



**eleven-labs.com**

WebGL - 26/02/2015



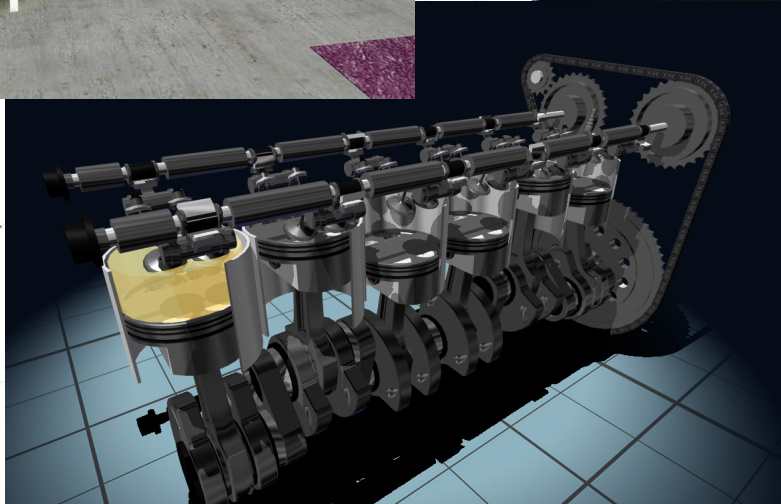
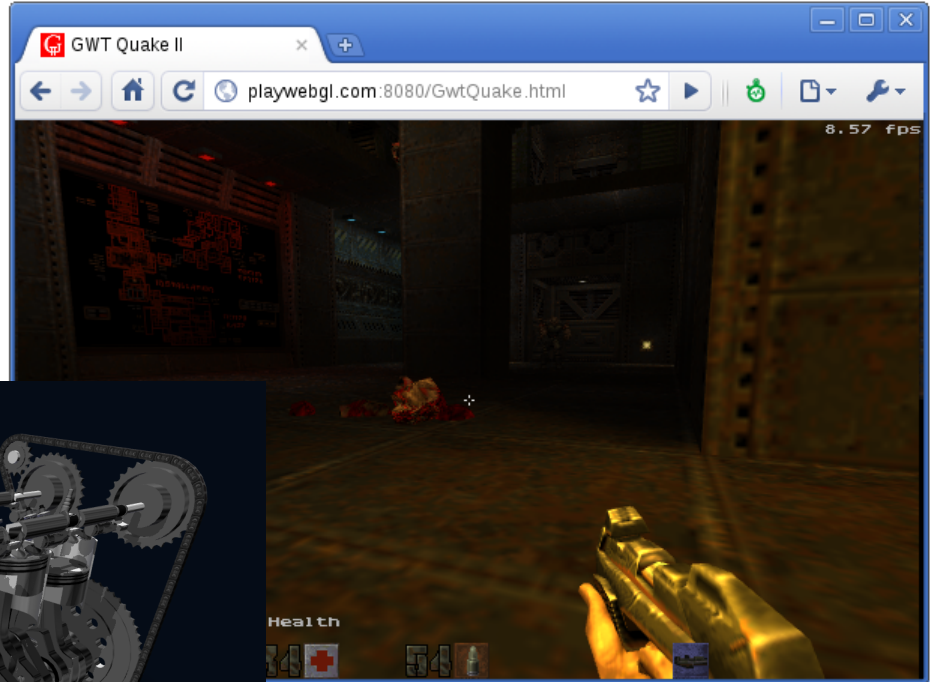
## Qu'est ce que WebGL ?

WebGL est une interface qui permet d'utiliser les standards OpenGL ES pour être utilisée dans une application Web.

## Son fonctionnement

1. WebGL s'appuie sur le langage javascript pour communiquer avec les API OpenGL.
2. Les instructions sont reçues par le pilote de la carte graphique et le rendu est effectué par cette dernière.
3. Le résultat du traitement est retourné au navigateur puis affiché.

Qu'est-ce qu'on fait ?





## Comment faire ?

Plusieurs librairies existent :

- Three.js
- BabylonJS
- Pixi.js

WebGL



Eleven Labs

On se lance ?

