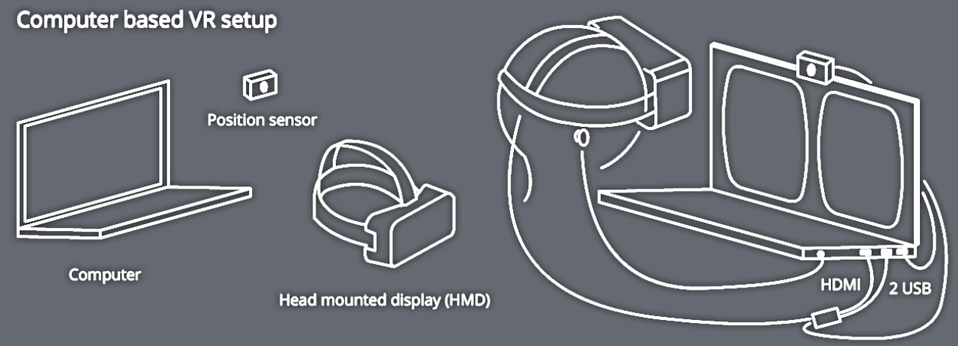
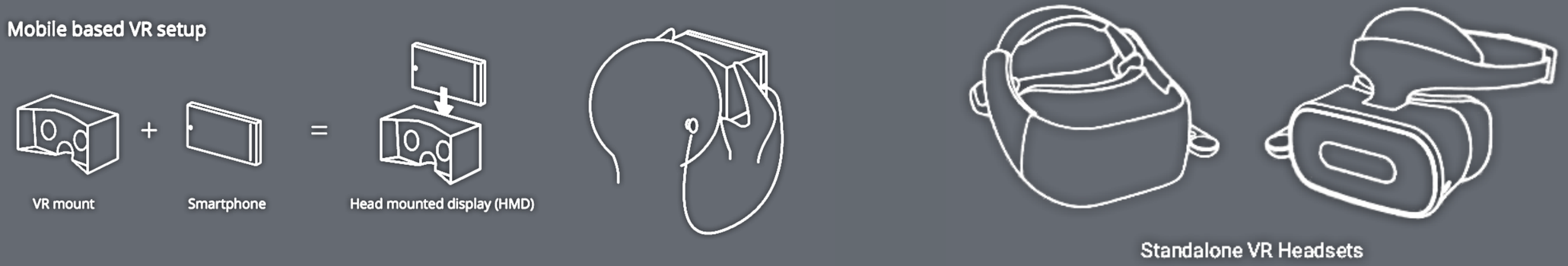
Agenda

- Welcome to A-Frame NYC - Introduction & Definitions
- VR Market - Current Tools and Gadgets
- Retrospective on VR on the Web
- WebGL & WebVR & A-Frame
- Workshop Session
- Close up Session

Introduction to VR on the Web

VR Setup*





* Illustrations from Mozilla

VR Experiences

New considerations for VR app development

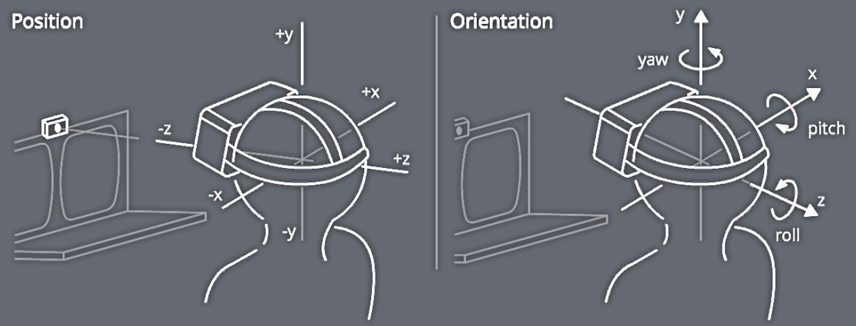
Focus on User Experience

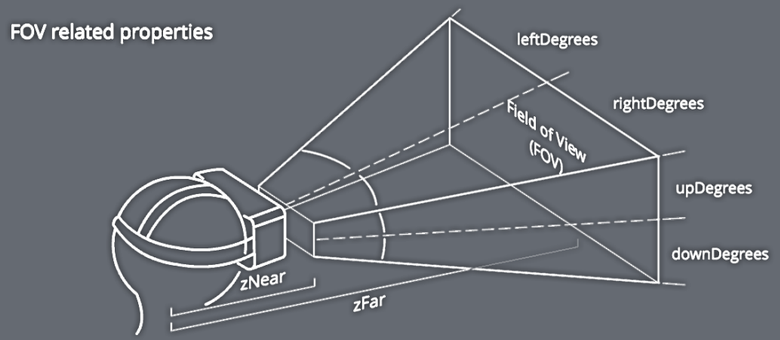
- Stereoscopic vision & interpupillary distance (IPD)
- Head tracking & degrees of freedom (DoF)
- Cone of focus & field of view (FOV)
- 3D positional audio
- Latency & frame rate (frames per second / FPS)

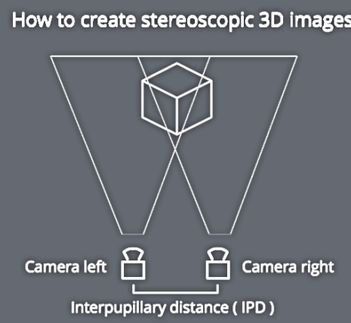
Focus on User Comfort

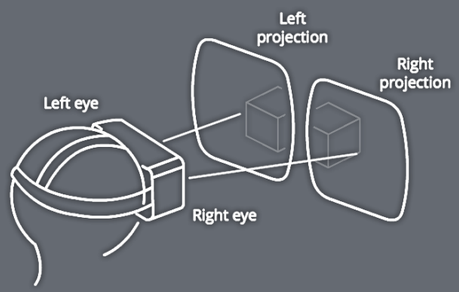
- Eye strain
- Motion sickness

Degrees of Freedom, Stereoscopic Vision & Interpupillary Distance*









* Illustrations from Mozilla

Native VR Devices (PC & Console)

