



FDC126343

Different Tools to Implement Your WebVR Application

Zhong Wu and Kevin Vandecar
Autodesk

Learning Objectives

- Learn how to use Three.js and Forge Viewer to make a WebVR application
- Learn about WebVR hardware and devices
- Learn about different frameworks that can be used to implement a WebVR application
- Learn about simple practical examples that can help with WebVR

Description

With all kinds of different VR devices released to the market, the concept of VR/AR/MR is becoming increasingly popular, but it's still an expensive experience. Is there any way to make the VR experience more accessible, and easier for your customers? WebVR provides a good solution. This class will introduce WebVR and show you some cool demos to help you get a quick project started with the technology. There are several frameworks you can start with, and even mix together - we will discuss some of these, including Three.js, the Autodesk Forge Viewer, and ReactVR. The class will show live demos, and the code behind the different examples.

Your Forge DevCon Expert(s)

Kevin Vandecar
Forge Developer Advocate
Autodesk, Inc.

Kevin Vandecar is a Forge developer advocate and also the manager for the Media & Entertainment Autodesk Developer Network Workgroup. His current specialty is 3ds Max software customization and programming areas. Recently, he has also been working on his web development skills and exploring areas in WebGL and Three.js, along with the new Autodesk Forge APIs.

Zhong Wu
Forge Developer Advocate
Autodesk, Inc.

Senior developer advocate in Autodesk with more than 6 years working experience on AutoCAD Architecture as a software developer, mainly focus on the API support/consultant for Maya/MotionBuilder, and also contribute in Forge support and evangelism. Experienced in C++, also be familiar with Javascript, Node.js, etc.



What is WebVR?

Virtual Reality on the Web?! Yes! Simple!? Can be...

A more proper definition provided by https://developer.mozilla.org/en-US/docs/Web/API/WebVR_API

“WebVR provides support for exposing virtual reality devices — for example head-mounted displays like the Oculus Rift or HTC Vive — to web apps, enabling developers to translate position and movement information from the display into movement around a 3D scene. This has numerous very interesting applications, from virtual product tours and interactive training apps to immersive first-person games.”

This presentation will cover the devices available, and how to develop applications on the web that will support WebVR, But first, let's discuss the benefits and why you may want to implement WebVR.

Specification and docs by WC3: <https://w3c.github.io/webvr/>

Mozilla docs: https://developer.mozilla.org/en-US/docs/Web/API/WebVR_API

Why WebVR?

Virtual reality (VR) has now become a common term. With Google Cardboard and Samsung GearVR, it is more accessible than ever. Most people have a smart phone and/or tablet that can support WebVR. But it's not as easy as just enabling it. The experience itself has to be pleasing and engaging. Design visualization, marketing, education and games are all great use cases. WebVR is a way to more easily push a VR experience to many devices.

Content – 2D

Using image and video technology is an easy way to get into WebVR. Stereo images and panorama image formats can be easy to generate and use. The drawback is that typically the user is stationary in a center location, and they spin themselves around to explore the VR space.

Autodesk's a360 Rendering service can also generate Stereo panoramic images. Autodesk techniques are discussed in a previous DevCon presentation:

<https://www.slideshare.net/Autodesk/creating-your-next-vr-walkthrough-with-cloud-rendered-stereo-panoramas>

Content – 3D

3D content can come from a variety of sources, and usually is a mix of different models into a scene, loaded by some engine. For example, you may have characters authored in 3ds Max or Blender, and combined with Fusion geometry, all loaded into an A-Frame scene as OBJ files. As an example, but certainly not complete, these tools may be of interest:

- Building/Authoring 3d Content
 - Autodesk 3dsMax and Maya for Entertainment
 - Blender for Entertainment



