OpenDisplayXR Project Update

Overview

The creation of accessible immersive stereo 3D technology is an R&D topic that I've been very passionate about for the last ~14 years.

This objective has carried me along the journey of developing countless plugins, scripts, examples, and technical guides that are used by fulldome planetariums, 360VR filmmakers, volumetric video professionals, compositing/pipeline TDs, colourists, and visual effects artists.

OpenDisplayXR

For the last few years I have been involved in a grass-roots research project that is working on improved multi-display rendering support in OpenXR. The current working title for this effort is "OpenDisplayXR".

The longterm goal is to provide a high-level software-agnostic way to extend "VR/XR" centric rendering modes in OpenXR compatible desktop software to power arbitrary display geometries through the use of a virtualized HMD virtual device driver. This approach will make it possible to support the connectivity of immersive theatre environments, virtual production video walls, and stereo monitors to any real-time graphics program that is able to work with a conventional OpenXR HMD like a Meta Quest, HTC Vive, etc.

The OpenDisplayXR project focus is to support active/passive stereo 3D displays, multi-projector setups, CAVES, LED video walls, and fulldome theatres. This approach will hopefully provide a viable cross-platform compatible open-source alternative to proprietary closed-source commercial offerings in this space.

The R&D process started when IMERSA's Daniel Neafus connected me with an NOAA visual simulation expert named Matthew Dougherty. Matthew posted on the OpenXR forums about the goal for multi-display support back in Nov 2021.

Since then I've explored a wide range of relevant tools and technologies like Monado, ImGUI, Vulkan, DirectX, OpenGL, OpenXR, WebXR, OpenVR, OpenComposite, CUDA, OpenCL, GLSL, NewTek NDI, SMPTE ST 2110, Amazon Cloud Digital Interface, OSC, TouchOSC, TouchDesigner, etc.

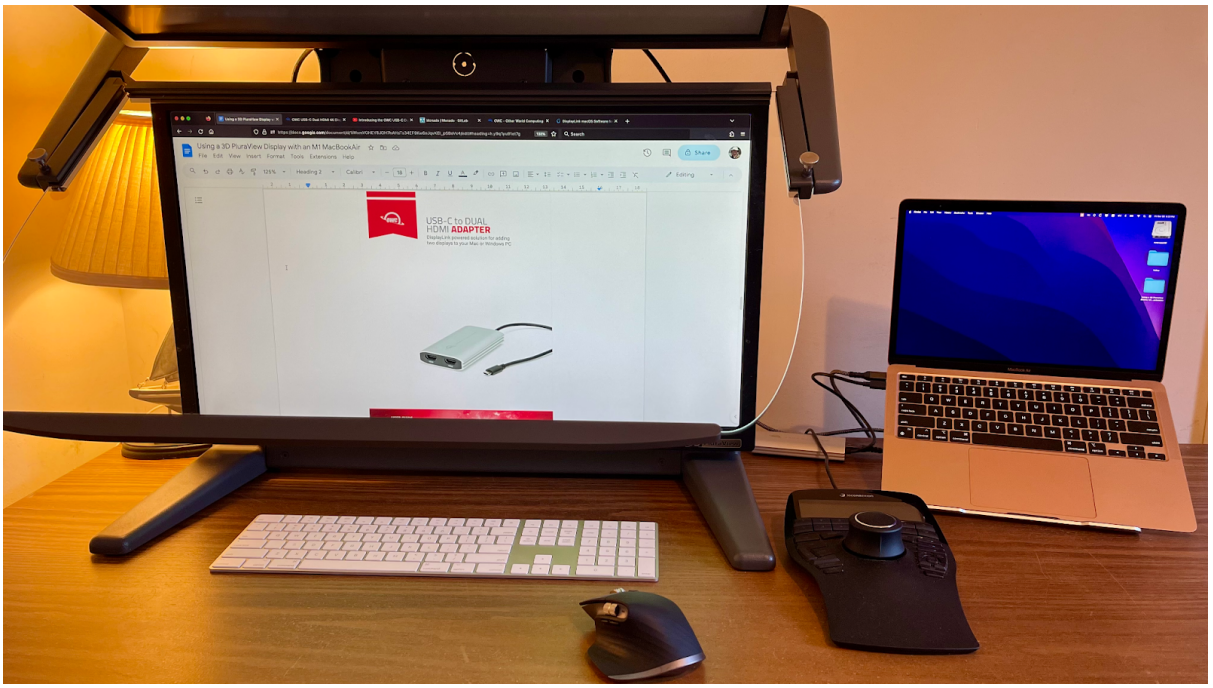
The Prototype

If you would like to follow along with the OpenDisplayXR project development and explore these concepts at home, here is a guide for an initial workflow prototype built around Monado usage with a passive stereo 3D polarizer based display solution.

