Development

Developing Immersive Applications

Created by: Chek Tien TAN

Learning Objectives:

- describe common tools for developing immersive applications
- differentiate WebXR and OpenXR standards
- setup programming environment for building WebXR applications
- implement a minimal WebXR scene with Babylon.js
- explain how changes to HMD hardware parameters (focal length, IPD, eye relief) affect rendered output through hands-on experimentation
- identify the causes of common visual artifacts (distortion, clipping, stereo mismatch) by manipulating HMD parameters

- analyze the relationship between optical parameters and rendering pipeline implementation
- trace how hardware specifications translate to software rendering matrices in the HMD Simulator codebase

Key APIs

API	Type	Platform	XR-Specific
WebXR	Open Standard	Web	Yes
OpenXR	Open Standard	Native	Yes
Vulkan	Open Standard	Native	No
WebGL	Open Standard	Web	No
OpenGL	Open Standard	Native	No
DirectX	Proprietary	Native (Windows)	No
Metal	Proprietary	Native (Apple)	No

Key Points:

- **WebXR** is the web standard for VR/AR — browser-based, no installation required
- **OpenXR** is the cross-platform native standard for VR/AR
- **Vulkan** provides low-level GPU control, best performance for complex apps
- **WebGL** is the foundational web graphics API, WebXR builds on top of it
- Use WebXR for maximum accessibility; OpenXR for cutting-edge native features

Development Frameworks

Framework	Type	WebXR	Key Features
Unity	Native Engine	Partial	Industry standard, C#, extensive asset store
Unreal Engine	Native Engine	Partial	High-fidelity graphics, Blueprint scripting
Babylon.js	Web Framework	Full	TypeScript, WebXR helpers, used in this module
Three.js	Web Framework	Manual	Most popular, flexible, manual setup
A-Frame	Web Framework	Full	HTML-based, most accessible for beginners
PlayCanvas	Web Framework	Full	Web editor, collaborative

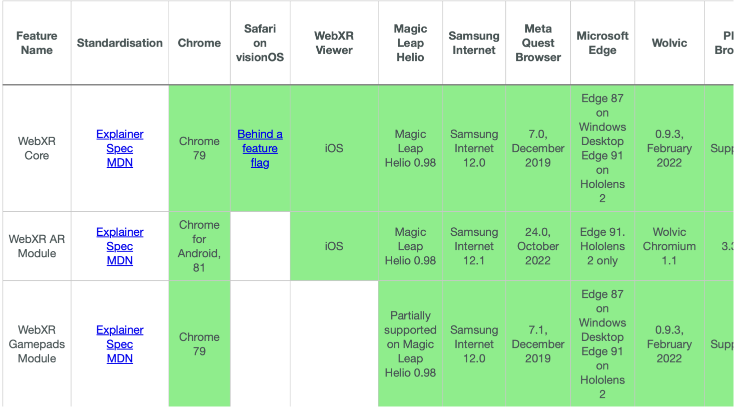
Development Frameworks

Framework	Type	WebXR	Key Features
CoSpaces	No-Code Tool	N/A	Educational, drag-and-drop
Spatial	No-Code Tool	N/A	AR/VR creation without coding

Choosing a Framework:

- **Maximum accessibility?** Use WebXR frameworks (Babylon.js, Three.js, A-Frame)
- **Novel hardware interaction?** Use OpenXR SDK in C++
- **High visual fidelity, have budget/time?** Use Unity or Unreal Engine
- **No coding experience?** Use CoSpaces or Spatial

WebXR Support Across Browsers



Feature Name	Standardisation	Chrome	Safari on visionOS	WebXR Viewer	Magic Leap Helio	Samsung Internet	Meta Quest Browser	Microsoft Edge	Wolvic	Pl Bro
WebXR Core	Explainer Spec MDN	Chrome 79	Behind a feature flag	iOS	Magic Leap Helio 0.98	Samsung Internet 12.0	7.0, December 2019	Edge 87 on Windows Desktop Edge 91 on Hololens 2	0.9.3, February 2022	Supp
WebXR AR Module	Explainer Spec MDN	Chrome for Android, 81		iOS	Magic Leap Helio 0.98	Samsung Internet 12.1	24.0, October 2022	Edge 91. Hololens 2 only	Wolvic Chromium 1.1	3.3
WebXR Gamepads Module	Explainer Spec MDN	Chrome 79			Partially supported on Magic Leap Helio 0.98	Samsung Internet 12.0	7.1, December 2019	Edge 87 on Windows Desktop Edge 91 on Hololens 2	0.9.3, February 2022	Supp

https://immersiveweb.dev