HTC Vive (room-scale VR), Oculus Rift, Playstation VR, Windows MR (Acer), Magic Leap, Fove, Razer OSVR

Mobile VR Devices (Smartphone)

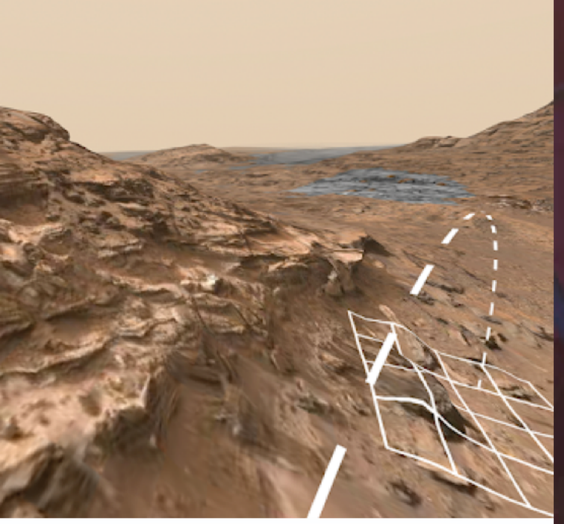
Samsung Gear VR
Daydream
Cardboard
Hundreds of third-party headsets

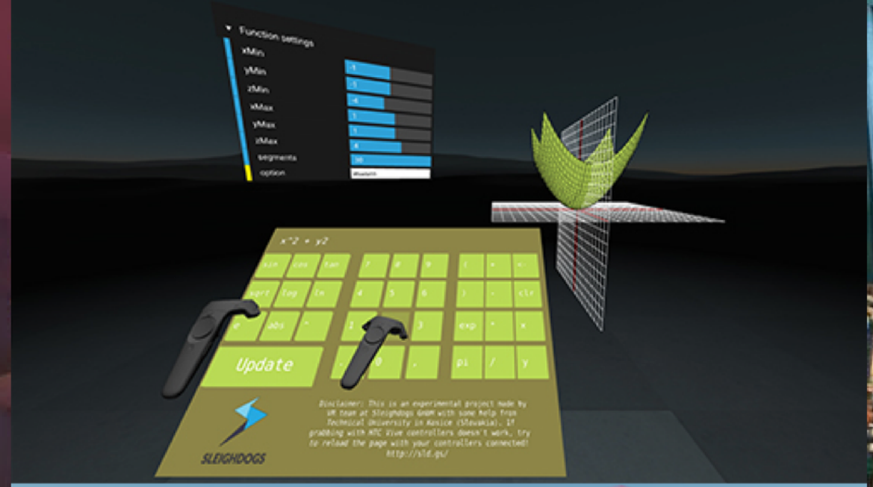
VR Programming

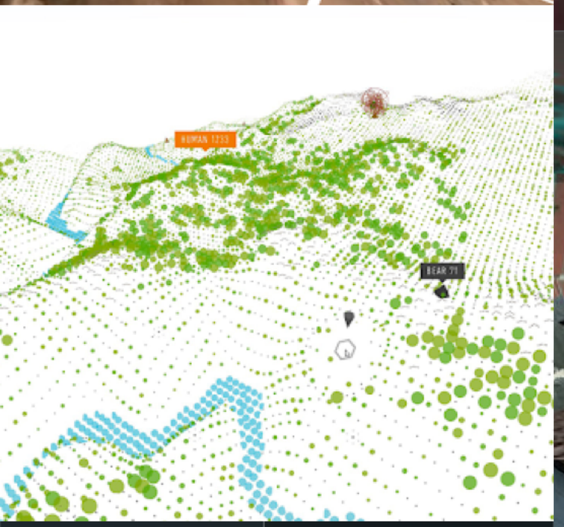
- Unreal Engine (C++)
- Unity (C#)
- Source Engine (C++)
- WebVR (WebXR Device API)

