Module ReV - mondes virtuels

Eric Maisel

2016

Réalité Virtuelle

Faire vivre une expérience sensorielle

- Simulation informatique
- Interactive
- Immersive
- Environnements réels ou imaginaires

Immersion

Etat Psychologique

- Le sujet ne se rend plus compte de son propre état physique
  - Il est ailleurs
  - Il n'est plus lui
- Modification du temps, de l'espace

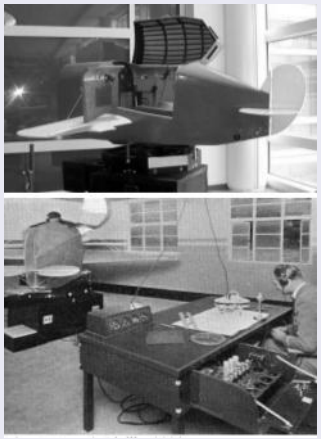
Interfaces comportementales

- Interfaces sensorielles
- Interfaces motrices

Boucle sensori-motrice

Histoire

- Simulateur de vol (E-Link, 1929)



- Sensorama (Helig 1956)

Histoire

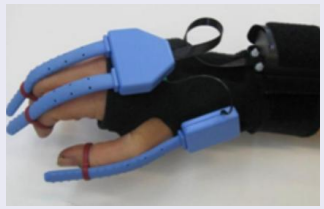
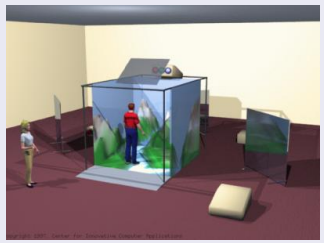
- 1960 : la souris
- 1965-67 : les casques 
- années 80 : les gants de données 
- 1980 : les CAVE 

Plate-formes

Machines dédiées

- Simulateurs de vol

Machines puissantes + environnements logiciels dédiés

Environnements banalisés

- Ordinateur de bureau
- Lunettes "low-cost" + "Leap-motion - like"
- Navigateur

Des instructions aux balises

Afficher un cube

Approche impérative, procédurale

WebGL

- env. 300 lignes
- instructions manipulant des buffers, des tableaux, des registres

Three.js

- env. 30 lignes
- instructions manipulant des vecteurs, des matrices, des textures, des polygones

Approche déclarative : GLAM, A-Frame (https://aframe.io)

- 3 lignes
- description d'objets, de relations entre objets

WebGwhatnow?

1. Create Buffers

```
var vertexBuffer;

vertexBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, vertexBuffer);

var verts = [
        // Front face
        -1.0, -1.0,  1.0,
         1.0, -1.0,  1.0,
         1.0,  1.0,  1.0,
        -1.0,  1.0,  1.0,

        // Back face
        -1.0, -1.0, -1.0,
        -1.0,  1.0, -1.0,
         1.0,  1.0, -1.0,
         1.0, -1.0, -1.0
        ...
];

gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(verts), gl.STATIC_DRAW);
```

2. Define Vertex and Fragment Shaders

```
var vertexShaderSource =
    "    attribute vec3 vertexPos;\n" +
    "    attribute vec2 texCoord;\n" +
    "    uniform mat4 modelViewMatrix;\n" +
    "    uniform mat4 projectionMatrix;\n" +
    "    varying vec2 vTexCoord;\n" +
    "    void main(void) {\n" +
    "        gl_Position = projectionMatrix * modelViewMatrix * \n" +
    "            vec4(vertexPos, 1.0);\n" +
    "        //Output the texture coordinate in vTexCoord\n" +
    "        vTexCoord = texCoord;\n" +
    "    }\n";

var fragmentShaderSource =
    "    precision mediump float;\n" +
    "    varying vec2 vTexCoord;\n" +
    "    uniform sampler2D uSampler;\n" +
    "    void main(void) {\n" +
    "        // Return the pixel color: always output white\n" +
    "    gl_FragColor = texture2D(uSampler, vec2(vTexCoord.s...);\n" +
"}\n";
```

3. Draw The Cube

```
function draw(gl, obj) {

    // clear the background (with black)
    gl.clearColor(0.0, 0.0, 0.0, 1.0);
    gl.enable(gl.DEPTH_TEST);
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
    // set the shader to use
    gl.useProgram(shaderProgram);

    // connect up the shader parameters: position, tex coord
    // projection/model matrices and texture
    // set up the buffers
    gl.bindBuffer(gl.ARRAY_BUFFER, obj.buffer);
    gl.vertexAttribPointer(shaderVertexPositionAttribute, obj.vertSize, gl.FLOAT, false
    gl.bindBuffer(gl.ARRAY_BUFFER, obj.texCoordBuffer);
    gl.vertexAttribPointer(shaderTexCoordAttribute, obj.texCoordSize, gl.FLOAT, false
    gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, obj.indices);
    gl.uniformMatrix4fv(shaderProjectionMatrixUniform, false, projectionMatrix);
    gl.uniformMatrix4fv(shaderModelViewMatrixUniform, false, modelViewMatrix);

    gl.activeTexture(gl.TEXTURE0);
    gl.bindTexture(gl.TEXTURE_2D, webGLTexture);
    gl.uniform1i(shaderSamplerUniform, 0);

    // draw the object
    gl.drawElements(obj.primtype, obj.nIndices, gl.UNSIGNED_SHORT, 0);
}
```

