

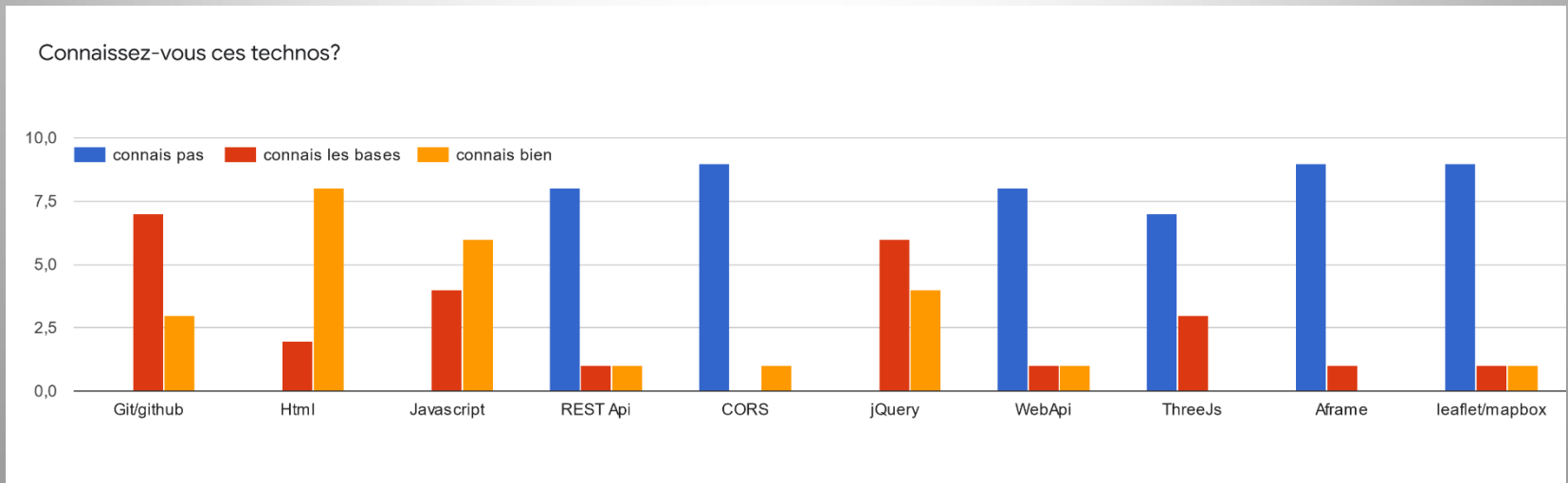
# **Sensibilisation à la programmation multimedia**

**Christophe Vestri**

Le vendredi 6 octobre 2023

# Plan du cours

- 12 septembre : Intro, github, Capteur/Geoloc en HTML5
- 19 septembre: carto/geo, leaflet/mapBox, rest Api
- Vendredi 6 octobre: 2D/3D: Canvas, WebGL et Three.js
- 10 octobre: Projet d'évaluation



# Plan Cours 3

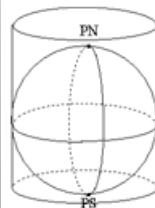
- Rappel dernier cours
- Canvas et SVG
- CSS3D
- WebGL et Three.js
  - Exercice: ThreeJs et Device Events

# Géo + Html5 + LeafLet.js

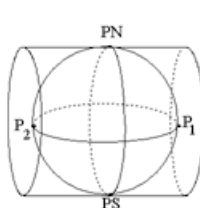
- Repères Géo et carto
- Acces capteur caméra: Géolocalisation, DeviceOrientation, DeviceMotion
- [Leafletjs](http://dmitrybaranovskiy.github.io/leaflet-js/), Mapbox, mapQuest
- Données géolocalisées (REST API)



Représentation cylindrique :

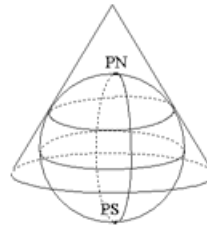


**directe**

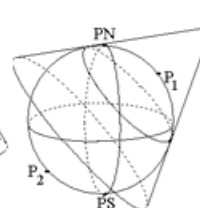


**transverse**

Représentation conique :



**directe**



**oblique**



# Solution exercice 3

- **Utiliser Leaflet**

Pour un fichier local:

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

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

- **sinon requête html avec format Geojson**

```
$.getJSON(url1, function(result)  
{L.geoJSON(result).addTo(map);});
```

```
function reqListener () { L.geoJson(JSON.parse(this.response)).addTo(map); }  
var xmlhttp = new XMLHttpRequest();  
xmlhttp.addEventListener("load", reqListener);  
xmlhttp.open("GET", "http://monapi")  
xmlhttp.send()
```