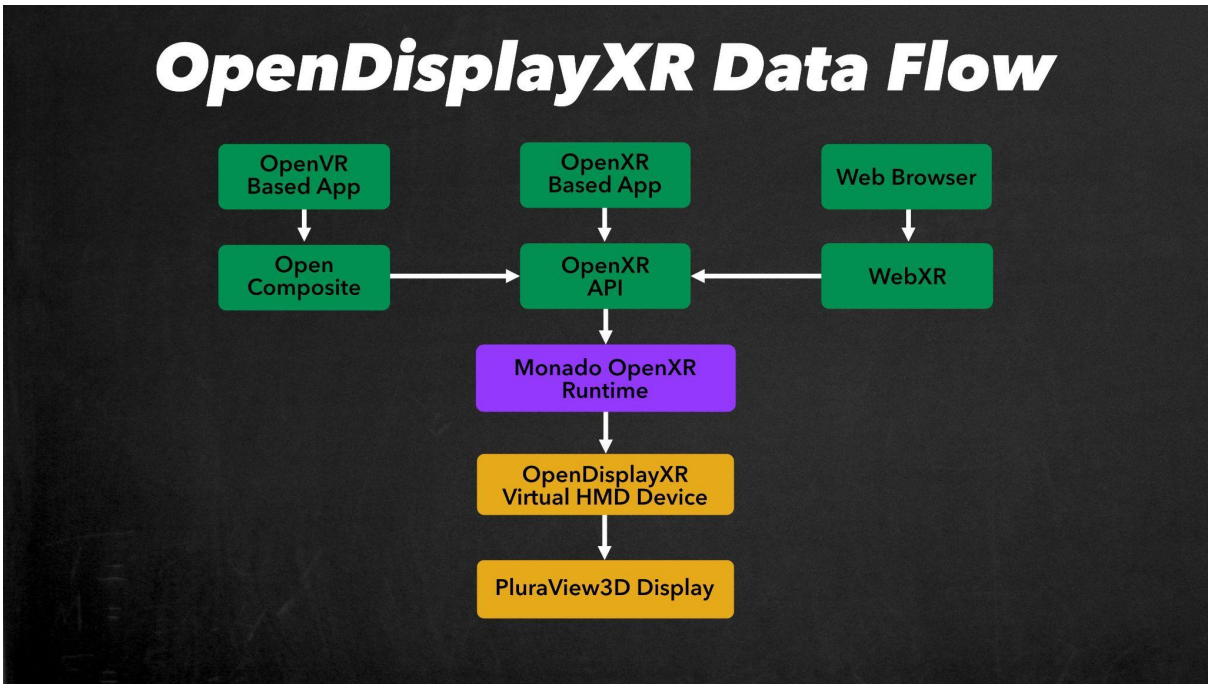
Through the use of the Monado OpenXR framework, a stereoscopic monitor is able to operate as if it is a simulated OpenXR HMD device:

- Monado Based OpenXR Virtual Devices | Windows Guide
- Monado Based OpenXR Virtual Devices | Linux Guide
- Using a 3D PluraView Display with an M1 MacBookAir

The specific monitor used when writing the guide is a beamsplitter based 3D PluraView device that was kindly provided as a gear loan by Josef Schneider of Schneider Digital.



Included below is an illustration of the data flow:

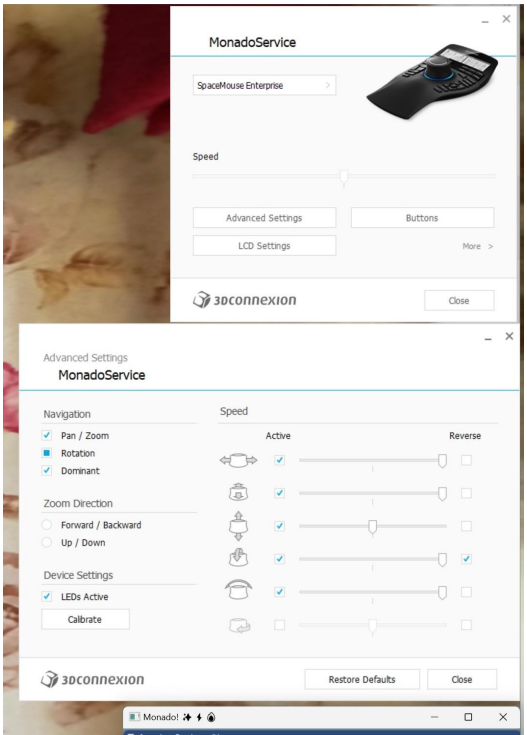


Input Devices

To enable seamless OpenXR based user input control, I created a 3Dconnexion SpaceMouse 3DxWare preset that works with the Monado OpenXR driver software.

The [3Dconnexion SpaceMouse] can be used to control the OpenXR session's camera position and rotation in a very responsive fashion. This works by emulating a head mounted display's 6DOF head tracking motion sensor.

The 3Dconnexion Space Mouse 3DxWare compatible "3DxMonado.3dxz" preset can be downloaded here: https://andrewhazelden.com/projects/kartaverse/downloads/3DxMonado.zip



What’s Next?

The next stage for OpenDisplayXR development is to get a Quadbuffer stereo graphics context working for the monitor output so displays don’t have to operate as extended desktops.

Other longterm goals are to add a new remote driver to allow OSC (Open Sound Control) to provide access to a wide range of user input hardware.

The use of hardware accelerated fragment shaders will allow live image projection transforms to occur by operating on the active framebuffer content. I am also interested in having LuaJIT based scripting support integrated.