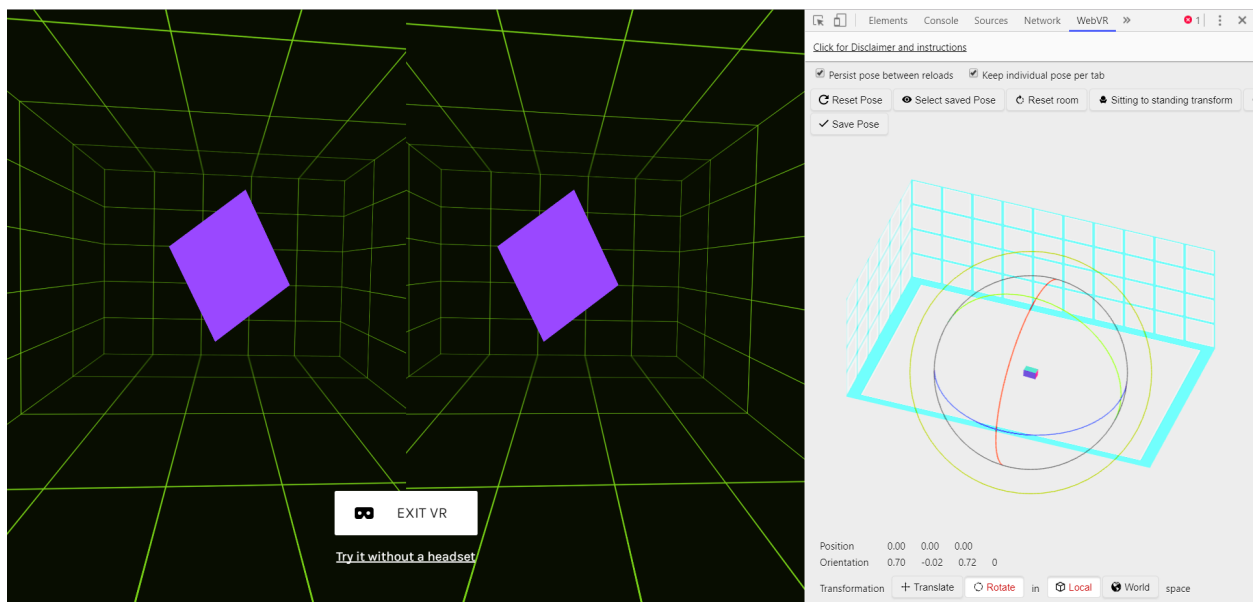
FORGE DevCon 2017

- Autodesk Fusion 360 for Mechanical/Product Design
- Dassault Systems Solidworks for Mechanical/Product Design
- Autodesk Revit
- Online Market Places
 - TurboSquid (<https://www.turbosquid.com/>)
 - Creative Market (<https://creativemarket.com/>)
 - GrabCAD (<https://grabcad.com/>)

Example:

Let's see a quick WebVR setup using ReactVR. For content, it is using a spherical image and a 3D model found as free resources online. <https://ironmanviewer.herokuapp.com/>

Note that in order to test, it helps to have a way to emulate the WebVR device on the desktop browser. One such tool for Chrome is the WebVR API Emulation plugin. You can find it in the Chrome store. It is currently located here: <https://chrome.google.com/webstore/detail/webvr-api-emulation/gbdnpaebafagiogggnhkacnaaahpiefil?hl=en>. Note it is an older tool, and uses the polyfill library to render the geometry. Make sure to test visual aspects of the WebVR experience without this extension present (which should normally be fine because would likely test your WebVR experience on a device that fully supports it; not on a desktop). This plugin allows you to emulate the WebVR device translation and rotational aspects on a desktop. For example:



Using the Web VR Chrome Extension



Virtual Reality via WebVR

Let's talk about the technology you need to work with WebVR. You will need a browser and a device that supports it.

Browsers

It's a little tricky, because different versions of the browsers, on different platforms can provide different experiences. There are two places currently listing the WebVR support for browsers. Check out:

- <http://createwebvr.com/>
- <https://webvr.rocks/>

The first thing you will notice is that the listings are not that consistent to even up-to-date. What about Apple Safari on iOS 11 for example? Not supported. And if you are using Desktop browser, we suggest you to try with Firefox, it's more stable according to our test currently.

WebVR Devices

There are a large number of devices that can support WebVR, here is the main list:

- Google Cardboard (widest variety of devices)
- Google Daydream
- Samsung VR Gear
- HTC Vive
- Oculus Rift
- Others: Windows Mixed Reality

Google Cardboard

Probably the largest variety of devices are around the Google Cardboard. Not only has cardboard been an easy to access VR technology, but it is also supported on a large variety of devices. The cardboards themselves range from simple-plain brown cardboard material, to very colorful, and highly marketable viewers.

For example, these are the ones I found at my house... Note none of these are the Forge branded viewers.



Cats are always popular:



Or what about custom-made?



Google Cardboard Device requirements are very simple:

- support WebGL
- have proper Sensors



Google Daydream

With Daydream, Google has changed the story from being very available, to only being supported on certain devices, including their own Pixel phone. See this link for the latest phones that are supported by Daydream: <https://vr.google.com/daydream/smartphonevr/phones/>

Google Cardboard vs. Daydream

Hardware and Software

- Cardboard... no real “current” requirements except sensors
 - Gyroscope, accelerometer, compass
 - Android 4.1, iOS 8, Windows Phone 10.
- Daydream...
 - More specific: <https://vr.google.com/daydream/smartphonevr/phones/>

Material

- Cardboard... well, made of cardboard
 - OEM-able
- Daydream... better material, with cloth that can be removed/washed
 - Currently, only google sells the device

Communication

- Cardboard... a washer/magnet
- Daydream... NFC

Controllers

- Cardboard... a washer/magnet, maybe Bluetooth if you are brave.
- Daydream... A specially designed controller

Samsung Gear VR and 360 Camera

The Samsung 360 Camera is useful if you want to shoot 360 spherical photos, or 360 videos.