300 Lines of JavaScript. Uniforms and Shady Things. NOT GROOVY.

http://www.tonyparisi.com

10/9/201

1. Create Renderer. Create Scene. Add Texture Map. Draw Cube. Done.

```
renderer = new THREE.WebGLRenderer( { canvas: canvas, antialias: true } );

scene = new THREE.Scene();

camera = new THREE.PerspectiveCamera( 45, canvas.width /
  canvas.height, 1, 4000 );
scene.add(camera);

var light = new THREE.DirectionalLight( 0xffffff, 1.5);
scene.add( light );

var mapUrl ="../images/webgl-logo-256.jpg";
var map = THREE.ImageUtils.loadTexture(mapUrl );
var material = new THREE.MeshPhongMaterial({ map: map });

var geometry = new THREE.CubeGeometry(2, 2, 2);
cube = new THREE.Mesh(geometry, material);
scene.add( cube );
```

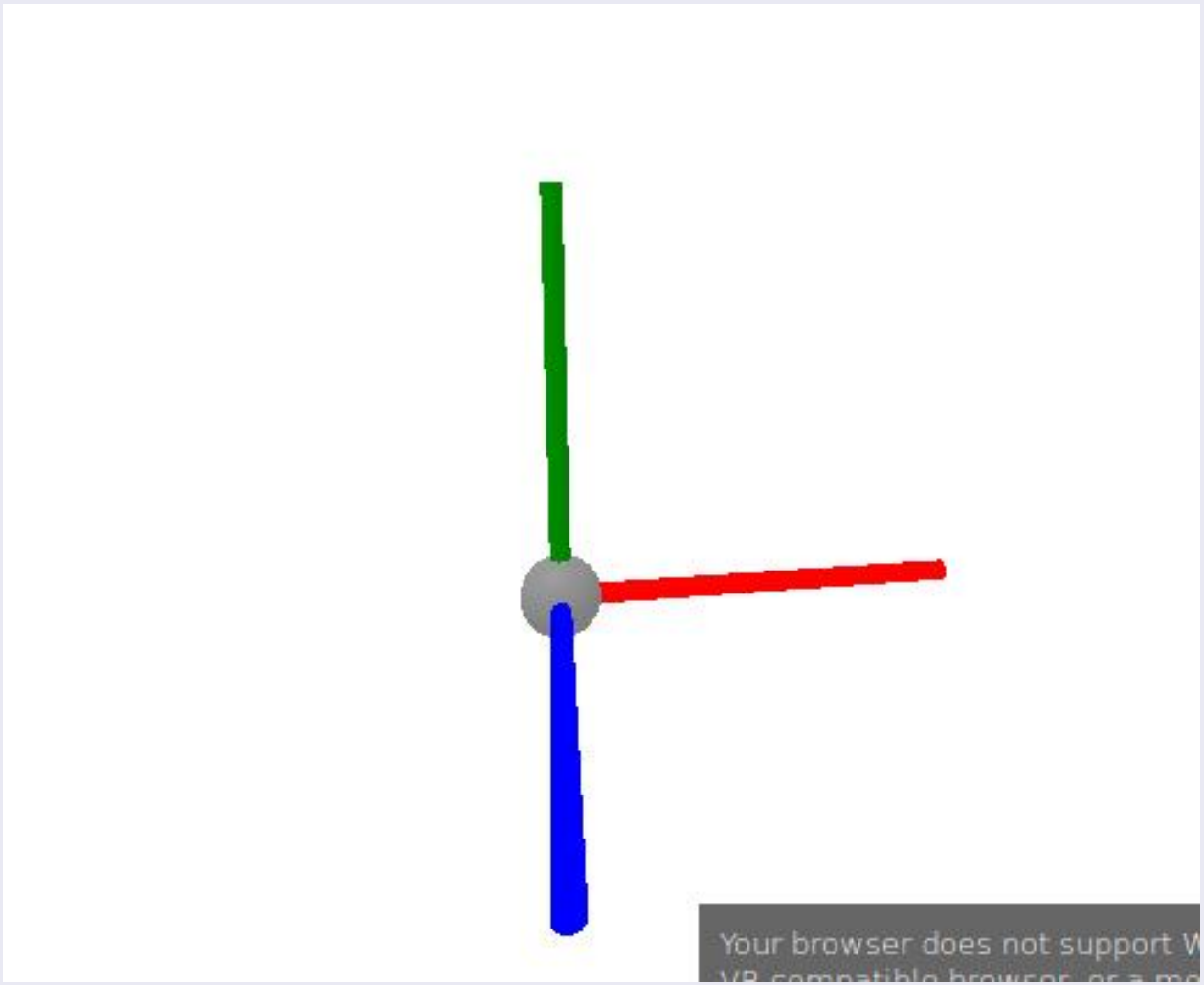
40 Lines of JavaScript.
That's Cool.

Un monde = une page Web

Un monde embarqué dans une page Web

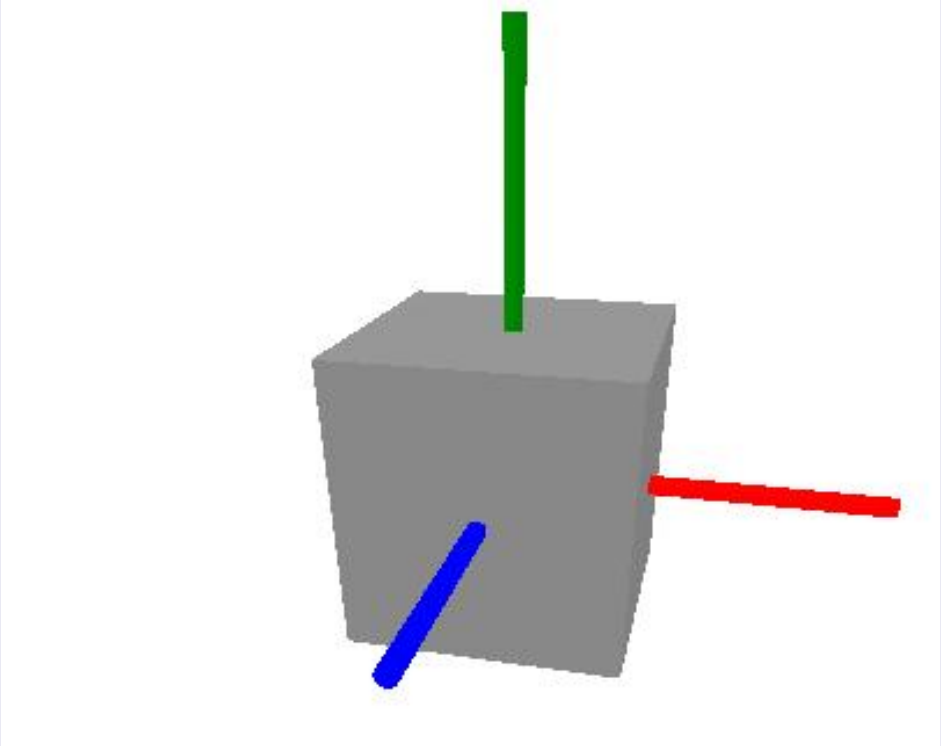
```
<html>
  <head>
    ...
    <script src="a-frame.min.js"> </script>
    ...
  </head>

  <body>
    ...
    <a-scene> ... </a-scene>

    <script> ... </script>
    ...
  </body>
</html>
```

