Once get models initialized and understood, maybe see if somehow you can make a box or something that makes it so part of model drawn through is the only one visible or deleted

**Stereokit** – MIT liscense, open source

Supports headsets OpenXR supports

Can use **Visual Studio**, **VS Code**, **or C/C++** (Native)

* Open source XR library for building AR and VR applications with C# and OpenXR
* Dependencies Include
  + **OpenXR**
  + [ReactPhysics3D](https://www.reactphysics3d.com/) - physics
    - Release under open-source Zlib license
  + [Cgltf](https://github.com/jkuhlmann/cgltf) – Gltf format support
  + [Meshoptimizer](https://github.com/zeux/meshoptimizer) – Gltf decompression and mesh optimization
    - MIT liscense
  + [Basis\_universal](https://github.com/BinomialLLC/basis_universal) – texture compression support
  + [Sean Barret’s stb libraries](https://github.com/nothings/stb) – Image and font format support
  + [QOI](https://github.com/phoboslab/qoi) – Image format support
    - MIT liscense
  + [Miniaudio](https://github.com/dr-soft/miniaudio) – Audio playback
    - Choice of public domain or MIT No Attribution
  + [Sokol\_time](https://github.com/floooh/sokol) – Cross platform high performance time
    - Zlib liscense

Can use just OpenXR/native but requires more work as your building project from scratch

* Might take me longer to learn

Engines Integrating OpenXR

* Epic Unreal Engine: 4.45 – pres
* Unity: 2020 LTS – pres
* Chrome & edge – webXR
  + webXR – supports as default backend
* Godot: Core 4.0 – pres
* Blender: VR scene graph inspection
* Autodesk: VRED & stereokit
* NVIDIA: Omniverse

Supported platforms for Meta Quest 3

Most can use android studio

* Godot
  + Free open source
  + [License – Godot Engine](https://godotengine.org/license/)
* Unity
  + Most likely use unity industry or enterprise license
  + Industry 450/mo per seat (seat is each person added to the organization on unity Cloud)
  + Enterprise is customized price
  + Use OpenXR extension when developing for headset
* Meta Horizon Worlds
* Unreal Engine
  + Most likely use seat-based pricing
    - 1,850 per seat per year
* **Meta Spatial SDK** (native android framework)
* **OpenXR**
  + Lets you write low level code accessible to a majority of headsets
  + Uses android developer
  + Need to decide platform and graphics API to be used
    - Android
      * OpenGL ES – high performance 2d and 3d graphics with open graphic library
      * Vulkan
    - Linux
      * OpenGL
  + [OpenXR Tutorial — OpenXR Tutorial documentation](https://openxr-tutorial.com/) see link for detailed tutorials
* WebXR
  + typically use renderers and frameworks already supported for WebXR <https://immersiveweb.dev/#gettingstartedbuildingawebxrwebsite>
  + seems to be able to work with the api directly <https://immersive-web.github.io/webxr-samples/>
* Android app developer

Meta Quest 3 hardware

* Around 2-3 hours of battery life

Meta website has tutorials for both the meta spatial sdk and native/OpenXR sdk if just focusing on development for the meta quest is alright

# Meta Horizon develop website

Tutorials and documentation available on the Meta Horizon website

Mixed Reality resources

* Passthrough
  + Blend virtual content with the physical world and enable people to see their surroundings in rich color.
* Scene
  + Scene empowers you to quickly build complex and scene-aware experiences with rich interactions in the user’s physical environment.
* Spatial Anchors
  + Place and persist virtual content in physical space with realistic depth and occlusion.
* Depth API/Occlusion
  + Render virtual objects so they become occluded by physical objects and surfaces in the real world and appear realistically embedded within their environment.
* Interaction SDK/Hand Tracking
  + Enable the ability to use hands as a controller and switch between controller and hand inputs.
* Voice SDK
* Audio SDK
* Body Tracking
* Native and OpenXR
  + OpenXR Mobile SDK
    - resources for native development and third-party engine integration on Meta Devices that use android
  + Uses Android studio
  + OpenXR Meta Extensions
    - Passthrough
    - Depth
      * Provides real time depth maps, allows virtual objects to be occluded by real world objects
    - Movement
      * Body, Face, and Eye Tracking
    - Scene
      * Enables one to build environments that can interact with the real world
      * Space Setup
        + System flow that lets users walk around and capture scene to generate scene model
      * Scene Model
        + Up to date representation of real physical world that’s easy to index and query.
        + Provides geometric and semantic representation of users space to build room scale mixed reality experiences
    - [Spatial Anchors](https://developers.meta.com/horizon/documentation/native/android/openxr-spatial-anchors-overview)
      * World locked frame of reference that gives position and orientation to a virtual object in the real world
    - Space Sharing
    - Colocation Discovery
      * Allows you to discover nearby users
    - Virtual Keyboard
    - Dynamic Object Tracking
      * Enables device to begin seeing real world objects as they move in real-time, other than hands or controllers
    - Haptic feedback
    - Micro gestures
    - Bicubic Filtering with OpenXR
    - Composition Layer Filtering
    - Application Spacewarp
* Web
  + Browser
    - Built in web browser for meta quest devices
  + WebXR
    - Allows you to develop VR experiences on the web
    - Mixed reality is supported
  + Progressive Web Apps
  + Web Tasks
* Meta Spatial SDK
  + Uses android studio with meta API’s
  + Kotlin based
  + Capabilities
    - Mixed Reality
    - Realistic 3D graphics
    - Complete scene composition
    - Interactive panels, UI
  + Can use meta extensions
* Android apps
* Unity
  + See above for pricing
* Unreal
  + See above for pricing