The Samsung Gear VR is built by Oculus. It is a step-up over google cardboard and has its own controller. Samsung provides a specialized app development platform, as well. If this is your primary platform, your options are better when using the VR SDK and building a native app. However, it supports WebVR, and could easily be included in your web based WebVR experience.

The typical WebVR requirement is to use Samsung Browser, included with the phones. It is also only supported on certain devices, and includes: Galaxy S8, S8+, S7, S7 edge, Note5, S6 edge+, S6, S6 edge.

For details, see:

- <https://www.samsung.com/us/mobile/virtual-reality/>
- <https://www.oculus.com/gear-vr/>



Other WebVR Devices

The Oculus Rift, HTC Vive also support WebVR. This might be a good choice if you are targeting a higher end experience. You probably would not target these platforms unless you know your customer base has the device, or perhaps you are building something for in-house use.

WebVR Controllers

Most simple WebVR experiences will not use a controller, especially if they are running in the Google Cardboard environment. The higher-end systems, including HTC Vive and Oculus Rift include their own controllers. Typically these are 6 degree of freedom devices and provide many options for navigation.

The Samsung Gear VR and Google Daydream also provide controllers and are typically referred to as 3 degree of freedom, or game pads. The pad itself is like a joystick.

There are also numerous Bluetooth devices out that that can be connected to browser. For example, even the xbox one controller comes in a Bluetooth version.

But for any of the devices, you will have to code the controller yourself. WebVR does not provide any default controller behavior.

See the WebVR API specification for controllers for details:

- https://developer.mozilla.org/en-US/docs/Web/API/WebVR_API/Using_VR_controllers_with_WebVR

Two interesting libraries that provide easier connectivity are for example:

- <https://github.com/stewdio/THREE.VRController>
- <https://github.com/borismus/ray-input>

Time for another demo!

This interesting demo was built by wall street journal to demonstrate data in a 3d world way. It demonstrates the NASDAQ data as a virtual roller coaster!

- three.js: <http://graphics.wsj.com/3d-nasdaq/>

JavaScript API

Let's dig into some code! We will discuss four different frameworks that support build websites for WebVR.

- three.js
- A-Frame
- React VR
- Autodesk Forge Viewer



three.js

three.js is a very well-known graphics library using WebGL. It provides a WebVR extension, that allows you to easily take any three.js code and bring a WebVR experience to it. three.js is opensource, and assets can be imported from several different formats. It is very flexible and provides a lot of capability and power for 3D experiences. The drawback, is that it can potentially be a lot of code to build the experience you want.

Demo from CodePen.io: <https://codepen.io/gonzomustang/pen/LOEXzi>

Another Code sample: <https://borismus.github.io/webvr-boilerplate/>

Reference: <https://threejs.org/>

A-Frame

Originally from Mozilla, A-Frame is a web framework specifically designed for building virtual reality (VR) experiences. A-Frame is also opensource, and again assets can be imported from several different formats. Although it is a JavaScript library, it is based on HTML, so your coding is in a HTML format.

Demo from CodePen.io: <https://codepen.io/gonzomustang/pen/wPvjJz>

Another Code Sample: <https://ngokevin.github.io/kframe/scenes/aincraft/>

Reference: <https://aframe.io/>

React VR

ReactVR is powered by Facebook with Oculus behind them. React VR is now open source. Again, assets can be directly loaded/imported. ReactVR is the natural platform of choice if you already are working with and know React already. It also works very well to mix 2D 360 content with 3D content.

