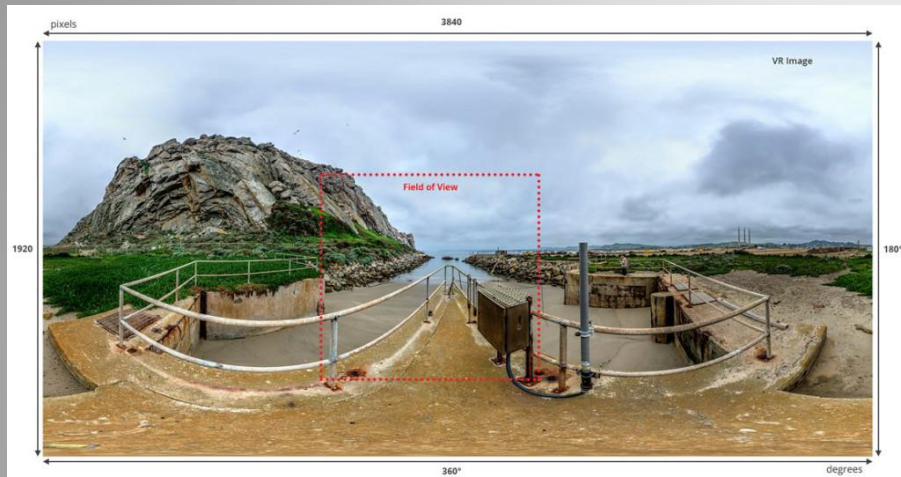
16k  
RESOLUTION

A 16K resolution (or around 128 Mpixels) is needed to display a Stereo Field of View resolution of approximately 116 Mpixels

source: [www.amd.com/whitepaperImmersiveVR](http://www.amd.com/whitepaperImmersiveVR)

©2015 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, LiquidVR, and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

# Rendu vidéo



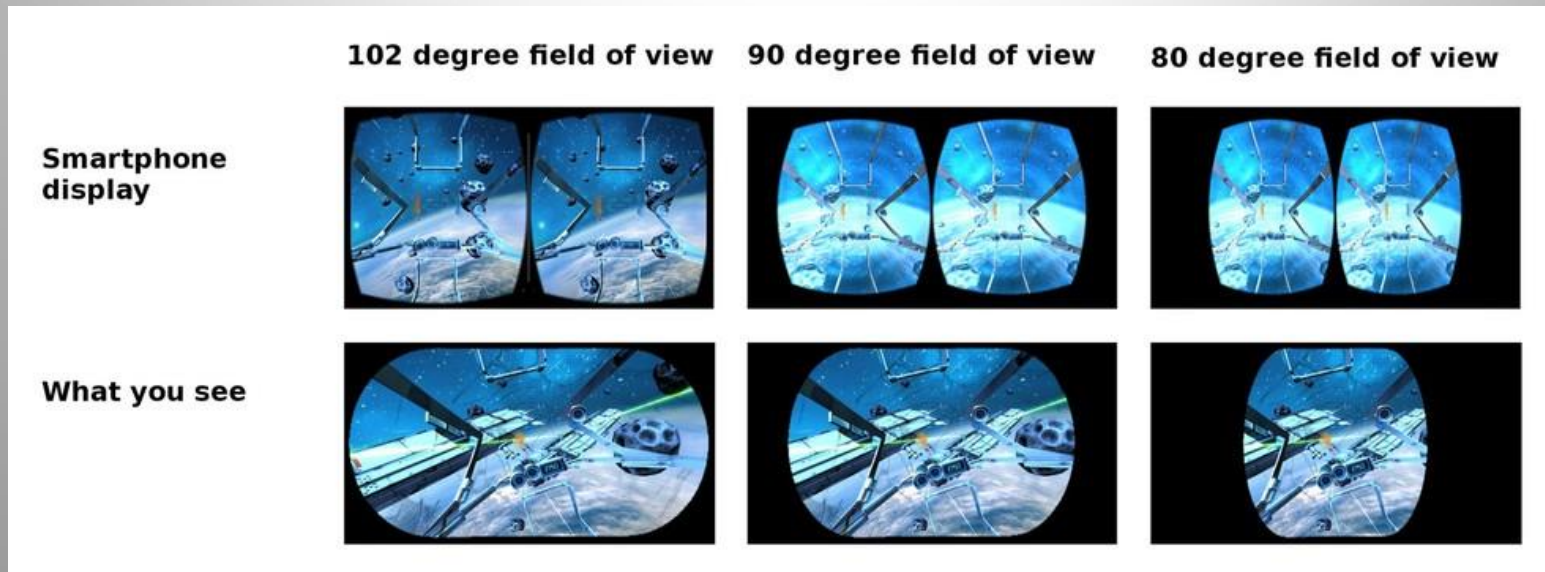
- 1 20/200 ← 1080p VR Video < 480p Regular Video
- 2 20/100 ← 4K VR Video = 720p Regular Video
- 3 20/70
- 4 20/50
- 5 20/40 ← 6K VR Video = 1080p Regular Video
- 6 20/30
- 7 20/25
- 8 20/20 ← 60 pixels in 1 deg: ~16K VR Video (Ideal)
- 9
- 10
- 11

