Architecture of the Immersive Web Platform: Where would accessibility fit in?

In order to understand the components of WebXR, it may be helpful to see the way it has evolved out of the 2d web platform.

Early CSS 3D Transform Experiments

- Early experiments, not enabled by default
- No tracked controllers
- 3dof only (DOF: Degrees of Freedom)
- Could make any DOM element the “root” of the VR scene
- Allowed DOM elements to be rendered twice at slightly different positions
- Allowed CSS content on flat planes in 3d space
- Fixed to 60hz framerate
- Not scalable
- Can only render a few flat planes and WebGL content is juddery

WebVR

- Shipped in multiple browsers
- 3dof and 6dof headsets
- Controllers enumerated as gamepads
- Extra capabilities exposed through the “Gamepad Extensions API”
    - Haptic Feedback
    - Touch sensitive pads and buttons
    - 3dof and 6dof poses in space
- Position of headset and parameters exposed to Javascript
- Head position from API can drive the WebAudio API
- Visuals limited only by creator’s imagination and the GPU performance, but are limited to WebGL.
- Loses semantic markup
- Limited to VR only

WebXR

- Shipping soon in multiple browsers
- Will deprecate WebVR

- Controllers are gamepad objects, but enumerated separately
- Introduces XRInputSource
    - Gives the browser a means to inform content how a basic “Primary Action” can be performed.
    - Works like a laser pointer, that can be gaze-based, emitted from a positionally tracked controller, or selected from a 2d screen surface.
    - Browser is involved in emitting the XRInputSourceEvent, but is not aware of the objects being hit by the ray.
- Will be used for AR as well.
- Still limited to WebGL initially.
- Designed with extensibility in mind, adopting a “module” system like CSS

Semantic Immersive Web?

- Summary…
    - Early web was semantic
    - Javascript added client side interactivity
    - Was still limited by CSS attributes decided by browser vendors
    - WebGL allows free-form addressable pixels, limitless expression, but is non-semantic
    - CSS 3d transforms as specced are limited in scope and scalability, leaving the immersive web dependent on WebGL
- There is demand for semantic markup, returning to the core principles of the web.
- Some accessibility improvements are easier with semantic markup describing scenes and interactivity.
- Semantic markup also increases privacy
- It is difficult to make a semantic spec that everyone can agree on. The challenge is beyond syntax -- rendering intents can vary. Creative expression will be limited and innovation will be bottlenecked by researchers in browser companies. UX of VR and AR is still in the “DOS-With-Graphics” era.
- Innovation is occurring rapidly in the library space with frameworks like Threejs, Aframe, Sumerian, and Mozilla’s ECSY.
- These frameworks are effectively creating the semantic immersive web.
- Non-semantic, WebGL, based rendering won’t go away.
- As common patterns and features emerge, they can be baked deeper into the platform, making accessibility standard.

Let’s Improve It!

- We can immediately act without waiting on the semantic web…
    - Many things are actionable within the WebXR spec, the browsers themselves, and in JS frameworks.
    - Augment the WebGL rendering in libraries with semantic annotation, perhaps with a parallel dom tree for accessibility.
    - Browsers’ own UI should be accessible with more extensive user customization.
        - Text input
        - Filtering controller poses

        - Processing audio output
        - Modular / pluggable systems for customization
    - Built-in browser functionality, such as 360 video playback should be able to present subtitles and have accessible transport controls.
    - Browser-implemented features can enable existing / unmodified WebXR content to be more accessible.
    - Specifications such as WebXR should be written in a more inclusive manner, recognizing the great potential of immersive content that can be experienced in more diverse ways.
    - JS Frameworks can be enhanced to make the best practices default and easy for content creators.

- We have a lot to learn from each other