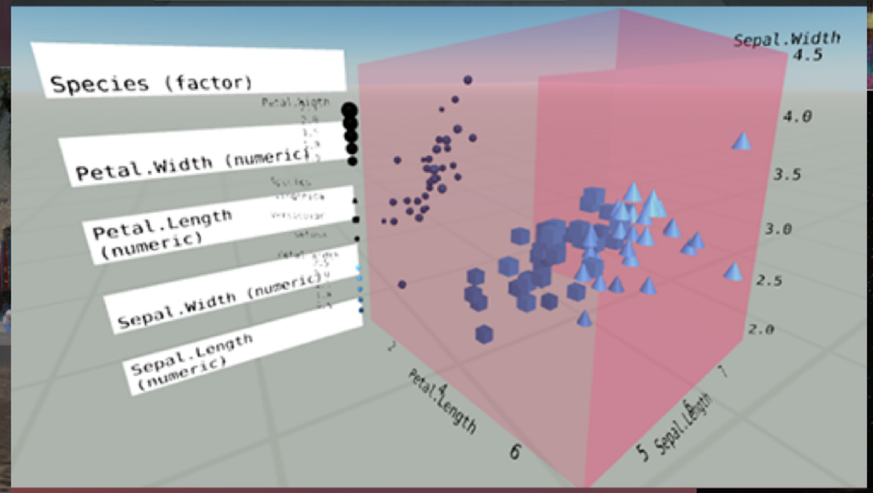
VR Distribution

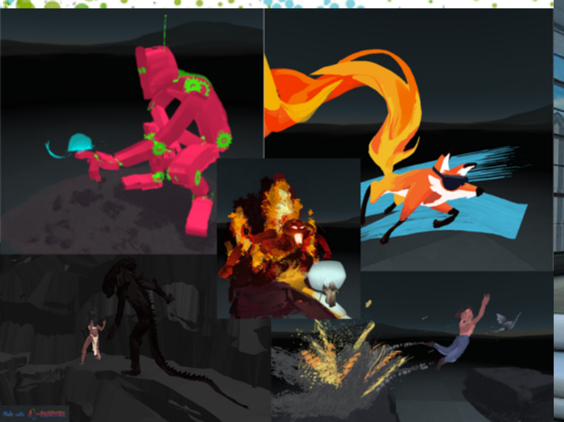
- SteamVR
- Google Play
- Oculus Store
- The Rift Arcade
- V "open alternative" to SteamVR




Function settings
xMin
yMin
zMin
xMax
yMax
zMax
segments
option
x^2 + y2
Update
SLEIGHDOGS

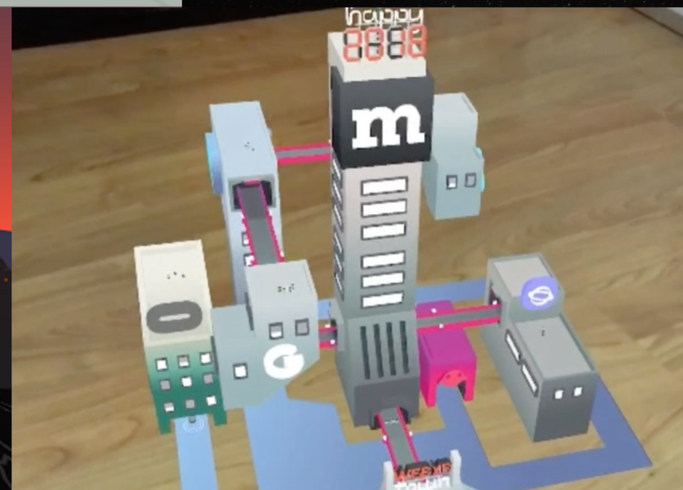



Species (factor)
Petal.Width (numeric)
Petal.Length (numeric)
Sepal.Width (numeric)
Sepal.Length (numeric)
Sepal.Width
4.5
4.0
3.5
3.0
2.5
2.0
Petal.Length
Sepal.Length




AltspaceVR


CBS data
Bevolking
Aantal inwoners
Geslacht
Leeftijdsgroepen
Burgerlijke staat
Allochtonen
Geboorte en sterfte
Particuliere huishoudens
Bevolkingsdichtheid
Own data: EnergyPerformance.csv
OK
1835
1870
7195
5615
2885


m

Why WebVR?

- No app store ecosystem, distribution via internet
- Mobile & desktop automatically supported
- Uses current tools and libraries for JS
- Easy switch between VR & non-VR mode
- Interfacing with hardware through the browser


Progressive Enhancement by Arturo Paracuellos

Virtual Reality on the Web

A Retrospective 1/2

- 1994 VRML - first attempt to create an internet-based 3D language. (Mark Pesce presented the Labyrinth demo he developed with Tony Parisi and Peter Kennard.)
- VRML2 (1997) - added many features (animation) and later was succeeded by X3D
  Problem of VRML: plugin-based technology that only came preinstalled on IE

Virtual Reality on the Web

A Retrospective 2/2

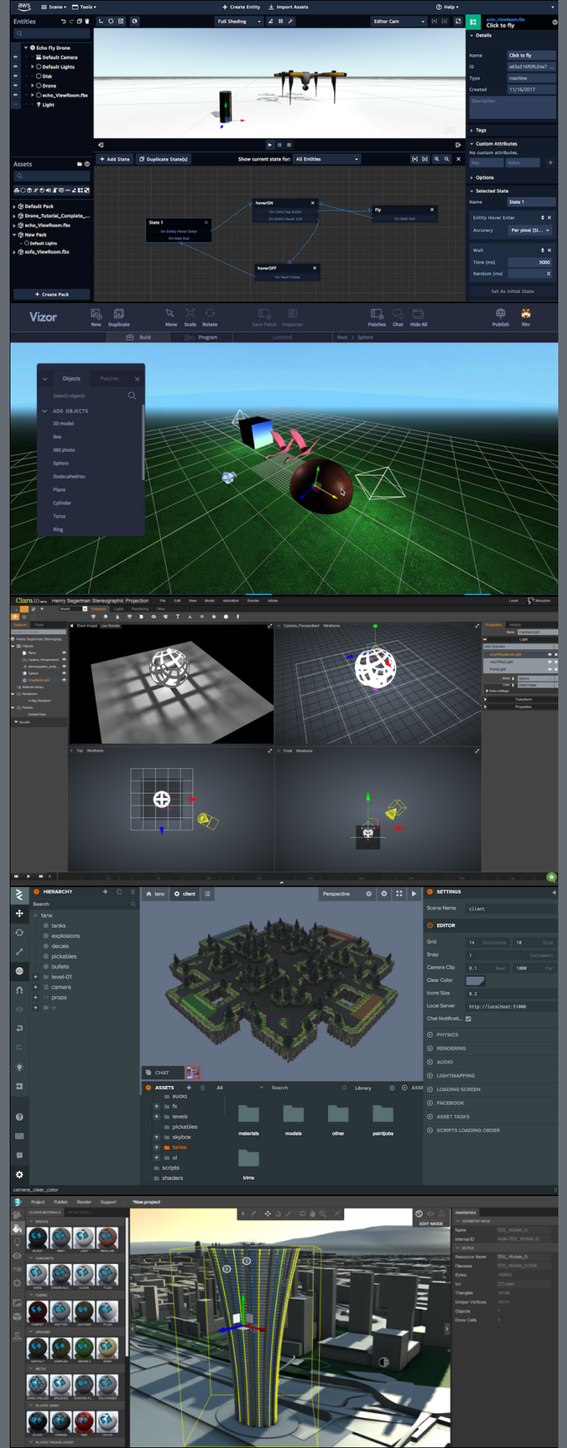
- 2003 OpenGL ES - cross-language and multi-platform 3D graphics API.
  Hardware accelerated rendering of 3D objects.
- WebGL 1.0 introduced by Mozilla based on OpenGL ES uses DOM (canvas element)
- X3D (web3d.org) introduced X3DOM