Un espace à remplir



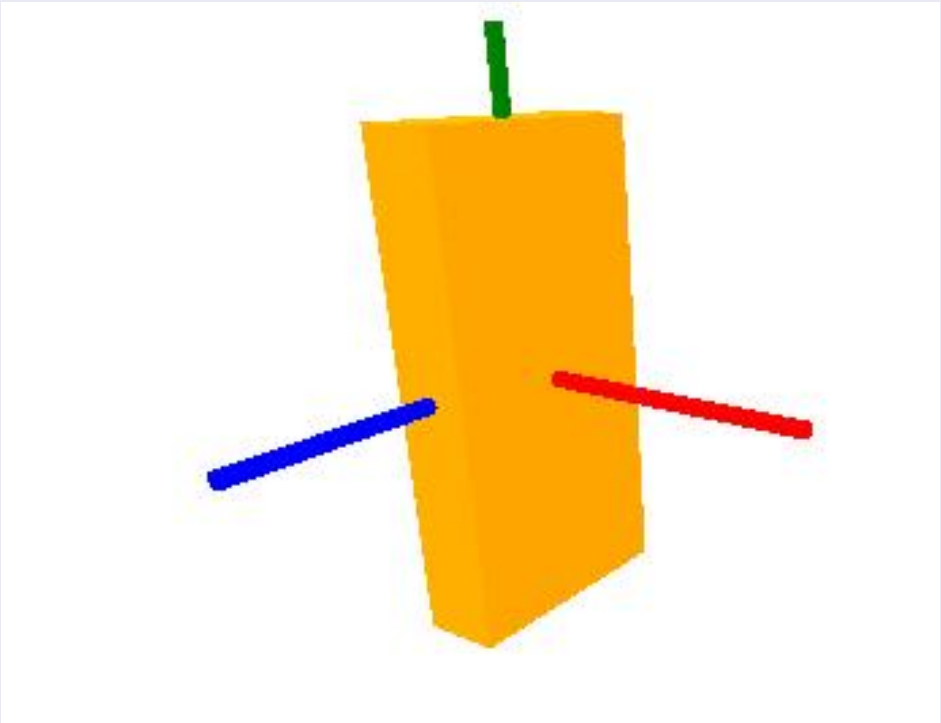
Un cube

```
<a-scene>
  <a-cube> </a-cube>
</a-scene>
```



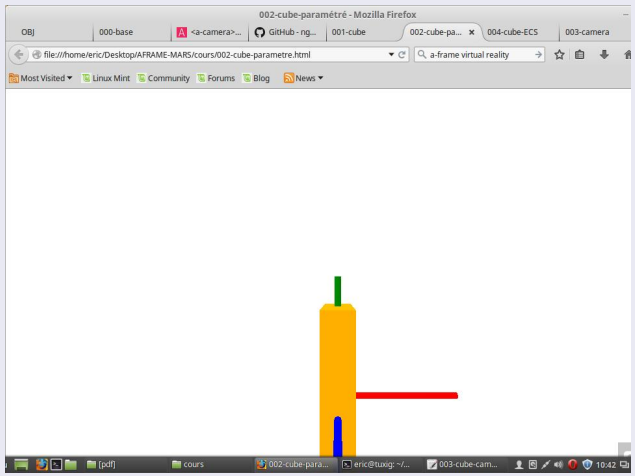
Un cube paramétré

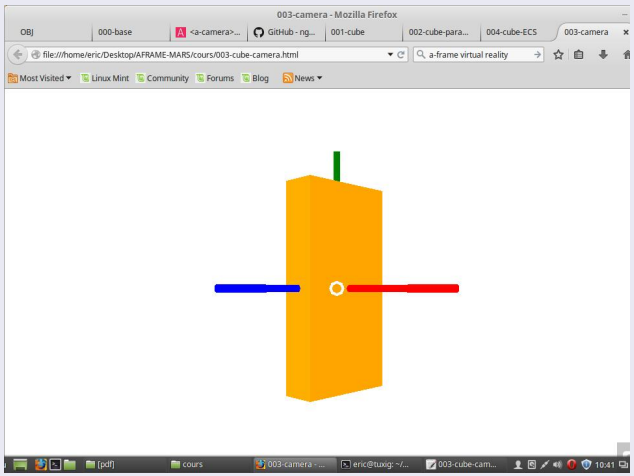
```
<a-scene>
  <a-cube>
    width="0.5" depth="1.5" height="3"
    color="orange"
  </a-cube>
</a-scene>
```