Source: Visbit, Inc.



# Caractéristiques clés

- Champ de vue
- Résolution
- Fréquence de rafraichissement
- Latence



# Principaux casques de VR

- **Oculus Rift**



- **Samsung Gear**



- **Google cardboard**



## HTC Vive



## Playstation VR



# Principaux casques de VR



Gear VR



Playstation VR



HTC Vive



Oculus Rift



Daydream View

Resolution

2560 x 1440

1920 x 1080

2160 x 1200

2160 x 1200

2560 x 1440  
Pixel XL  
1920 x 1080  
Pixel

Per Eye

1280 x 1440

960 x 1080

1080x 1200

1080 x 1200

1280 x 1440  
Pixel XL  
960x 1080  
Pixel

Field of View

100°

100°

110°

110°

90°

# Plus d'infos

- Cours de VR:

- <http://www.ensiie.fr/~bouyer/RVSI/>
- <https://moodle.insa-rouen.fr/course/view.php?id=936>
- <https://eu.udacity.com/course/introduction-to-virtual-reality--ud1012>
- <https://fr.coursera.org/specializations/virtual-reality>
- <https://www.edx.org/course/creating-virtual-reality-vr-apps-uc-san-diegox-cse190x>

- Guides pratiques:

- <https://developer.oculus.com/design/latest/concepts/book-bp/>

# WebVR

- **Qu'est-ce que WebVR**
  - **Débuté en 2014**
  - **Javascript Api pour immersion 3D, Réalité virtuelle dans le navigateur**
  - **Supporté par Mozilla VR team et Google Chrome**
  - **Version 1.1 en 2017, remplacement par WebXR**
  - **<https://webvr.rocks/> (video)**

# WebVR

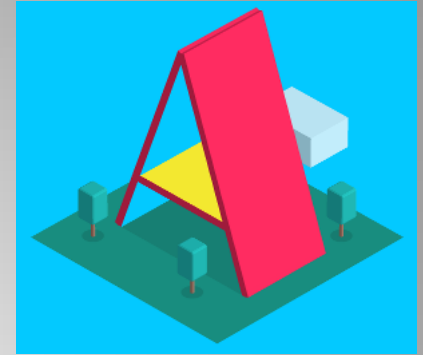
- Liens

- Spec: <https://immersive-web.github.io/webvr/spec/1.1/>
- WebXR <https://immersive-web.github.io/webxr/>

- Quelques demos:

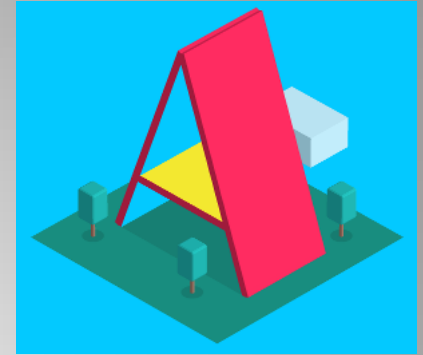
- <https://webvr.info/samples/>
- <https://experiments.withgoogle.com/webvr>

# AFrame



- **Qu'est-ce que Aframe:**
  - **Framework Opensource**
  - **Compatibles avec tous les navigateurs**
  - **Maintenu par Mozilla VR team**
  - **Basé sur three.js, utilise WebVR et WebGL**
  - **Langage déclaratif de type Html**
  - **Système d'entités-composantes**
  - **Inspecteur/éditeur visuel**

