HTML5 Conference 2016

そしてWebVR

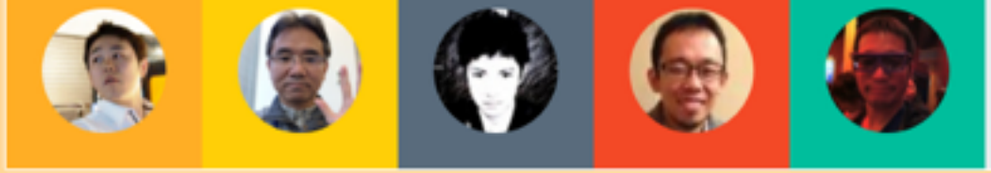
@2016.09.03 比留間 和也

自己紹介

比留間　和也

@edo_m18　　@edom18

HTML5 Experts.jp
日本に、もっとエキスパートを。



HTML5 Experts.jp
エキスパート紹介

JSだけでVRできる！『WebVR』ことはじめ NEW

比留間 和也

Oculus Rift, three.js, VR, WebVR

2016年3月23日

今年はVR元年と呼ばれています。
実は過去にも何度かVR元年と呼ばれ、VRが来る、と言われていた年があります。
ですが、今年はいよいよそれが本格的になりそうな状況になってきました。

そこで今回は、「今からWebVRに備えよう！」ということで、WebVRとはなにか、それを利用して何ができるのか、利用シーンはどうか、といったことに焦点を当てたいと思います。

VR元年

本格的になりそうなのはなぜか。まずひとつ挙げられるのはOculus Rift（詳細は後述）に代表される、いくつかのヘッドマウントディスプレイ（以下、HMD）が比較的安価に家庭で利用できるようになったことです。
今年に発売される予定のものだけでも、Oculus Rift、HTC Vive、PlayStation VR、FOVE(開発キット版)など様々なHMDが市場に登場する予定になっています。
またこれらが注目される理由として、現在のPCのハードウェア性能が、HMDの必要としている要件を満たしてきた、という点も大きいでしょう。

しかしながら、Oculus Riftに関して言えば 2,160 x 1,200 という高解像度の画面を90FPS、つまり1秒間に90回もレンダリングをしなければなりません。
これは最近のPCであっても、ミドルハイクラス〜ハイエンドクラスのGPUが搭載されているPCが必要で

週間PVランキング > more

1 JSだけでVRできる！『WebVR』ことはじめ
2 Windows 10の2つのWebブラウザ、Microsoft EdgeとInternet Explorer 11
3 今話題のReact.jsはどのようなWebアプリケーションに適しているか？ Introduction To React— Frontrend Conference
4 Angular2実践入門〜ng-japan 2016 セッションレポート〜
5 Bootstrap3超速レビュー！刷新されたグリッドシステムを理解しよう！

新着記事 > more

JSだけでVRできる！『WebVR』ことはじめ
Angular2実践入門〜ng-japan 2016 セッションレポート〜
エキスパートたちが語り尽くす、WebRTCの「つらみ」ーWebRTC Conferenceパネルディスカッションレポート
夏野剛・及川卓也・白石俊平が語る「WebRTCが切り拓く2020年のIoT」〜リアルタイムコミュニケーションがもたらす破壊的イノベーション〜