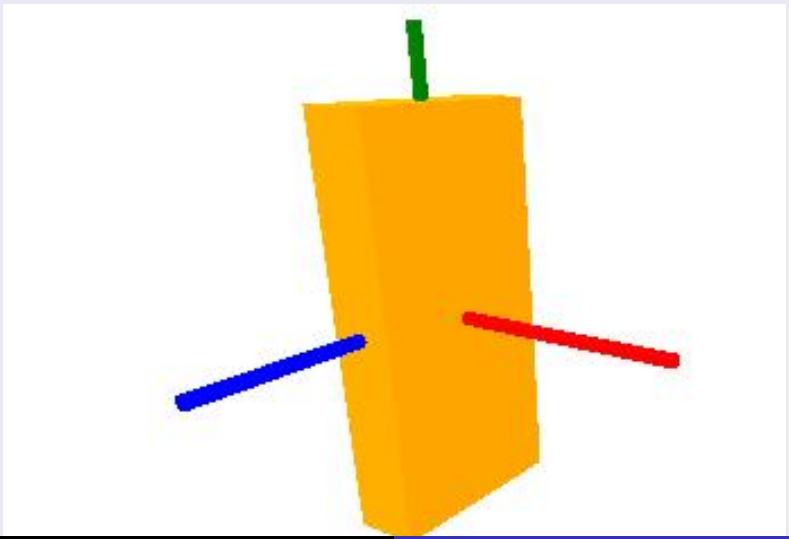
Une caméra

```
<a-scene>
  <a-camera
    fov = "80.0" near = "0.01" far = "1000.0"
    rotation = "0 45 0"
    position = "5 0 5">
  </a-camera>

  <a-cube> ... </a-cube>
</a-scene>
```





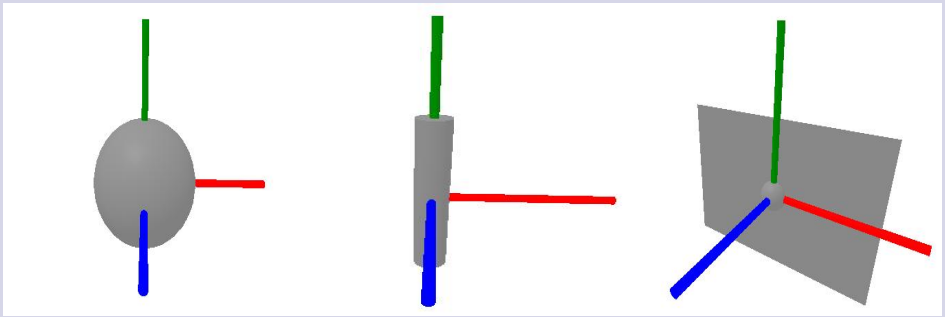
```
<a-scene>
  <a-entity>
    id       = "boite-00"
    geometry = "primitive: box;
                width: 0.5; depth: 1.5; height: 3;"
    material = "color: orange;"
  </a-entity>
</a-scene>
```



- **Entity** : objet générique + id
- **Component** : ensemble de propriétés qui peuvent modifier
  - la pose
  - l'aspect
  - le comportement
  - les fonctionnalités
  - ...

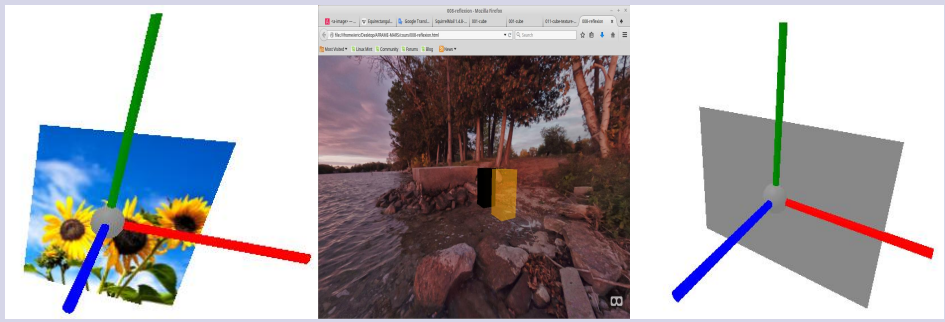
D'autres primitives (1)

```
<a-sphere radius="2.0"></a-sphere>
<a-cylinder height="2.0" radius="0.2"></a-cylinder>
<a-plane height="2.0" width="3.0"></a-plane>
```



D'autres primitives (2)

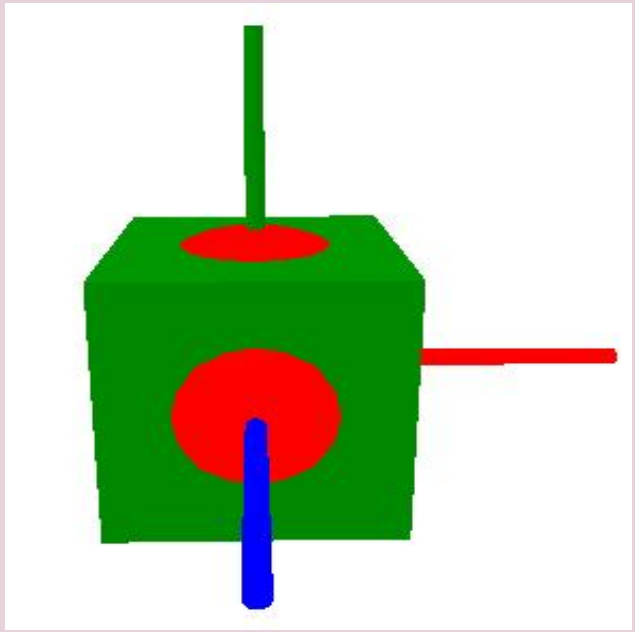
```
<a-image width="2.0" height="2.0"
          src="images/flowers.jpg"></a-image>
<a-sky src="skies/lake.jpg"></a-sky>
<a-video height="2.0" width="3.0"></a-plane>
<a-video src="penguin.mp4"
         width="16" height="9"></a-video>
```