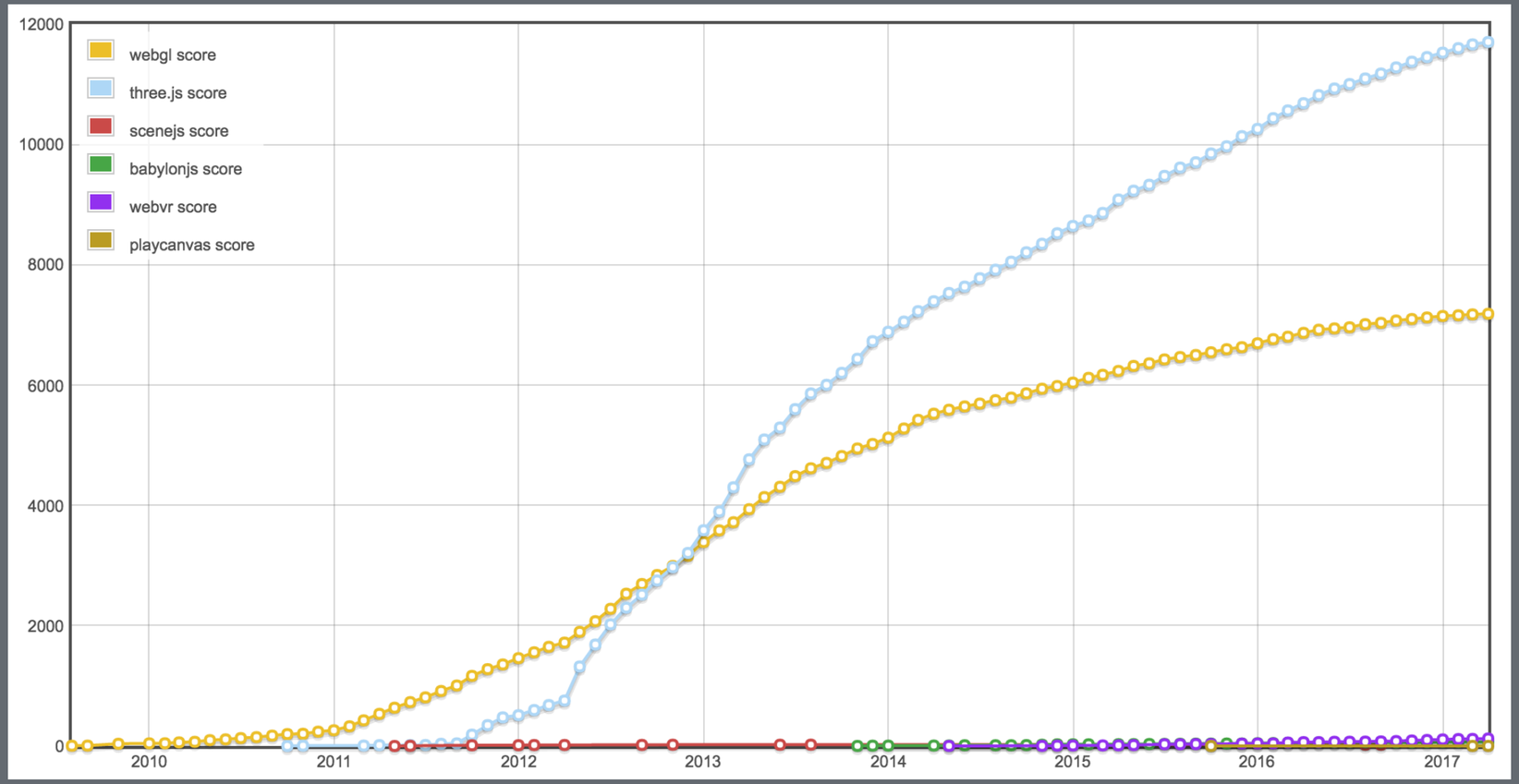
Most popular WebGL Engines

- Three.js by Ricardo Cabello in 2010
- Babylon.js
- p5*js
- argon.js
- Scene.js
- Turbulenz

Frameworks & Online WebVR Platforms

- A-Frame
- ReactVR
- Primrose
- Janus VR
- AWS Sumerian
- Vizor.io
- Clara.io
- Play Canvas
- Cl3ver





Stackoverflow score growth over time by tag comparison

WebGL Framework Comparison

What is WebVR?

WebVR is a *experimental* Javascript API that provides interfaces to VR hardware to allow developers to build compelling, comfortable VR experiences on the web.

