What is WebXR?

API designed to access sensors and display of immersive hardware (VR, AR and Mixed Reality).

WebXR identifies what the user is looking at then rendering in WebGL (any other rendering library is compatible but for this project I have chosen OpenGL).

WebXR aims to support existing devices but also paving a way for future devices.

For the project the focus will be on Virtual Reality with an Oculus Rift as the main device which WebXR will access for sensors. (I also have access to a HTC Vive which could

be interesting for compatibility testing.)

An alternate path is the use of smartphones for virtual reality. Devices exist where a user could place their phone inside the headset, this is a popular alternative due to the

demand for powerful hardware that traditional virtual reality headsets required.

AR/Mixed Reality instead overlay the digital content over the surrounding areas, these are a more expensive option than the virtual reality headsets but are an interesting

path.

WebXR developed as a solution to the problem of WebVR only supporting traditional headset virtual reality which do not encompass the entire field. WebVR was deprecated and

now has been replaced with WebXR.

https://mixedreality.mozilla.org/hello-webxr/ <-- This site can demonstrate the capabilities of the WebXR technologies and gives ideas to the kind of things the project could

incorporate (see: xylophone example), also note that even without a headset connected the site falls back to a traditional webpage.

WebXR's device API presents two features to help interaction with the virtual reality

The first is the WebXR gamepad api, this is similar to a normal gamepad api present in the browser (for xbox controllers and such.)

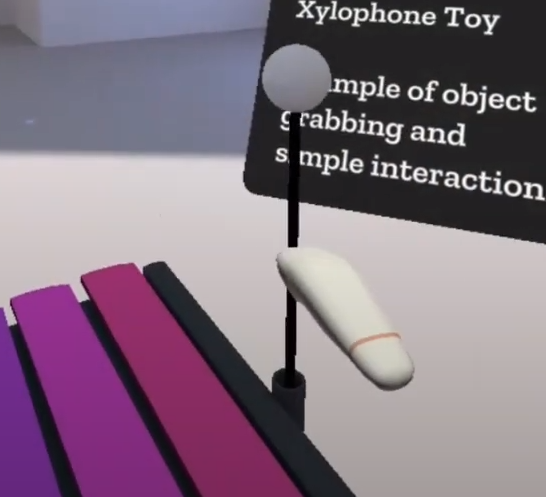
WebXR is only accessible from the WebXR api, so there is no conflict with the existing gamepad api.

The second is a large library which is composed of 3 main parts.

3D Models of virtual reality controller hardware provided by the manufacturers.

JSON Files mapped to the input of these controllers and how their buttons map from the gamepad api to the 3d models.

Library which makes the connection.

Hardware manufacturers are able to update their products on the library.

See above, the left side shows the default controller representation while the right side shows the controller represented with the library.

WebXR is WebGL based api

* This means that you cannot use html or css to create and display things such as labels

WebGL has quite few libraries

* Three.js
* A-Frame
  + Describing scenes using html
  + Allows for declaring 3d objects using html
  + Wrapper around three.js
* Babylon.js
  + Maintained by microsoft
* React360
  + Abstraction over javascript library using REACT
* PLAYCANVAS

DOM overlay api allows for taking a html element into the WebXR api, this is a way to work around the WebGL only restriction.