# AFrame



- Liens:

- <https://aframe.io/>
- <https://aframe.io/docs/0.8.0/introduction/#getting-started>

- Exemples

- <https://aframe.io/aframe/examples/>
- [http://fluxo.fr/dokuwiki/doku.php?id=labz\\_connectic\\_lab3d](http://fluxo.fr/dokuwiki/doku.php?id=labz_connectic_lab3d)

- Cours:

- <https://aframe.io/aframe-school/#/>



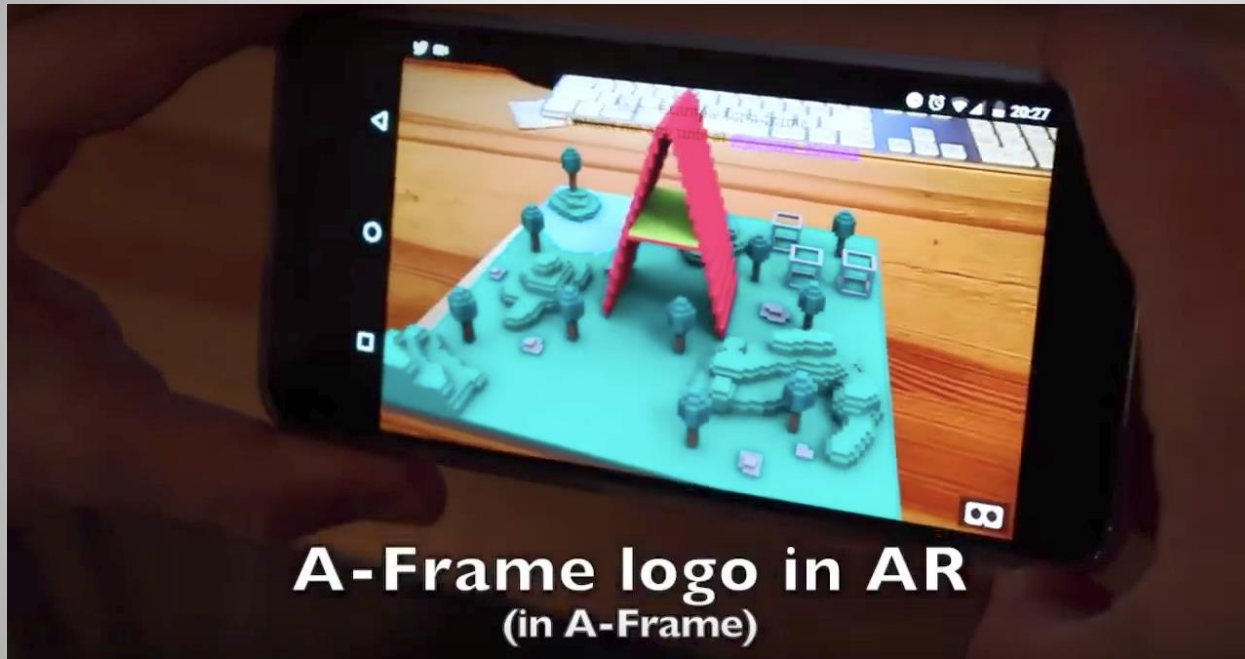
# Exercices

- Construire une scène Egyptienne
  - Camera, lumière jaune, sky
  - Environnement Egypte + fog (sable)
  - Une primitive texturée + modèle 3D
- Exemples:
  - <https://aframe.io/docs/0.8.0/guides/building-a-basic-scene.html>
  - [www.3dvtech.com/DemoPhoto360VR/](http://www.3dvtech.com/DemoPhoto360VR/)
  - <http://www.3dvtech.com/DemoVR/>