WebVRについての記事書きました

VRとは？

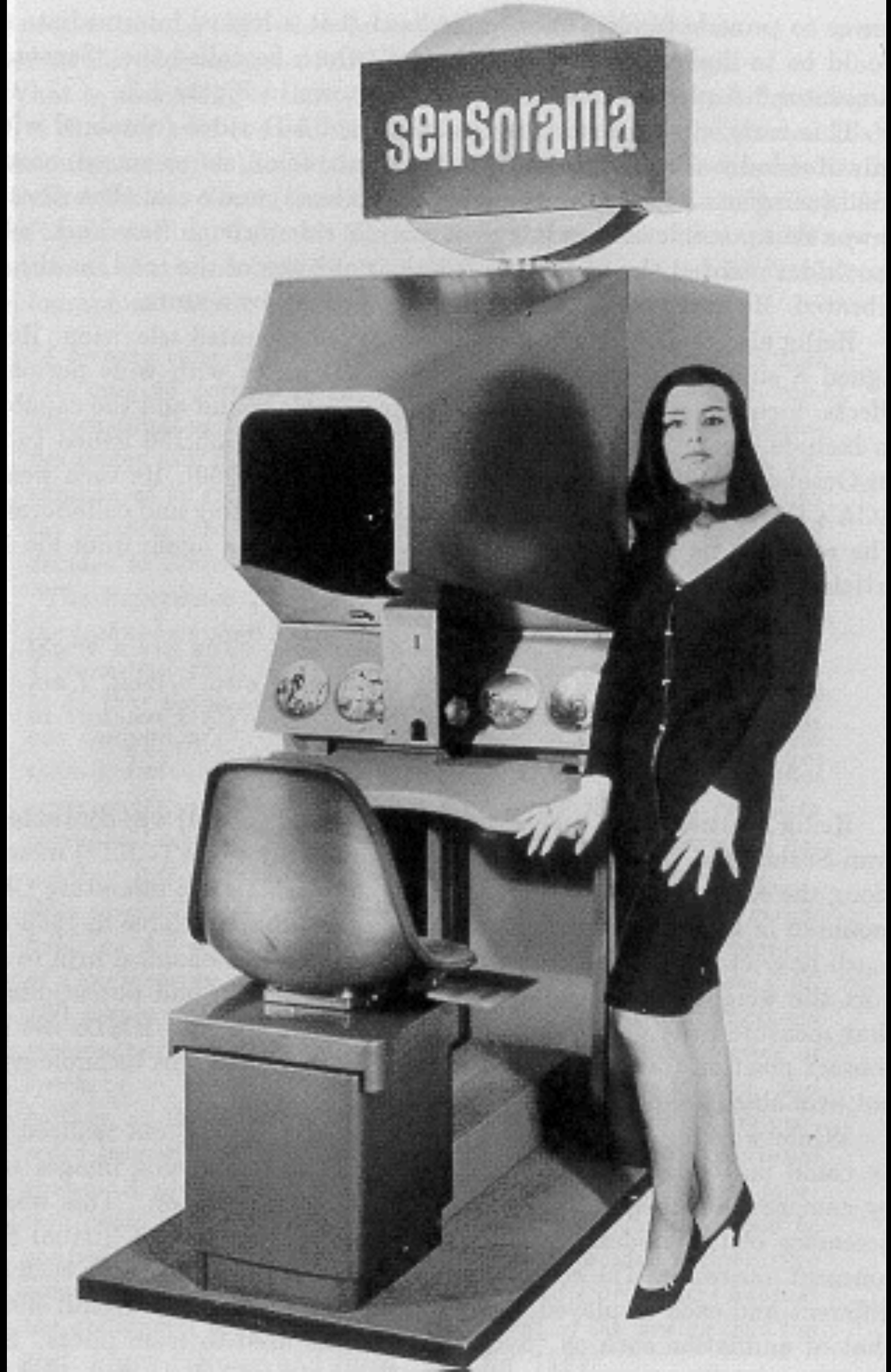
今年は

VR元年

VR元年？

実はVRの登場は

1960年代

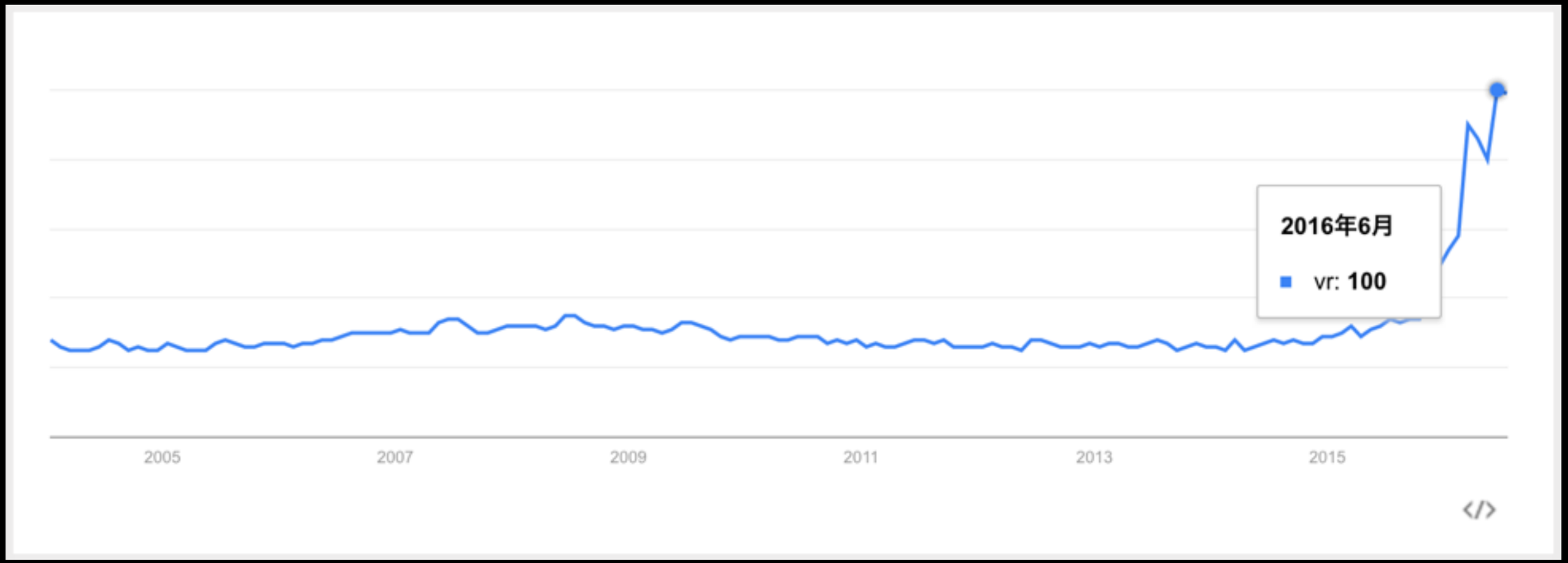


画像元

当時は**デパート**の

アトラクション

時は流れて2016年



Googleトレンドで急上昇！

この流れの立役者は


oculus


Oculus Rift

DK1は2012年8月に登場

画像元

翌々年、2014年3月にDK2が発表

画像元

そして様々なHMDが

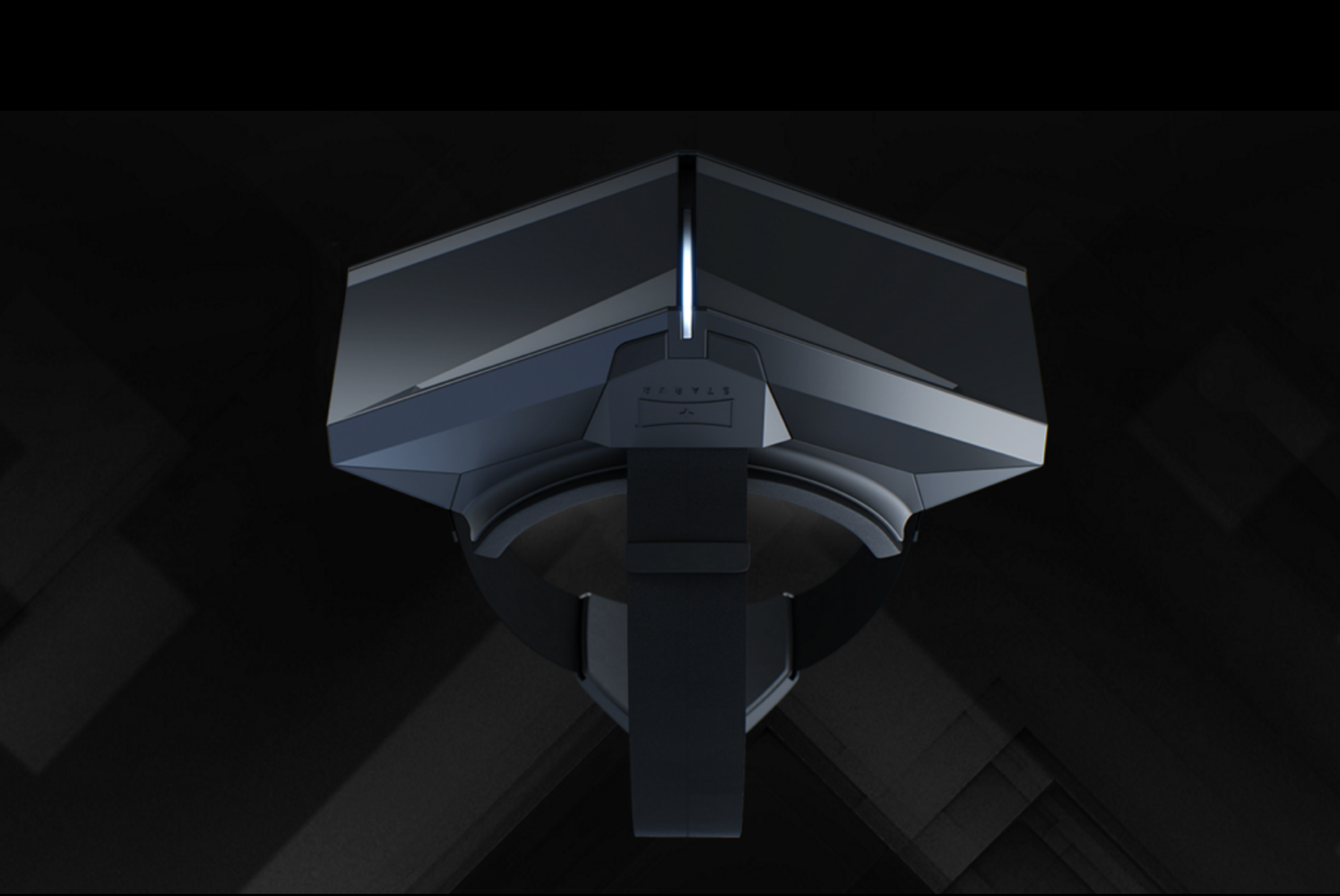

htc
htc
STEAM VR
POWERED


SONY
PS4

FOVE

FOVE

WATCH VIDEO




OSVR
OSVR
OSVR


ハコスコ
手軽で楽しい360°VR体験
360°動画を見る
オリジナルプリント


Google

改めて

WebVRとは？

VRをWebで

実現する技術

ではない

WebVRは

APIの名称

仕様はW3CのGitHub上に
ホストされている

https://w3c.github.io/webvr/

仕様一覧

Table of contents

Table of contents

たったこれだけ

VR体験のための要素

Components of a VR experience

Let's take a look at the key components required for any VR experience:

1. The VR display that we are rendering content to.
2. The user's pose. The orientation and positioning of the headset in space.
3. Eye parameters that define stereo separation and field of view.

Here's a look at the workflow sequence for getting content into the headset:

1. `navigator.getVRDisplays()` to retrieve a VR Display.
2. Create a `<canvas>` element which we will use to render content.
3. Use `VRDisplay.requestPresent()` to pass in the canvas element.
4. Create the VR device-specific animation loop in which we will perform content rendering.
    a. `VRDisplay.getPose()` to update the user's pose.
    b. Perform calculations and rendering.
    c. Use `VRDisplay.submitFrame()` to indicate to the compositor when the canvas element content is ready to be presented in the VR display.