Placement des objets (1)

Primitives : objets canoniques

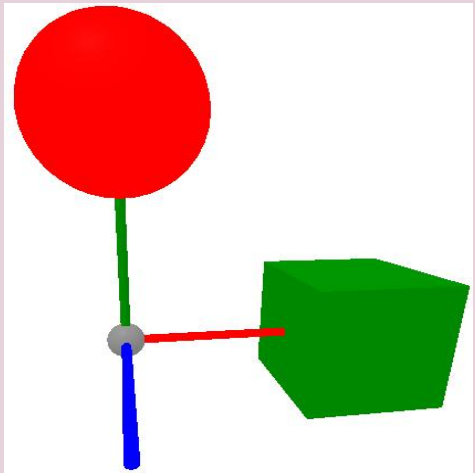
```
<a-scene>
  <a-cube color="green"></a-cube>
  <a-sphere color="red"></a-sphere>
</a-scene>
```



Placement des objets (2)

Placement des objets

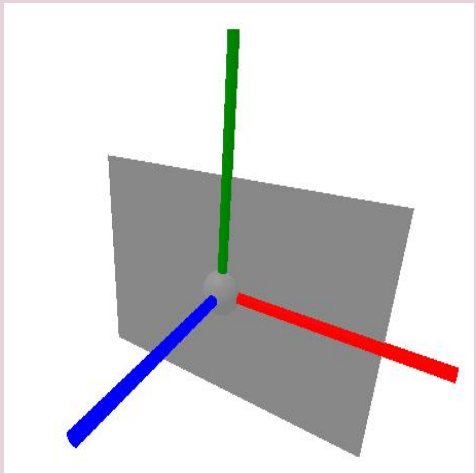
```
<a-scene>
  <a-cube   color="green"
            position="2.5 0 0"></a-cube>
  <a-sphere color="red"
            position="0 2.5 0"></a-sphere>
</a-scene>
```

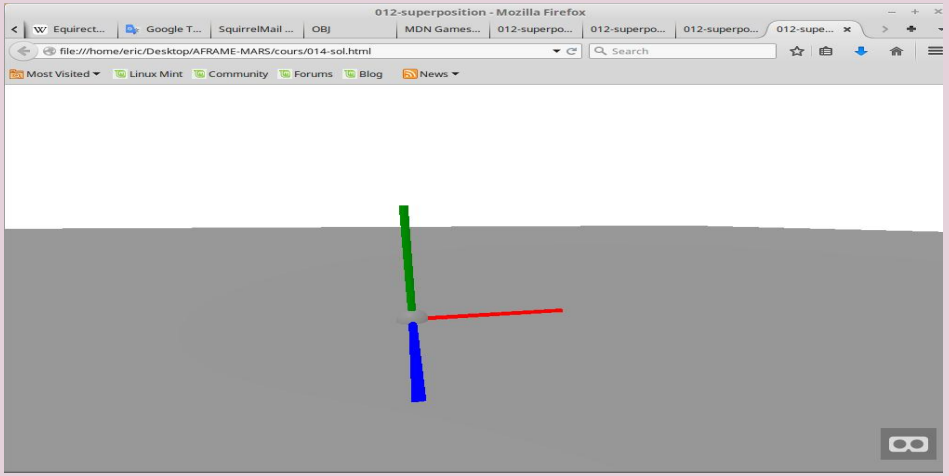


Placement des objets (3)

Créer un sol

```
<a-scene>
  <a-plane width="100" height="100"
           rotation="90 0 0"
  ></a-plane>
</a-scene>
```





Placement des objets (4)

Graphe de scène

Description d'un pilier

```
<a-assets>
  <a-mixin id="pilier"
           geometry="primitive: box;
                     width: 1; depth: 1; height: 3"
  ></a-mixin>
</a-assets>
```

Placement des objets (5)

Graphe de scène

Description de la porte canonique

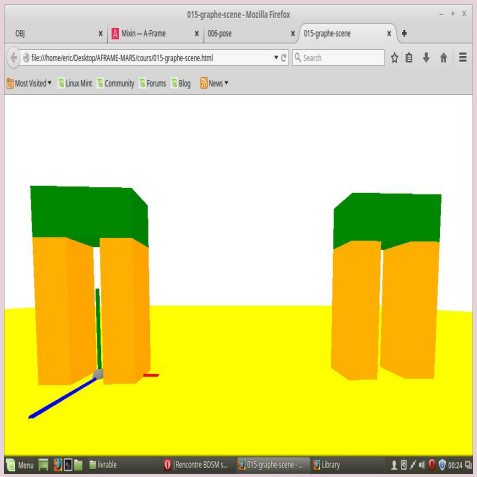
```
<a-scene>
  <a-entity mixin="pilier" material="color: orange"
            position="-1.0 1.5 0.0"
  ></a-entity>

  <a-entity mixin="pilier" material="color: orange"
            position="1.0 1.5 0.0"
  ></a-entity>

 <a-entity mixin="pilier" material="color: green"
        rotation="0 0 90" position="0.0 3.5 0.0"
  ></a-entity>
</a-scene>
```

Placement des objets (6)

Graphe de scène

Description de la 2^{eme} porte

```
<a-entity position="10 0 0">
  <!-- ICI le code des 3 mixins -->
</a-entity>
```



Face animable

Proposez un modèle représentant un visage. On y trouvera les éléments suivants :