[https://www.w3schools.com/js/js\\_json\\_http.asp](https://www.w3schools.com/js/js_json_http.asp)

# Graphique en HTML

- **Canvas 2D**
- **SVG: Scalable Vector Graphics**
- **CSS3D: pour des effets de rendu 3D**
- **WebGL: pour de la 3D basique**
- **Three.js: pour de la 3D plus poussée**
  - **Ex1 Canvas+SVG=45min**
  - **Exo2 Threejs: 2h**

# 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://developer.mozilla.org/fr/docs/Tutoriel_canvas)

[https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas\\_image](https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas_image)

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

# SVG-Scalable Vector Graphic

- **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](https://www.w3schools.com/graphics/svg_intro.asp)
- <http://edutechwiki.unige.ch/fr/Tutoriel SVG avec HTML5>
- SVG or canvas: <https://css-tricks.com/when-to-use-svg-vs-when-to-use-canvas/>



# CSS 3D Transform

- Tous les elements (graphiques) peuvent être transformés:
  - Shift, rotation, perspective....
- [https://www.w3schools.com/css/css3\\_3dtransforms.asp](https://www.w3schools.com/css/css3_3dtransforms.asp)
- <https://drafts.csswg.org/css-transforms/>
- <https://keithclark.co.uk/labs/css-fps/>

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

- Blacklist (quelques vieux smartphones):

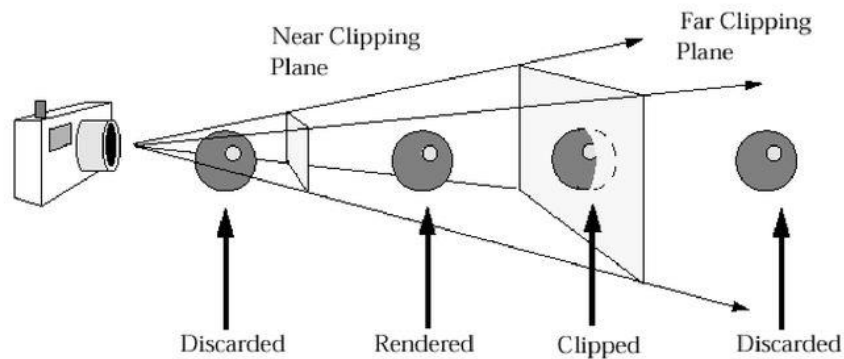
<https://www.khronos.org/webgl/wiki/BlacklistsAndWhitelists>