WebVR

WebVR v1.1 (Legacy) *Editor's Draft, 12/12/17*
- deviceOrientation
- VRDisplay object handles almost everything
- available on multiple platforms

WebXR Device API (WebVR 2.0) *Editor's Draft, 01/11/18*
- not backwards compatible
- device support
- introduction of VRDevice & VRSession to replace VRDisplay
- read Explainer doc!

Goals of WebVR

- support for accessing virtual reality (VR) and augmented reality (AR) devices, including sensors and head-mounted displays, on the Web.
- Detect available virtual reality devices.
- Query the device's capabilities.
- Poll the device's position and orientation.
- Display imagery on the device at the appropriate frame rate.

Non-goals of WebVR

- Define how a virtual reality browser would work.
- Take full advantage of augmented reality devices.
- Build “The Metaverse.”

Check out https://webvr.rocks/ for support updates

Demo Comparison A & B

```
var isUserInteracting = false,
onMouseDownMouseX = 0, onMouseDownMouseY = 0,
lon = 0, onMouseDownLon = 0,
lat = 0, onMouseDownLat = 0,
phi = 0, theta = 0;

init();
animate();

function init() {

    var container, mesh;

    container = document.getElementById( 'container' );

    camera = new THREE.PerspectiveCamera( 75, window.innerWidth /
    camera.target = new THREE.Vector3( 0, 0, 0 );

    scene = new THREE.Scene();

    var geometry = new THREE.SphereGeometry( 500, 60, 40 );
    geometry.scale( - 1, 1, 1 );

    var material = new THREE.MeshBasicMaterial( {
        map: new THREE.TextureLoader().load( 'assets/sky.jpg' )
    } );

    mesh = new THREE.Mesh( geometry, material );

    scene.add( mesh );

    renderer = new THREE.WebGLRenderer();
}

function onDocumentMouseWheel( event ) {

    camera.fov += event.deltaY * 0.05;
    camera.updateProjectionMatrix();

}

function animate() {

    requestAnimationFrame( animate );
    update();

}

function update() {

    if ( isUserInteracting === false ) {

        lon += 0.05;

    }

    lat = Math.max( - 85, Math.min( 85, lat ) );
    phi = THREE.Math.degToRad( 90 - lat );
    theta = THREE.Math.degToRad( lon );

    camera.target.x = 500 * Math.sin( phi ) * Math.cos( theta );
    camera.target.y = 500 * Math.cos( phi );
    camera.target.z = 500 * Math.sin( phi ) * Math.sin( theta );
```

Three.js and WebVR

comparison with

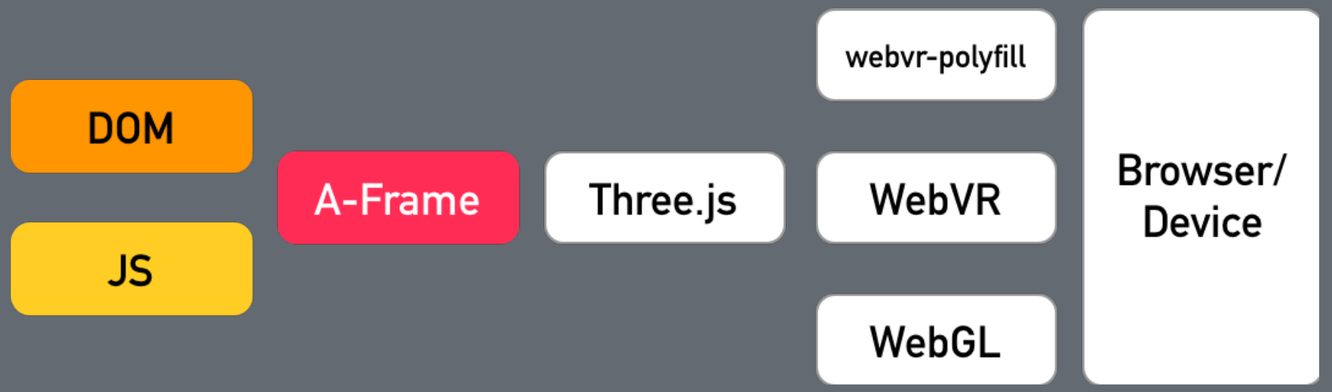
A-Frame

```
<!DOCTYPE html>
<html>
    <head>
        <title>A-Frame - Photosphere demo</title>
        <meta name="description" content="A-Frame - Photosphere demo">
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
        <script src="https://aframe.io/releases/0.5.0/aframe.min.js"></script>
    </head>
    <body>
        <a-scene>
            <a-sky src="assets/sky.jpg">
                <a-animation attribute="rotation"
                    dur="90000"
                    fill="forwards"
                    to="0 360 0"
                    repeat="indefinite"
                    easing="linear"></a-animation>
            </a-sky>
        </a-scene>
    </body>
</html>
```