- La face
- des yeux (animables)
- un nez
- une bouche animable

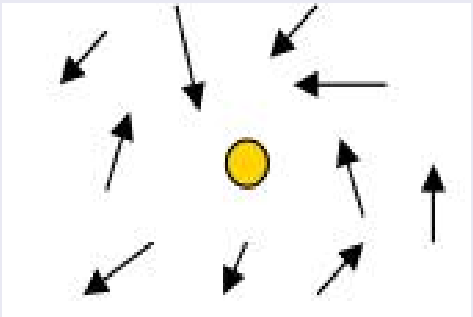
Aspect des objets
Eclairement

X

Aspect des objets
Lumière

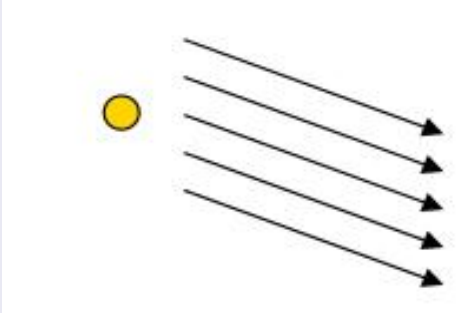
- Rayonnement électro-magnétique
- Spectre lumineux
- Echantillonnage sur 3 longueurs d'onde

Lumière ambiante



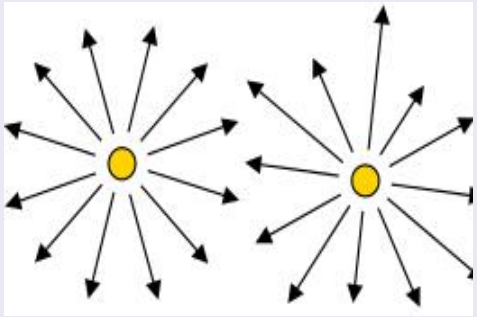
```
<a-entity light="type: ambient; color=#CCC"
></a-entity>
```

Source directionnelle



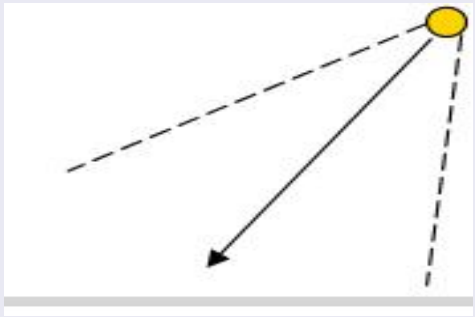
```
<a-entity light="type: directional;
                 color: #CCC; intensity: 0.5"
          position="-1 1 0"
></a-entity>
```

Source ponctuelle



```
<a-entity light="type: point;
                 intensity: 0.5 ;
                 distance: 10; decay: 2"
          position="5 3 0"
></a-entity>
```

Source spot



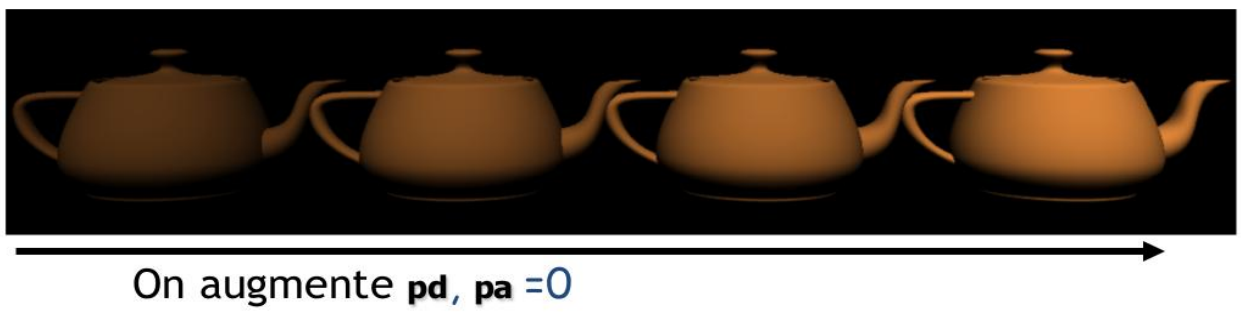
```
<a-entity light="type: spot;
                 intensity: 1.0 ;
                 distance: 10; decay: 1;
                 angle:60 ; exponent:10.0;"

          position="5 3 0"
></a-entity>
```

$$I_p(\lambda) = p_a I_a(\lambda)$$

Aspect des objets
Réflexion diffuse (modèle de Lambert)

$$I_p(\lambda) = p_d(\lambda)\cos(\theta)I_s(\lambda)$$

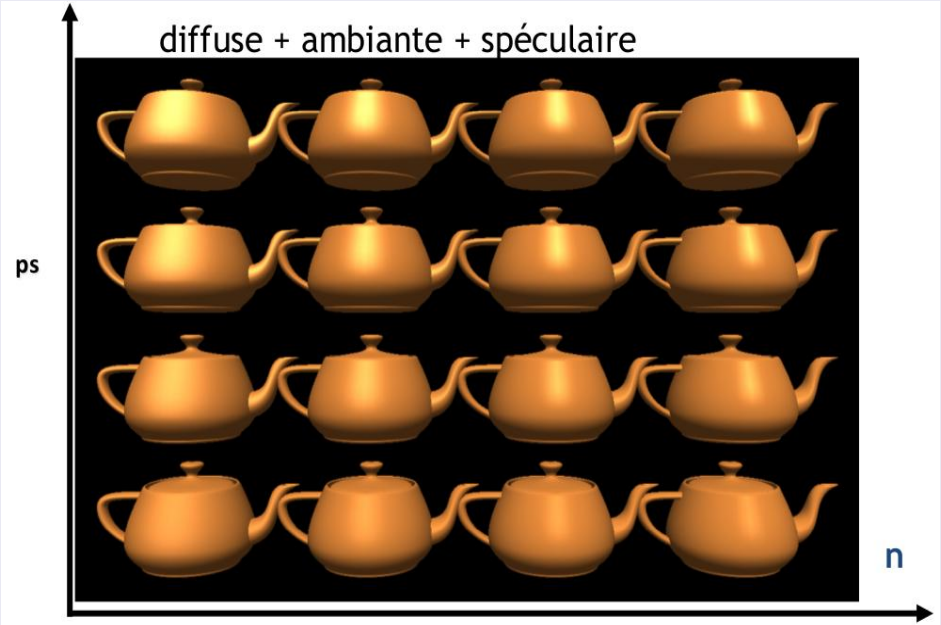
```
<a-entity ...
  material="color: #FF0000 ; roughness: 0.75"
></a-entity>
```