# WebGL

- **Computer graphics**

## 3D Clipping

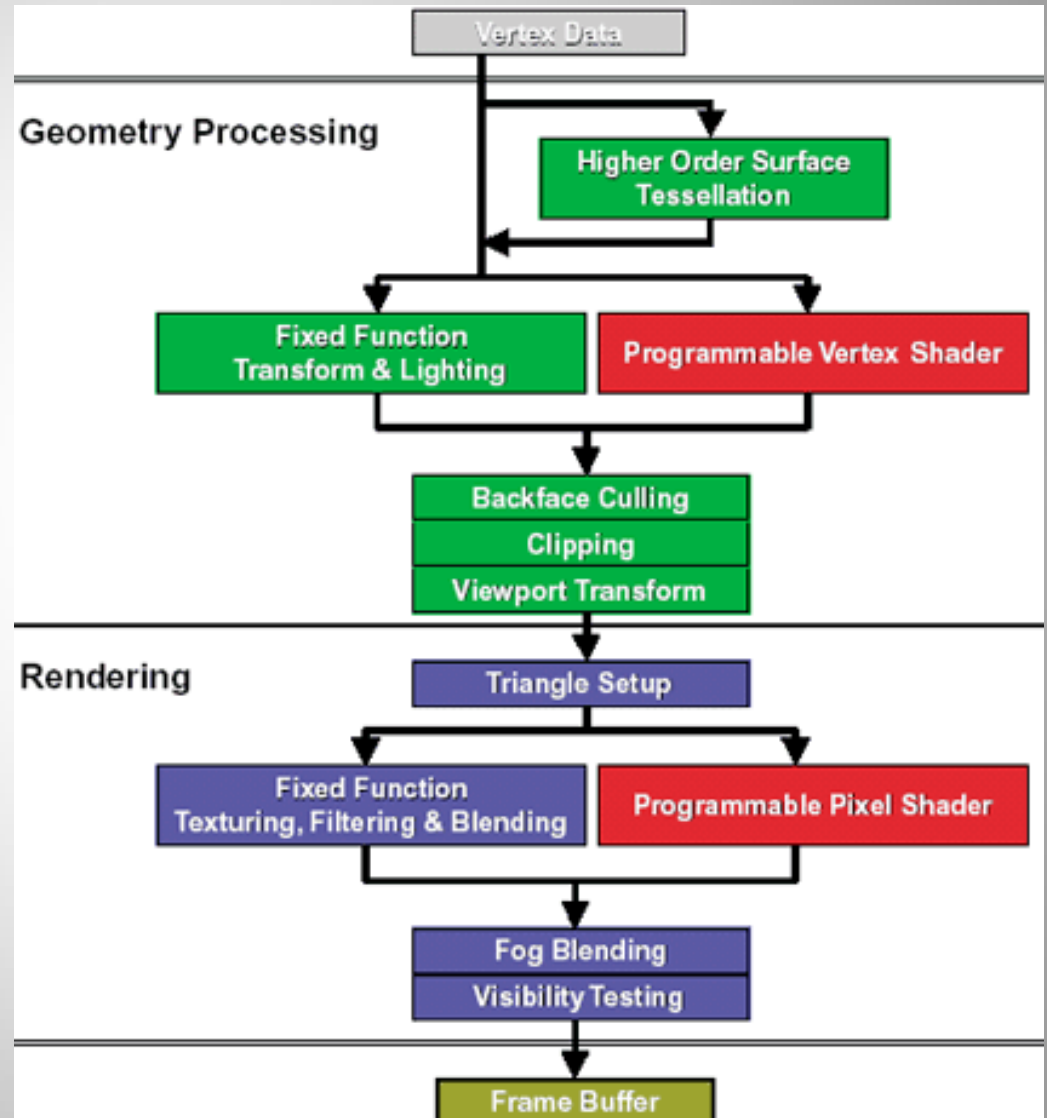
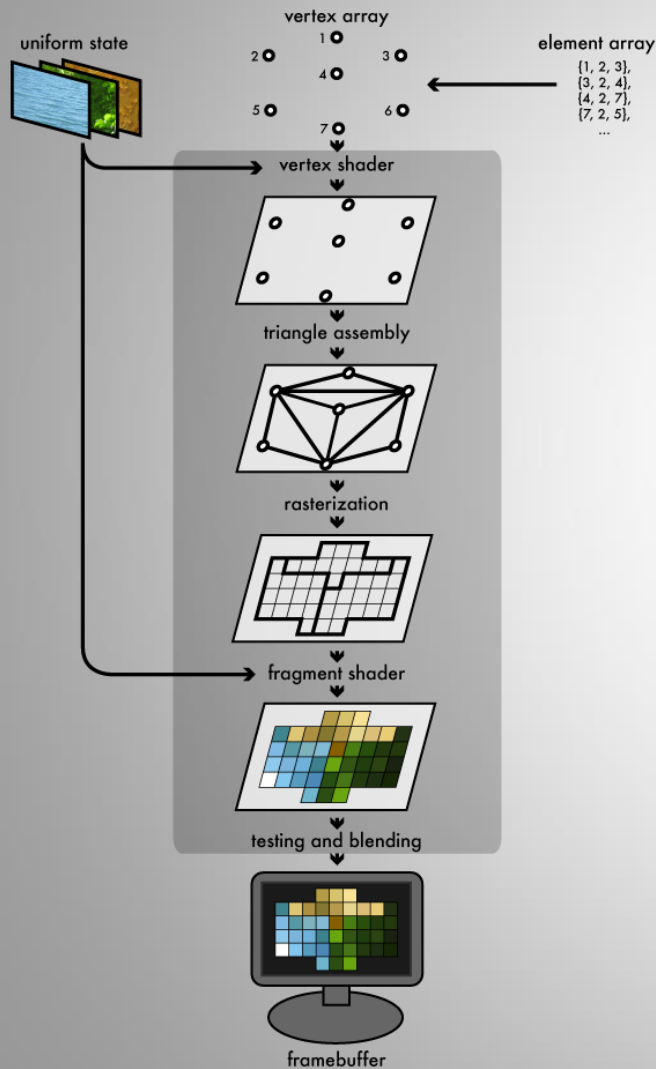
- Objects that are partially within the viewing volume need to be clipped – just like the 2D case