Google Street View **fait maison**



BabylonJS



*Eleven labs*

# BabylonJS

**David Rousset et David Catuhe** développeurs chez Microsoft

v1 mai 2013

v2 depuis février 2015

Licence Apache 2.0



BabylonJS



Eleven labs

## Fonctionnalités

HandJs

Gestion des collisions

Moteur physique

Oculus Rift, Gamepad camera, Touch camera, Follow camera

Import des scènes créés sous Blender, 3DS Max

IE11

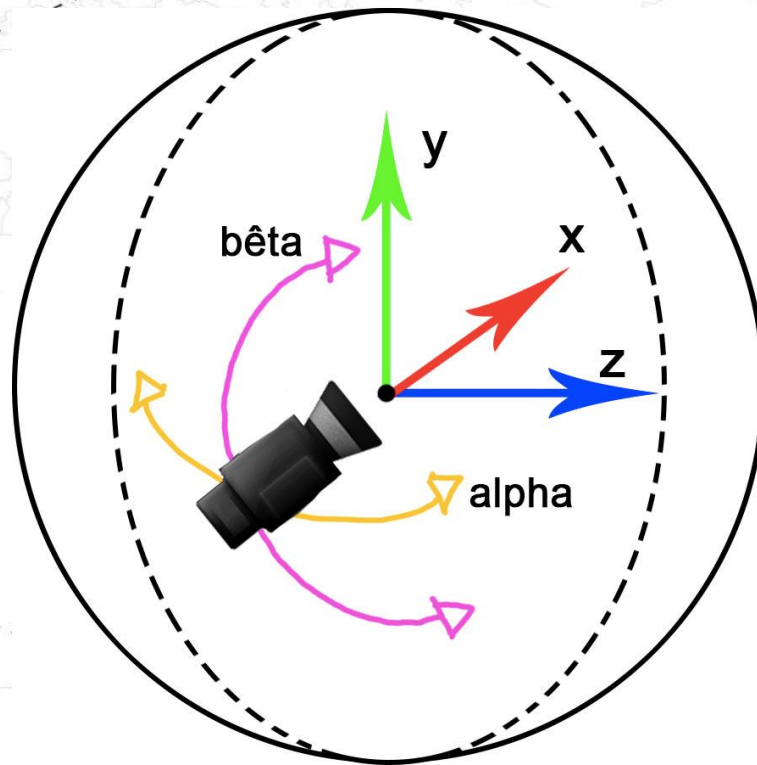




## Pourquoi ?

- Fonction photosphère depuis Android KitKat
- Passion pour la photo
- Privacy compliant

# La caméra





## La scène

```
global.canvas = document.getElementById("renderCanvas");  
global.engine = new BABYLON.Engine(global.canvas, true);  
global.scene = new BABYLON.Scene(global.engine);
```



## La caméra

```
camera = new BABYLON.ArcRotateCamera(  
    "ArcRotateCamera",  
    0,  
    Math.PI/2,  
    10,  
    new BABYLON.Vector3(0, 0, 0),  
    global.scene  
);  
camera.attachControl(global.canvas, false);
```



## Une sphère

```
var sphere = BABYLON.Mesh.CreateSphere("sphere1", 32, 100, global.scene);

var texture = new BABYLON.Texture("photos/" + current + ".jpg", global.scene);
texture.wAng = Math.PI/-2;

var materialPhotoSphere = new BABYLON.StandardMaterial(
    "texturePhotoSphere",
    global.scene
);
materialPhotoSphere.diffuseTexture = texture;

sphere.material = materialPhotoSphere;
```



Un cône

```
var moveBlock = BABYLON.Mesh.CreateCylinder()
```



## Le rendu

```
global.engine.runRenderLoop(function () {  
  global.scene.render();  
});
```



WebGL



Eleven Labs

Démo

<http://127.0.0.1/photosphere>



## Ressources

<https://github.com/lepiaf/photosphere>

<https://github.com/topheman/BabylonJSFirstSteps>

<http://www.simplicity.be/a-vr-photo-sphere-in-webgl/>

<http://www.babylonjs.com/>

<http://www.sitepoint.com/three-js-babylon-js-comparison-webgl-frameworks/>

<https://github.com/BabylonJS/Babylon.js#features>

<http://www.ekino.com/interview-david-rousset-babylonjs/>