Easy getting started: <http://facebook.github.io/react-vr/index.html>

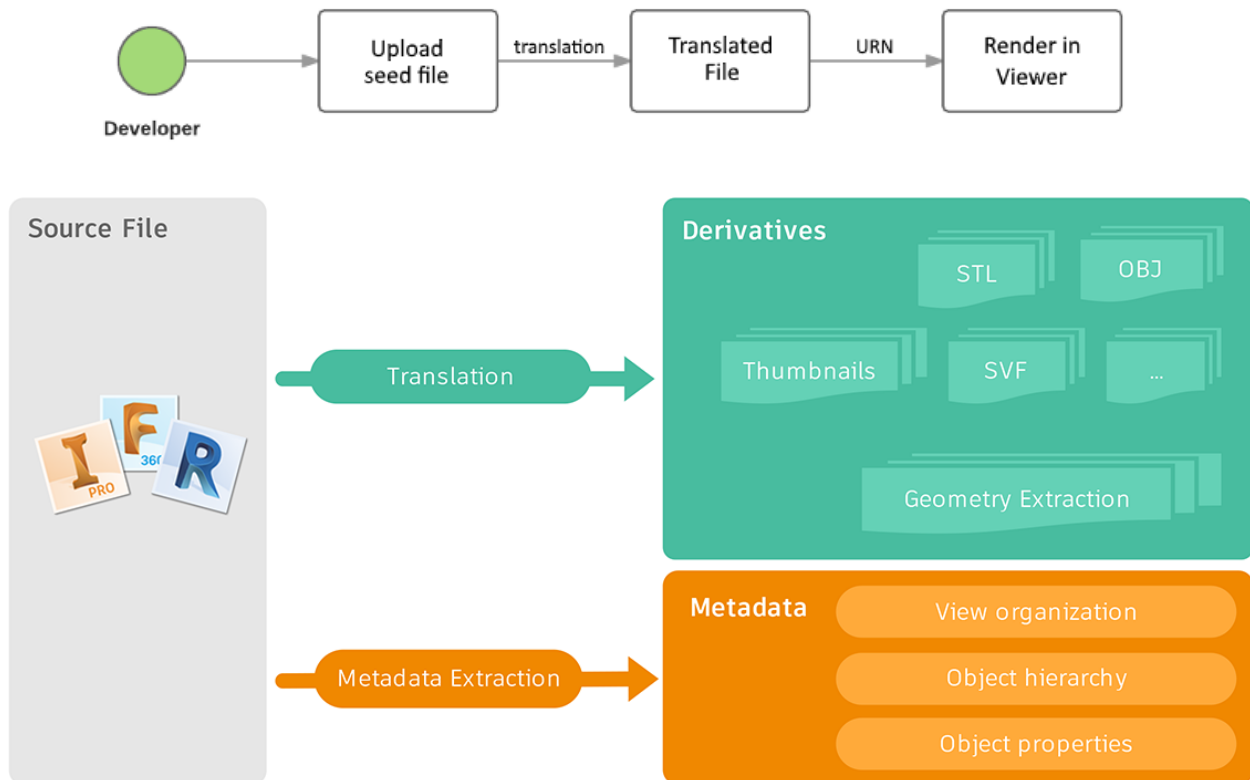
Demo: <https://ironmanviewer.herokuapp.com/>

Code Sample: <https://github.com/JohnOnSoftware/ironmanviewer>

Autodesk Forge Viewer

The Autodesk Forge Viewer is a JavaScript library that can easily be hosted in a website. It uses three.js, so can mix three.js code with the viewer. The assets are via the Forge Model Derivative SVF format. The viewer provides many 3D navigation, viewing and data aspects, and includes meta data from the original asset. It is also capable of viewing very large models/data sets. These built-in features make it potentially small amount of code for you to write. The viewer supports WebVR through an extension, so it is very easy to enable as desired.

Model Derivative SVF



Demo from CodePen.io: <https://codepen.io/gonzomustang/pen/WXwNdz>

Another Code sample: <https://github.com/KeanW/Vrok-It>

Another Demo: www.vrok.it/v2

Authoring Solutions

If you prefer to minimize or eliminate any coding, there are numerous solutions for that as well. Some examples include:

General:

- Autodesk Play: <https://play.autodesk.com>
- Vizor: <https://vizor.io/>
- Insta VR: <http://www.instavr.co/>



What about Augmented Reality (AR)?

Currently there are several frameworks coming from various sources. However, a common ground has not yet been reached. For example, there is nothing yet like a “WebAR” general solution. The nature of AR is to be able to mix the virtual and real worlds, and browsers are not yet there, plus hardware is only just coming out to provide a good experience in this space. For more information, see:

- ARkit from Apple: <https://developer.apple.com/arkit/>
- ARCore from Google: <https://developers.google.com/ar/>

A cool mainstream experience (aside from Pokémon go, of course) I believe is the Merge Cube. See: <https://miniverse.io/cube>. Get yours now before Christmas sell-out!

Resources

WebVR Info: <https://webvr.info>

Create WebVR: <https://createwebvr.com>

WebVR Rocks: <https://webvr.rocks>

ThreeJS: <https://threejs.org>

A-Frame: <https://aframe.io>

React VR: <https://facebook.github.io/react-vr/>

Autodesk Forge Viewer: <https://developer.autodesk.com>

WebVR Slack @<https://webvr.slack.com>

Samples and Demos that use WebVR

<https://experiments.withgoogle.com/webvr>

<https://mozvr.com/webvr-demos/>

<https://webvr.info/samples/>

<https://tutorialzine.com/2017/09/12-exciting-webvr-demos-and-experiments>