Aspect des objets
Réflexion spéculaire (modèle de Phong)

$$I_p(\lambda) = p_s \cos^n(\theta') I_s(\lambda)$$

```
<a-entity ...
  material="color: #FF0000 ; metalness: 0.75"
></a-entity>
```



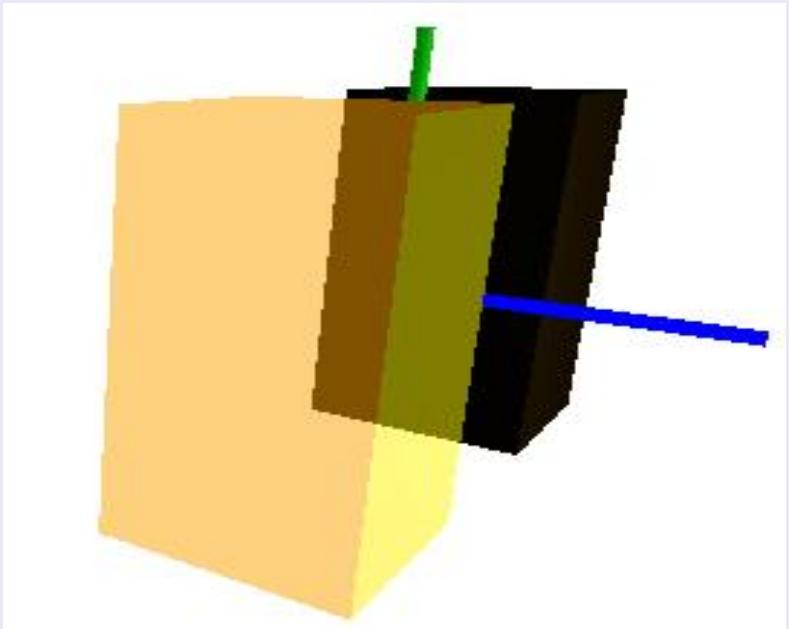
Aspect des objets

Modèle additif (sources, fréquences, modèles d'éclairement)

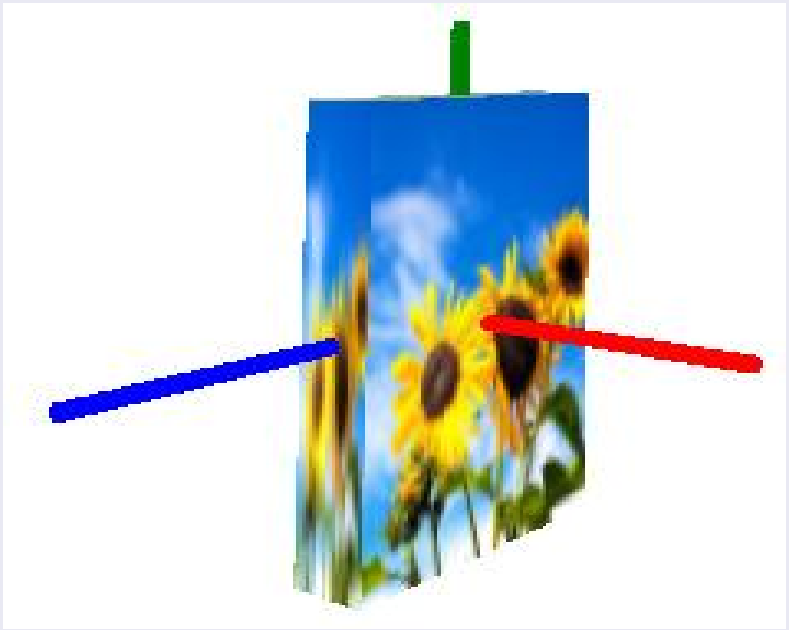
$$I_p(\lambda) = p_a I_a(\lambda) + p_d(\lambda) cos(\theta) I_s(\lambda) + p_s \cos^n(\theta') I_s(\lambda)$$

Combinaison d'un objet et du fond

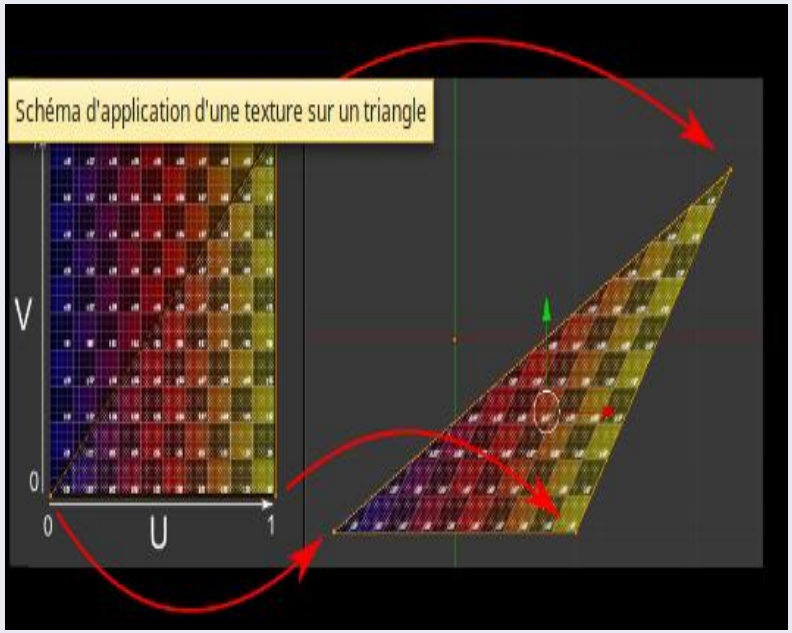
```
material = "color: orange;
            metalness: 0.0 ;
            transparent: true ; opacity: 0.5; "
```