The following sections describe each of these actions in detail.

https://hacks.mozilla.org/2016/03/introducing-the-webvr-1-0-api-proposal/

Components of a VR experience

Let's take a look at the key components required for any VR experience:

1. The VR display that we are rendering content to.
2. The user's pose. The orientation and positioning of the headset in space.
3. Eye parameters that define stereo separation and field of view.

Here's a look at the workflow sequence for getting content into the headset:

1. `navigator.getVRDisplays()` to retrieve a VR Display.
2. Create a `<canvas>` element which we will use to render content.
3. Use `VRDisplay.requestPresent()` to pass in the canvas element.
4. Create the VR device-specific animation loop in which we will perform content rendering.
    a. `VRDisplay.getPose()` to update the user's pose.
    b. Perform calculations and rendering.
    c. Use `VRDisplay.submitFrame()` to indicate to the compositor when the canvas element content is ready to be presented in the VR display.

The following sections describe each of these actions in detail.

https://hacks.mozilla.org/2016/03/introducing-the-webvr-1-0-api-proposal/

つまり

APIはデバイスやセンサーからの
情報取得が主なお仕事

3D部分は

WebGLで作る

あくまで3Dコンテンツ制作の

延長線上にVRがある

セットアップ方法

基本的に

センサーからの値を取り出して
画面を更新するだけ

セットアップコード

```
navigator.getVRDisplays().then(function (displays) {
  if (!displays.length) {
    // WebVR is supported, no VRDisplays are found.
    return;
  }

  // Handle VRDisplay objects. (Exposing as a global variable for use elsewhere.)
  vrDisplay = displays.length[0];
}).catch(function (err) {
  console.error('Could not get VRDisplays', err.stack);
});
```

```
// Use 'left' or 'right'.
var eyeParameter = vrDisplay.getEyeParameters('left');

var width = eyeParameter.renderWidth;
var height = eyeParameter.renderHeight;
```

```
// Select WebGL canvas element from document.
var webglCanvas = document.querySelector('#webglcanvas');
var enterVRBtn = document.querySelector('#entervr');

enterVRBtn.addEventListener('click', function () {
  // Request to present WebGL canvas into the VR display.
  vrDisplay.requestPresent({source: webglCanvas});
});

// To later discontinue presenting content into the headset.
vrDisplay.exitPresent();
```

```
var id = vrDisplay.requestAnimationFrame(onAnimationFrame);

function onAnimationFrame () {
  // Render loop.
  id = vrDisplay.requestAnimationFrame(onAnimationFrame);
}

// To cancel the animation loop.
vrDisplay.cancelRequestAnimationFrame(id);
```

```
var pose = vrDisplay.getPose();

// Returns a quaternion.
var orientation = pose.orientation;

// Returns a three-component vector of absolute position.
var position = pose.position;
```

```
// Pass in either 'left' or 'right' eye as parameter.
var eyeParameters = vrDisplay.getEyeParameters('left');

// After translating world coordinates based on VRPose, transform again by negative of the eye offset.
var eyeOffset = eyeParameters.offset;

// Project with a projection matrix.
var eyeMatrix = makeProjectionMatrix(vrDisplay, eyeParameters);
```

```
function fieldOfViewToProjectionMatrix (fov, zNear, zFar) {
    var upTan = Math.tan(fov.upDegrees * Math.PI / 180.0);
    var downTan = Math.tan(fov.downDegrees * Math.PI / 180.0);
    var leftTan = Math.tan(fov.leftDegrees * Math.PI / 180.0);
    var rightTan = Math.tan(fov.rightDegrees * Math.PI / 180.0);
    var xScale = 2.0 / (leftTan + rightTan);
    var yScale = 2.0 / (upTan + downTan);

    var out = new Float32Array(16);
    out[0] = xScale;
    out[1] = 0.0;
    out[2] = 0.0;
    out[3] = 0.0;
    out[4] = 0.0;
    out[5] = yScale;
    out[6] = 0.0;
    out[7] = 0.0;
    out[8] = -((leftTan - rightTan) * xScale * 0.5);
    out[9] = ((upTan - downTan) * yScale * 0.5);
    out[10] = -(zNear + zFar) / (zFar - zNear);
    out[11] = -1.0;
    out[12] = 0.0;
    out[13] = 0.0;
    out[14] = -(2.0 * zFar * zNear) / (zFar - zNear);
    out[15] = 0.0;

    return out;
}
```