A-Frame

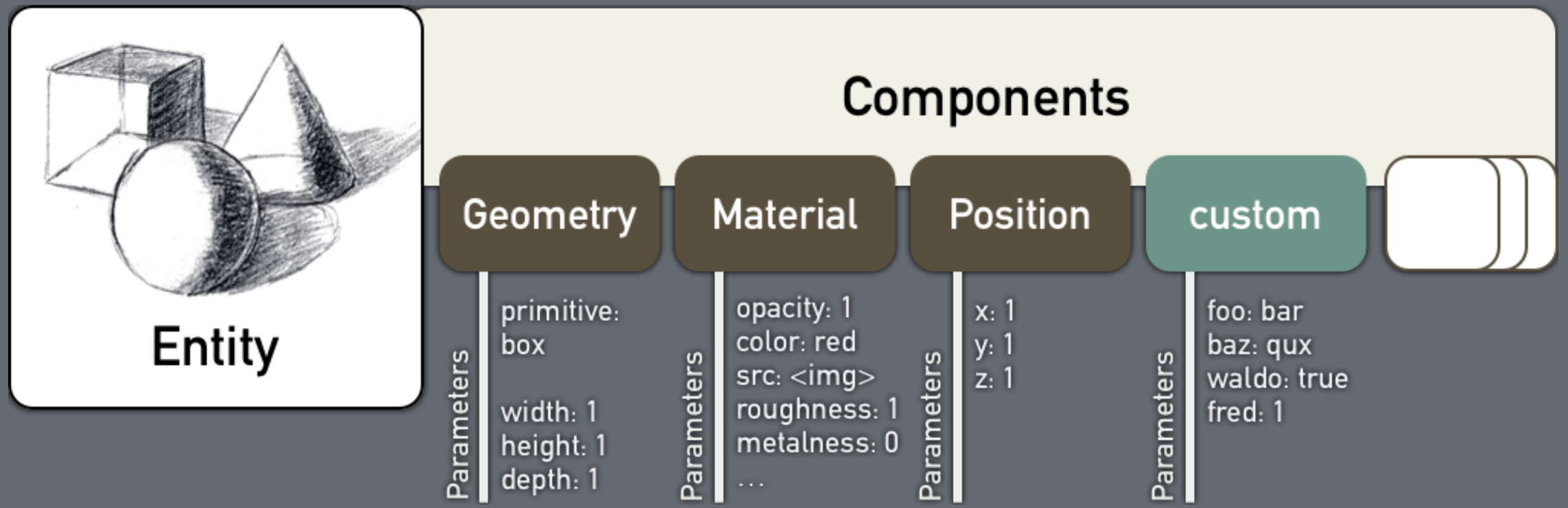
- a 3DML (3D markup language) like X3Dom and ReactVR
- DOM-based Entity-Component System => declarative & extensible (similar to Unity/UE4/PlayCanvas)

The A-Frame Stack:



A-Frame

Entity-Component System



A-Frame

- Entities are HTML elements (i.e., `<a-entity>`)
- Components are HTML attributes, set on the entity

```
<body>
    <a-scene>
        <a-entity geometry="primitive:box" material="color:#c00">
        </a-entity>
    </a-scene>
</body>
```

Entity

- general purpose objects (e.g. create a player, ball, or field)
- they inherently have a position, rotation and scale in a scene

```
<a-entity
    geometry="primitive:box;
              width:0.2;
              height:0.3;
              depth:0.5;"
    material="color:#c00"
    position="0 0 -1"
    rotation="0 30 30"
    material="color:#c00">
</a-entity>
```

Primitives (concise, semantic building blocks)

```
<a-entity geometry="primitive:box;
                    width:0.2;
                    height:0.3;
                    depth:0.5;"
          material="color:#c00">
</a-entity>

<a-box width="0.2"
       height="0.3"
       depth="0.5"
       material="color:#c00">
</a-box>
```

Mixins

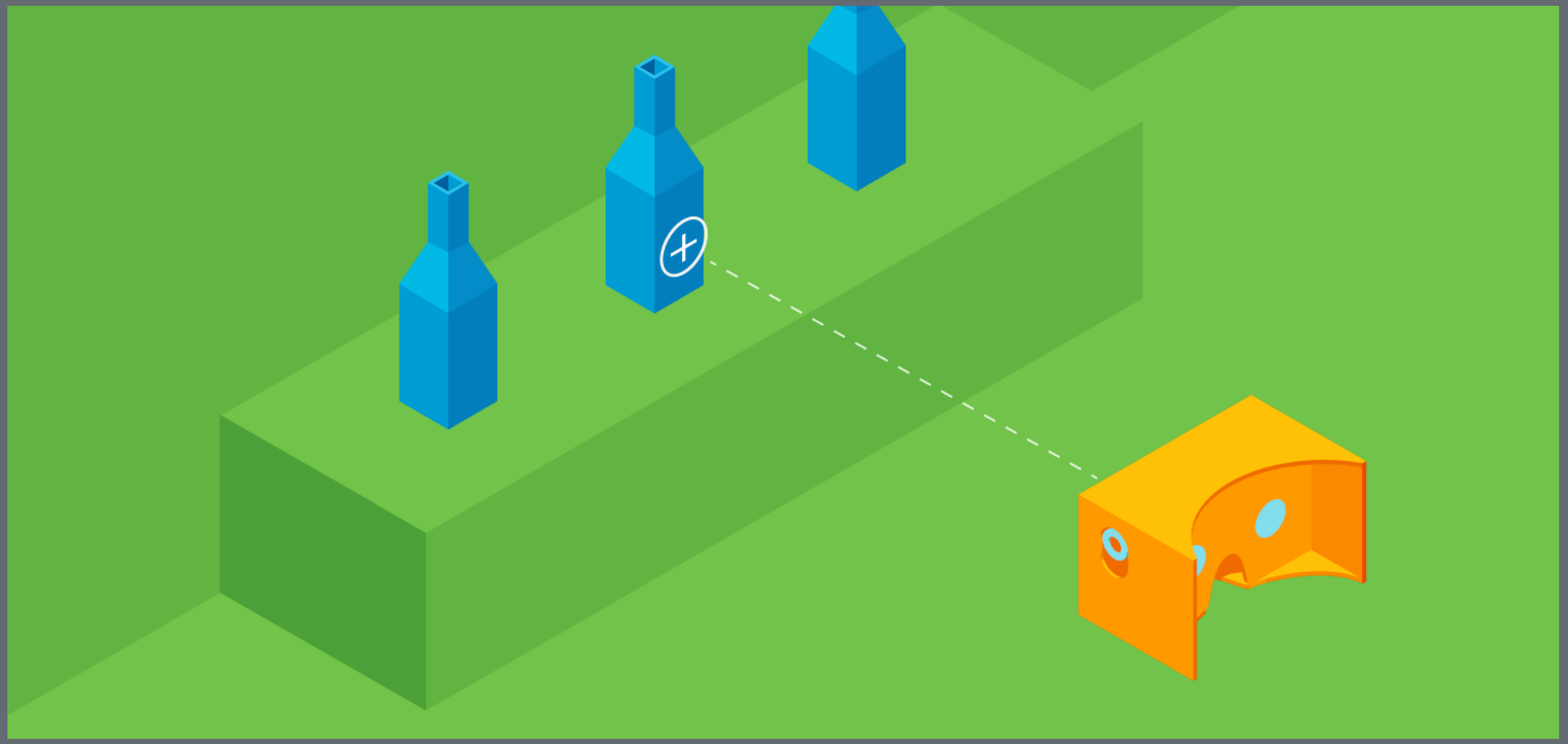
```
<a-mixin id="box"
         geometry="primitive:box;
                   width:0.2;
                   height:0.3;
                   depth:0.5;"
         material="color:#c00">
</a-mixin>

<a-entity mixin="box">
</a-entity>
```