# WebGL

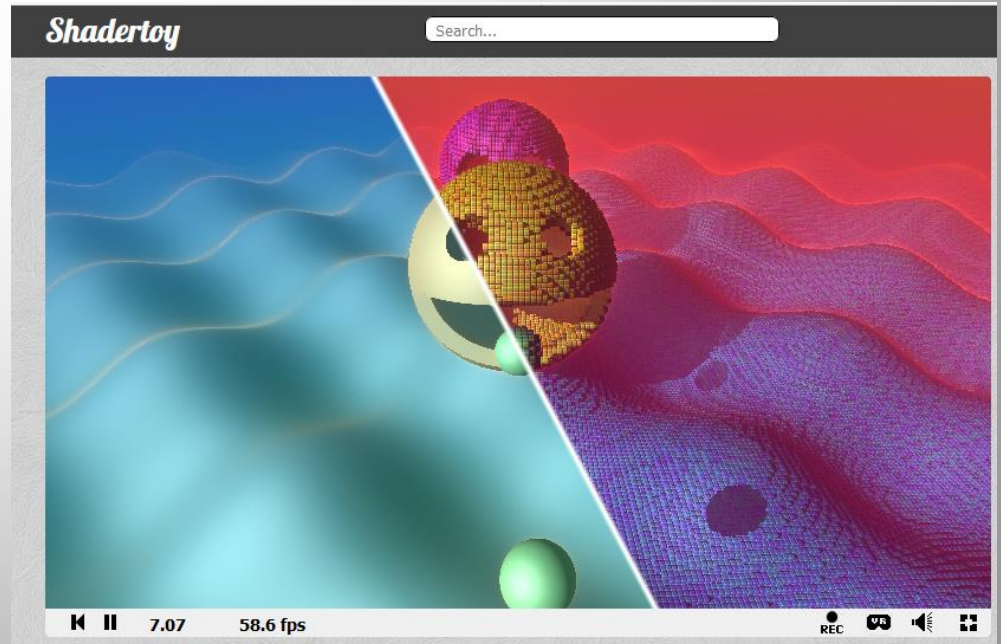
- **Low-level API**
  - GLSL OpenGL Shading Language
  - Machine d'état: OpenGL Context
  - Calcul de matrices et transformations
  - Buffers de vertex: positions, normals, color, texture
  - Depth buffer, Blending, transparency
  - Lighting, Cameras...
  - [https://developer.mozilla.org/fr/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/fr/docs/Web/API/WebGL_API)
  - <https://webglfundamentals.org/webgl/lessons/fr/>