Aspect des objets
Textures (1)



```
<a-entity
   geometry = "primitive: box "
   material = "src: url(images/flowers.jpg); "
 ></a-entity>
```



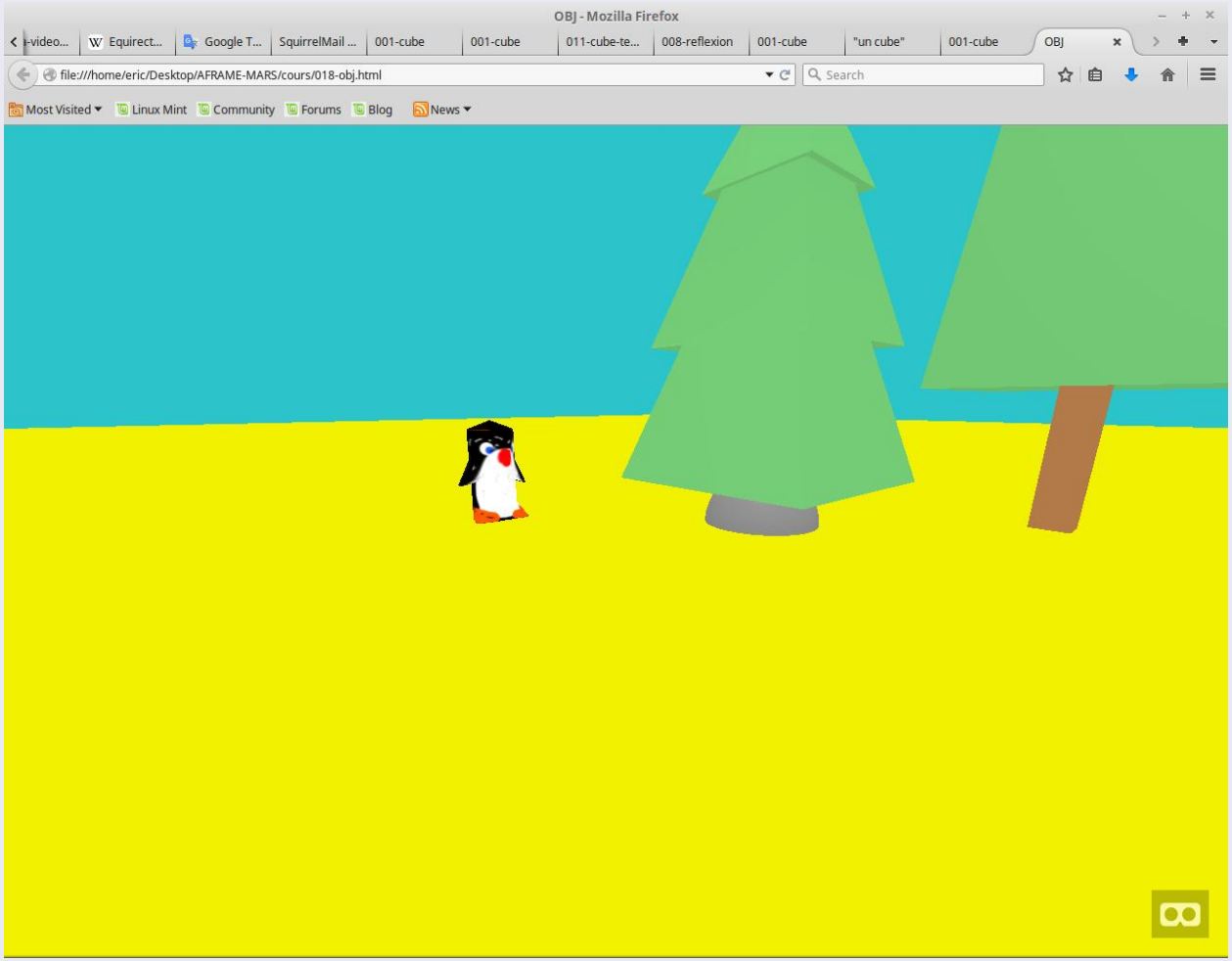
- Coordonnées 3d $\rightarrow$ Coordonnées texels

Maillages (1)

Maillages (2)

```
<head>
  ...
  <script src="../dist/aframe-obj-loader-component.js"
  ></script>
  ...
</head>
  ...
  <a-scene>
    ...
    <a-entity obj-loader="src:url(./obj/tree.obj);
                          mtl:url(./obj/tree2.mtl);"
    ></a-entity>
    ...
  </a-scene>
  ...
```

Maillages (3)


OBJ - Mozilla Firefox
file:///home/eric/Desktop/AFRAME-MARS/cours/018-obj.html
Most Visited
Linux Mint
Community
Forums
Blog
News

Animation

- Vaste sujet
- A-Frame :
  - Balise d'animation
  - Utilisation de JavaScript

Interaction

- Vaste sujet
- A-Frame :
  - Balise d'animation
  - Utilisation de JavaScript

```
<a-scene>
  <a-camera
    cursor-visible="true" cursor-color="blue"
    position="0 1.8 4"
  ></a-camera>

  <a-entity id="cube"
            geometry = "primitive: box"
            sound    = "src: birds.mp3 ; on: click"
  ></a-entity>
</a-scene>
```