引用: https://w3c.github.io/webvr/#interface-interface-vrfieldofview

3Dコンテンツの制作および
画面の更新は
自分で作らないとならない

Three.jsを使えば手軽

```
// Setup scene
scene = new THREE.Scene();

// Setup camera (player)
player = new THREE.Object3D();
player.name = 'player';
player.position.set(0, 1.4, 10);

camera = new THREE.PerspectiveCamera(70, window.innerWidth / window.innerHeight, 0.1, 100000);
player.add(camera);
scene.add(player);


// Setup renderer
renderer = new THREE.WebGLRenderer({antialias: true});
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0x000000);
document.body.appendChild(renderer.domElement);

// setup light
light = new THREE.DirectionalLight(0xffffff);
light.position.set(0, -3, 4);
scene.add(light);

// Setup skybox
var urls = [
    'img/Vasa/posx.jpg',
    'img/Vasa/negx.jpg',
    'img/Vasa/posy.jpg',
    'img/Vasa/negy.jpg',
    'img/Vasa/posz.jpg',
    'img/Vasa/negz.jpg',
];

// Fixed warning that `there is no texture bound to the unit 0`
new THREE.CubeTextureLoader().load(urls, function (texture) {
    texture.mapping = THREE.CubeRefractionMapping;
    var shader = THREE.ShaderLib['cube'];
    shader.uniforms['tCube'].value = texture;
    var material = new THREE.ShaderMaterial({
```

```
// Setup scene
scene = new THREE.Scene();

// Setup camera (player)
player = new THREE.Object3D();
player.name = 'player';
player.position.set(0, 1.4, 10);

camera = new THREE.PerspectiveCamera(70, window.innerWidth / window.innerHeight, 0.1, 100000);
player.add(camera);
scene.add(player);


// Setup renderer
renderer = new THREE.WebGLRenderer({antialias: true});
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0x000000);
document.body.appendChild(renderer.domElement);

// setup light
light = new THREE.DirectionalLight(0xffffff);
light.position.set(0, -3, 4);
scene.add(light);

// Setup skybox
var urls = [
    'img/Vasa/posx.jpg',
    'img/Vasa/negx.jpg',
    'img/Vasa/posy.jpg',
    'img/Vasa/negy.jpg',
    'img/Vasa/posz.jpg',
    'img/Vasa/negz.jpg',
];

// Fixed warning that `there is no texture bound to the unit 0`
new THREE.CubeTextureLoader().load(urls, function (texture) {
    texture.mapping = THREE.CubeRefractionMapping;
    var shader = THREE.ShaderLib['cube'];
    shader.uniforms['tCube'].value = texture;
    var material = new THREE.ShaderMaterial({
```

```
// Setup scene
scene = new THREE.Scene();

// Setup camera (player)
player = new THREE.Object3D();
player.name = 'player';
player.position.set(0, 1.4, 10);

camera = new THREE.PerspectiveCamera(70, window.innerWidth / window.innerHeight, 0.1, 100000);
player.add(camera);
scene.add(player);


// Setup renderer
renderer = new THREE.WebGLRenderer({antialias: true});
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0x000000);
document.body.appendChild(renderer.domElement);

// setup light
light = new THREE.DirectionalLight(0xffffff);
light.position.set(0, -3, 4);
scene.add(light);

// Setup skybox
var urls = [
    'img/Vasa/posx.jpg',
    'img/Vasa/negx.jpg',
    'img/Vasa/posy.jpg',
    'img/Vasa/negy.jpg',
    'img/Vasa/posz.jpg',
    'img/Vasa/negz.jpg',
];

// Fixed warning that `there is no texture bound to the unit 0`
new THREE.CubeTextureLoader().load(urls, function (texture) {
    texture.mapping = THREE.CubeRefractionMapping;
    var shader = THREE.ShaderLib['cube'];
    shader.uniforms['tCube'].value = texture;
    var material = new THREE.ShaderMaterial({