Custom Components

```
AFRAME.registerComponent('foo', {
    schema: {
        bar: {type: 'number'},
        baz: {type: 'string'}
    },
    init: function () {
        // Do something when component is plugged in.
    },
    update: function () {
        // Do something when component's data is updated.
    }
});
```

Gaze-Based Control

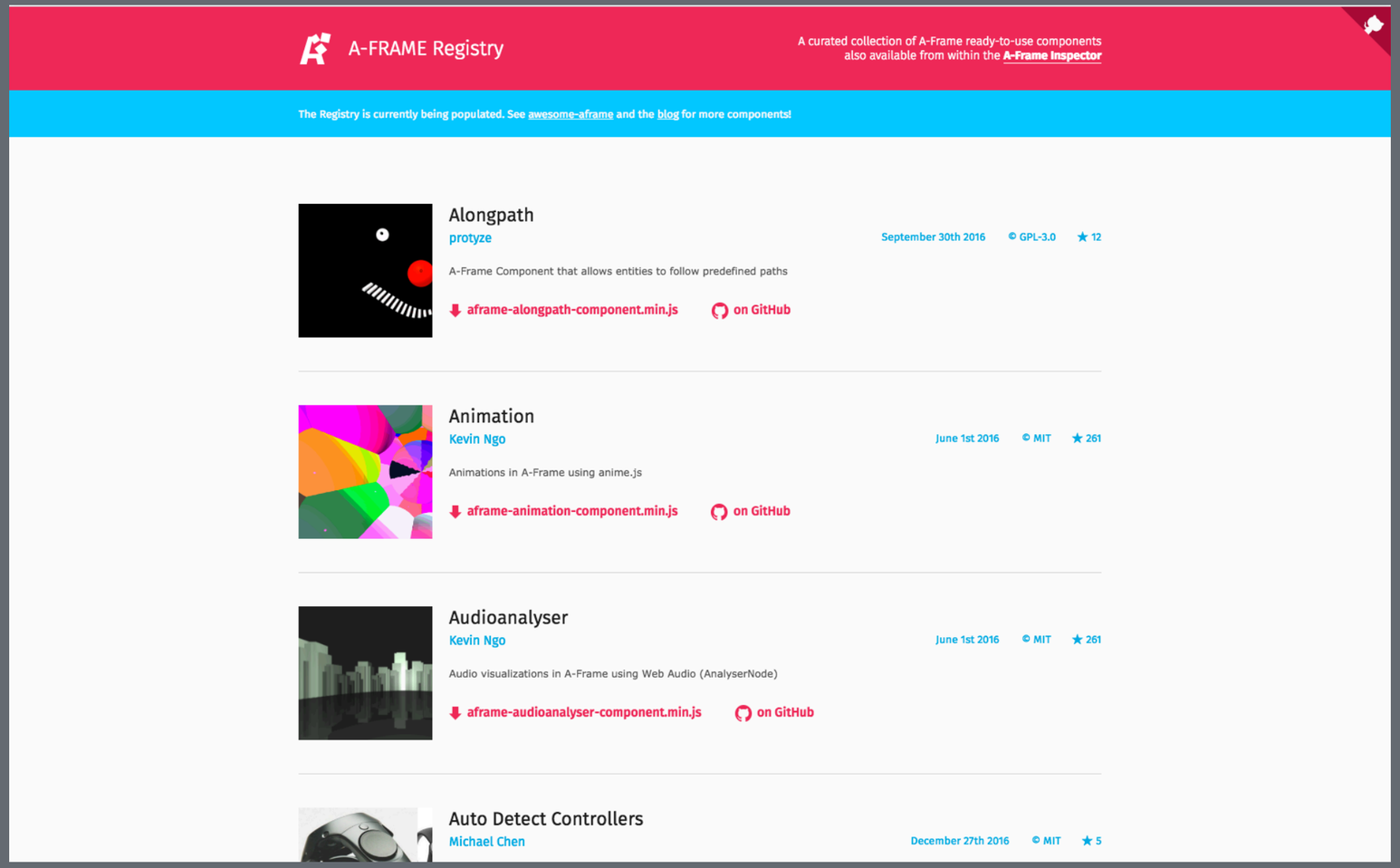


Gaze-Based Cursor Interactions

```
AFRAME.registerComponent('clickable', {
    init: function () {
        var el = this.el;
        el.addEventListener('mouseenter', function () {
            el.setAttribute('color', '#f00');
        });
        el.addEventListener('mouseleave', function () {
            el.setAttribute('color', '#fff');
        });
        el.addEventListener('click', function () {
            el.setAttribute('material', 'src' , 'assets/football.png');
        });
    }
});
```

```
<a-sphere clickable src="assets/basketball.png" radius="0.5" color="#fff"></a-sphere>
<a-camera><a-cursor></a-cursor></a-camera>
```

A-Frame Component Registry



Introduction

A-Frame is a web framework for building virtual reality experiences. It was started by Mozilla VR to make WebVR content creation easier, faster, and more accessible.