# AR.js

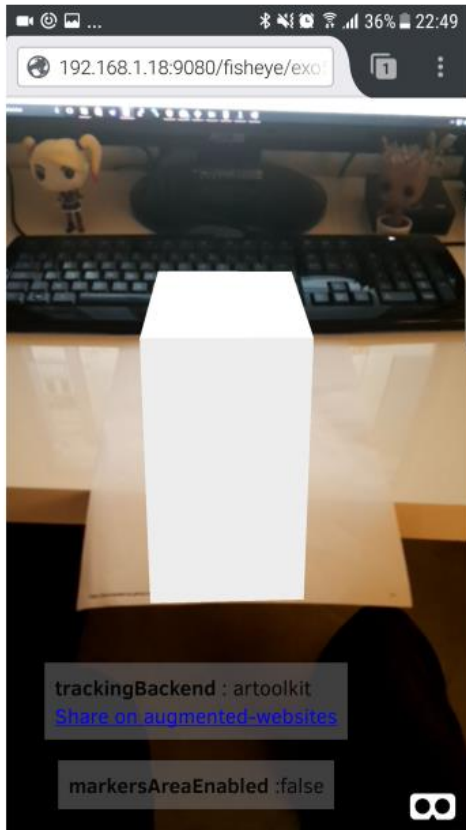


- Artoolkit + Aframe



- <https://aframe.io/blog/arjs/>
- <https://github.com/jeromeetienne/ar.js>

# AR.js



```
1. <script src="https://aframe.io/releases/0.7.0/aframe.min.js"></script>
2. <script
   src="https://jeromeetienne.github.io/AR.js/aframe/build/aframe-ar.js"></script>
3. <body style='margin : 0px; overflow: hidden;'>
4.   <a-scene embedded arjs>
5.     <a-marker preset=hiro>
6.       <a-box></a-box>
7.     </a-marker>
8.     <a-entity camera></a-entity>
9.   </a-scene>
10. </body>
```

# Unity 3D

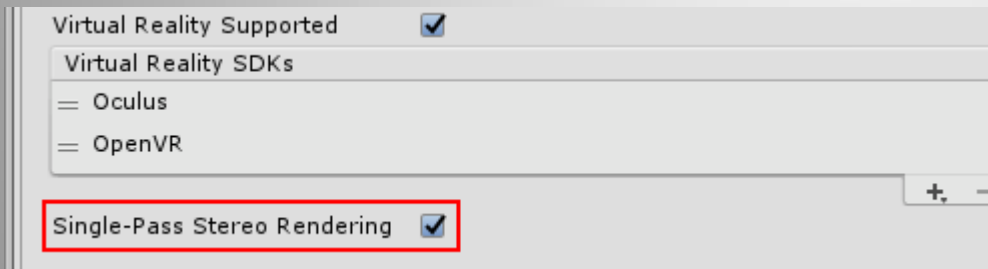
- Unity XR
- <https://docs.unity3d.com/Manual/XR.html>
- <https://developers.google.com/vr/?hl=fr>

Download the Google VR SDK for Unity

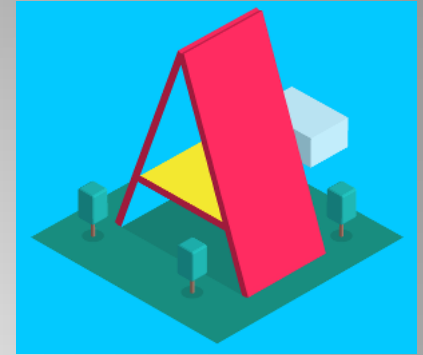
- Download the latest `GoogleVRForUnity_*.unitypackage` from the [releases page](#).

The SDK includes the following demo scenes for Daydream and Cardboard:

Scene	Description
HelloVR	Simple VR game in which you find and select a geometric shape
KeyboardDemo	Daydream: Shows keyboard input on a UI canvas
PermissionsDemo	Daydream: Shows a correct user permissions request flow
VideoDemo	Shows various ways to use stereo or 360° video through playback or remote streaming



# Exercices



- **AR.Js:**
  - Lancez un serveur local:  
`python -m http.server, wamp ...`  
<http://localhost:8000/>
  - Augmenter les exemples en local, (modèle 3D)
    - <https://github.com/jeromeetienne/ar.js>
- **Unity3D:**
  - <https://unity3d.com/fr/learn/tutorials/s/xr>
  - <https://developers.google.com/vr/?hl=fr>

# Projet final

- **Projet final**
  - Un projet avec de la GéoLocalisation/capteurs/VR
  - Un peu de ce qu'on a vu: RA, VR, GeoLoc ou Capteur
  - Exemples:
    - Mélanger aruco/jsfeat/leaflet/geoloc/deviceApi
    - Afficher des objets Géolocalisés flottants, se balader sur une carte ou labyrinthe (Unity/js?)

# 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)