```

```
// Setup scene
scene = new THREE.Scene();

// Setup camera (player)
player = new THREE.Object3D();
player.name = 'player';
player.position.set(0, 1.4, 10);

camera = new THREE.PerspectiveCamera(70, window.innerWidth / window.innerHeight, 0.1, 100000);
player.add(camera);
scene.add(player);


// Setup renderer
renderer = new THREE.WebGLRenderer({antialias: true});
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.setClearColor(0x000000);
document.body.appendChild(renderer.domElement);
```

```
// setup light
light = new THREE.DirectionalLight(0xffffff);
light.position.set(0, -3, 4);
scene.add(light);

// Setup skybox
var urls = [
    'img/Vasa/posx.jpg',
    'img/Vasa/negx.jpg',
    'img/Vasa/posy.jpg',
    'img/Vasa/negy.jpg',
    'img/Vasa/posz.jpg',
    'img/Vasa/negz.jpg',
];

// Fixed warning that `there is no texture bound to the unit 0`
new THREE.CubeTextureLoader().load(urls, function (texture) {
    texture.mapping = THREE.CubeRefractionMapping;
    var shader = THREE.ShaderLib['cube'];
    shader.uniforms['tCube'].value = texture;
```

```
// setup light
light = new THREE.DirectionalLight(0xffffff);
light.position.set(0, -3, 4);
scene.add(light);

// Setup skybox
var urls = [
    'img/Vasa/posx.jpg',
    'img/Vasa/negx.jpg',
    'img/Vasa/posy.jpg',
    'img/Vasa/negy.jpg',
    'img/Vasa/posz.jpg',
    'img/Vasa/negz.jpg',
];

// Fixed warning that `there is no texture bound to the unit 0`
new THREE.CubeTextureLoader().load(urls, function (texture) {
    texture.mapping = THREE.CubeRefractionMapping;
    var shader = THREE.ShaderLib['cube'];
    shader.uniforms['tCube'].value = texture;
    var material = new THREE.ShaderMaterial({
        fragmentShader: shader.fragmentShader,
        vertexShader: shader.vertexShader,
        uniforms: shader.uniforms,
        side: THREE.BackSide
    });

    var s = 50000;
    var mesh = new THREE.Mesh(new THREE.BoxGeometry(s, s, s), material);
    scene.add(mesh);
});

// VR Setup
controls = new THREE.VRControls(camera);
effect = new THREE.VREffect(renderer);

// Start VR preseting.
document.addEventListener('click', function () {
    effect.setSize(window.innerWidth, window.innerHeight);
    effect.requestPresent();
}, false);
```

```
// setup light
light = new THREE.DirectionalLight(0xffffff);
light.position.set(0, -3, 4);
scene.add(light);

// Setup skybox
var urls = [
    'img/Vasa/posx.jpg',
    'img/Vasa/negx.jpg',
    'img/Vasa/posy.jpg',
    'img/Vasa/negy.jpg',
    'img/Vasa/posz.jpg',
    'img/Vasa/negz.jpg',
];

// Fixed warning that `there is no texture bound to the unit 0`
new THREE.CubeTextureLoader().load(urls, function (texture) {
    texture.mapping = THREE.CubeRefractionMapping;
    var shader = THREE.ShaderLib['cube'];
    shader.uniforms['tCube'].value = texture;
    var material = new THREE.ShaderMaterial({
        fragmentShader: shader.fragmentShader,
        vertexShader: shader.vertexShader,
        uniforms: shader.uniforms,
        side: THREE.BackSide
    });

    var s = 50000;
    var mesh = new THREE.Mesh(new THREE.BoxGeometry(s, s, s), material);
    scene.add(mesh);
});

// VR Setup
controls = new THREE.VRControls(camera);
effect = new THREE.VREffect(renderer);

// Start VR preseting.
document.addEventListener('click', function () {
    effect.setSize(window.innerWidth, window.innerHeight);
    effect.requestPresent();
}, false);
```