A-Frame lets you build scenes with just **HTML** while having unlimited access to JavaScript, three.js, and all existing Web APIs. A-Frame uses an **entity-component-system** pattern that promotes composition and extensibility. It is free and open source with a welcoming community and a thriving **ecosystem of tools and components**.

HTML

HTML is one of the easiest languages to understand, and many of us are already familiar with it. There are no build steps or boilerplate required nor anything to install; all we need is an HTML file:

```HTML
<html>
  <head>
    <script src="https://aframe.io/releases/0.5.0/aframe.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-box color="#6173F4" opacity="0.8" depth="2"></a-box>
      <a-sphere radius="2" src="texture.png" position="1 1 0"></a-sphere>
```

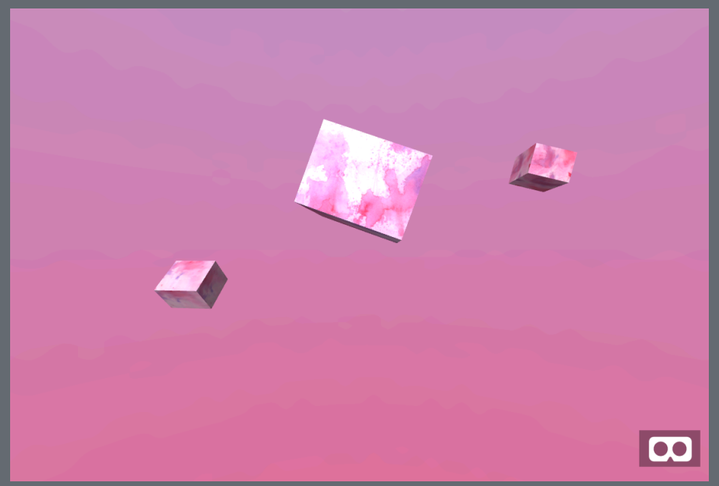
A-Frame Inspector

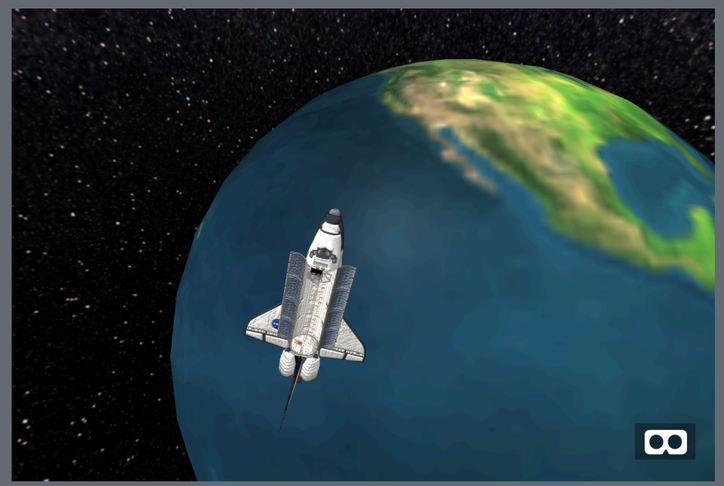
[control]+[option]+[i]

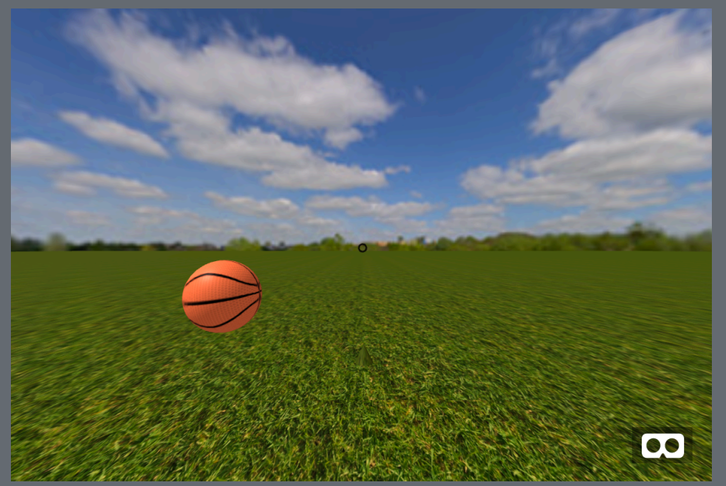

Create responsibly!

It's code time! Here's what we're building:

- github.com/roland-dubois/aframe-meetup-nyc
- XAMPP, MAMP or `$ python -m SimpleHTTPServer`
- **JSBin** or **Glitch**







Close up discussion

Mozilla is hosting a new WebVR Experience Challenge
Submissions close **April 2nd**!

- Group projects
- Challenges / Codathons
- Deeper looks into A-Frame Components

Resources

- A-Frame Slack
- A-Frame Component Registry
- vr.mozilla.org
- WebVR API
- The UX of VR

Get In Touch

@rolanddubois | rolanddubois.com