# WebGL Pipeline



# WebGL Examples

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

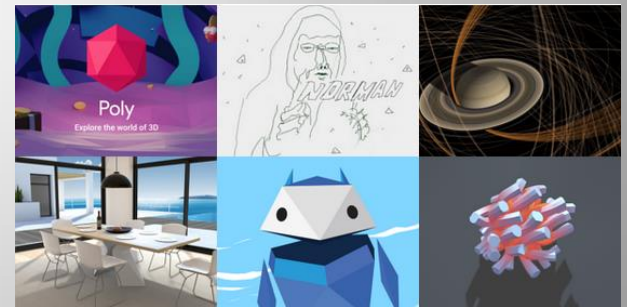


# Three.js

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



- <https://davidlyons.dev/threejs-intro/>



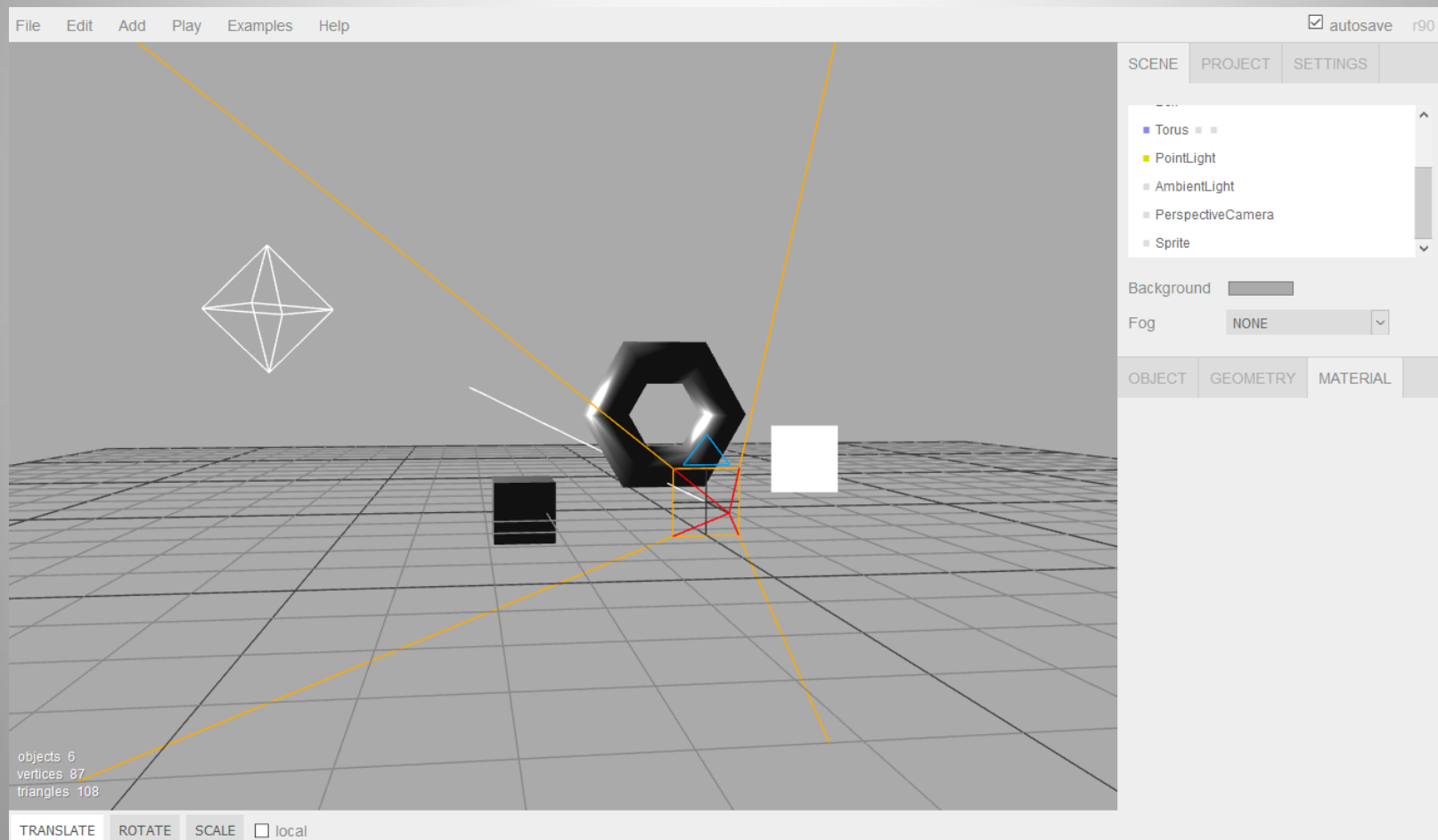
# Fonctionnalités

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

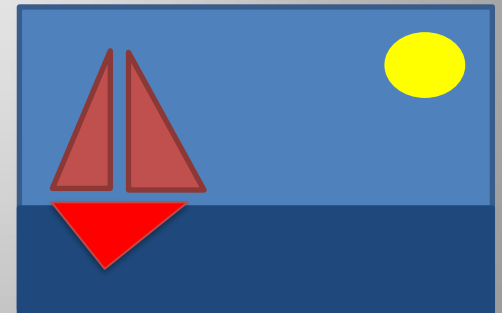
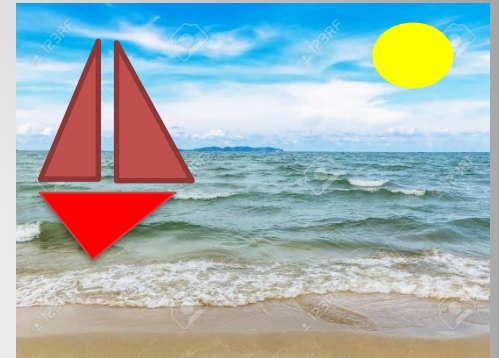
# Threejs Editor

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



# Exercice 1 (45min)

- Dessiner dans un canvas et un svg (1 pages ou 2 pages séparées)
  - Canvas:
    - Choisir et afficher une image avec ciel, mer, bateaux
    - Dessiner bateaux + soleil: Rectangle + triangle + ronds
  - SVG:
    - Dessiner ciel, mer, bateau et soleil
    - Comportement: changement de couleur (clic ou survol)



# 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/>

# Exercice 2 – Three.js

- **Exercice 2 (2h) :**
  - 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 et/ou DeviceMotion
  - Ajoutez Fog/pluie ou particules
- **Exo3 (bonus): mettre un contexte:**  
compas/gyro, système solaire, animation....