対応ブラウザ

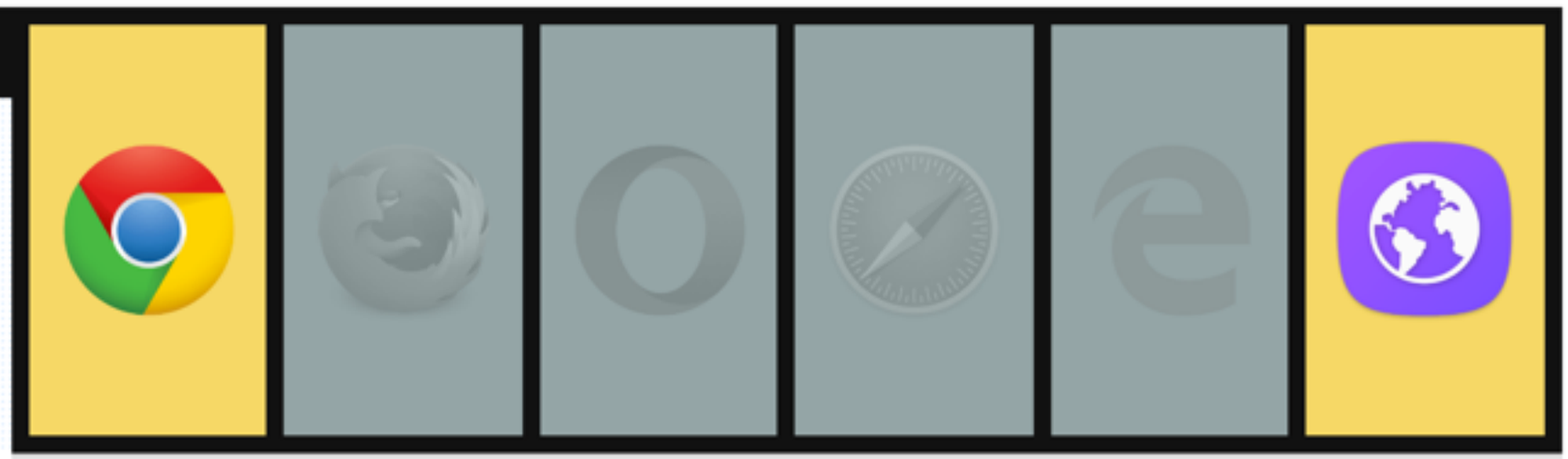
WebVR enthusiasm

The first thing any implementation needs.

navigator.getVRDisplays

Namespace for page-side WebVR API. Resolves an array of available VRDisplays.

Spec. Test.



Chrome: Supported in experimental builds.

Firefox: Uses old API interface: navigator.getVRDevices. New builds landing in Nightly end of May.

Samsung Internet for GearVR: Experimental support can be enabled.

参考: Is WebVR Ready?

まだ

安定版ビルドでサポートした

ブラウザはない

（特定のビルドやプラグインを必要とする）

VRの成功・活性化は

Webが鍵

ガチVRに比べて、少しだけ

VRを使うこともできる

例えば

ECサイトの
商品紹介を**VR**にする

http://getnews.jp/archives/1515661

例えば

不動産サイトで物件の閲覧をVRで

WebにもVR活用の

ネタはたくさんある

まとめ

- WebVRはAPIの名称
- WebVR APIがやってくれるのはセンサーの値を取るなどの最低限の処理のみ
- Three.jsを使えば手軽にVRコンテンツの制作が可能
- とはいえまだサポートブラウザが（ほぼ）ない
- しかし、Webでの活用はまだまだ（いい意味で）未知数

- アンケートにご協力を。
  WebGLやWebVRについて、もっと深く聞きたいとか、こういう内容でやってほしい、などありましたらアンケートにてその旨をお伝え下さい。

ご清聴ありがとうございました