Juniper

A Cross-Platform VR and AR Framework

By Sean T. McBeth

Digital Reality Senior Engineer

# Abstract

Virtual and Augmented Reality (VR, AR, or XR combined) are particularly hard fields of software development. They require a broader set of competencies than Web and mobile application development to be able to execute well, and poor execution has deeper consequences to both the business and the users. Coupled with the relative immaturity of tools for developing XR applications with regards to common enterprise application development best practices, XR projects are extremely risky to undertake. A team like Digital Reality that only focuses on XR does not have the luxury of “safe” projects to offset losses in risky bets.

The risks fall into 4 main categories:

* **Burn rate:** XR projects require a broad set of competencies, increasing the weekly burn rate of the project, decreasing the timeline for fixed-budget projects.
* **Platform Churn:** With the rapid pace of change in the XR technology landscape, there is little time to gain expertise on any given platform before it becomes obsolete. We lose work done on older platforms and we run up against new walls on new platforms in the middle of project work.
* **Reimplementing the Wheel:** Platform churn can lead to many assets being very difficult to share between projects. It’s often faster to reimplement features from scratch on the new platform than to try to adapt features from an existing platform.
* **Uncertainty:** Working in cutting edge technologies, on brand new hardware platforms, decreases the certainty of being able to implement requested features, either at all, on time, or on budget.

The Juniper project seeks to address each of these categories through a combination of hardware abstraction and codified best practices. Using Juniper, XR projects require fewer senior practitioners, can be adapted to any existing hardware platform in minutes, can benefit from design and development work of past projects (regardless of their source platform), and can reach MVP faster to validate features and assumptions.

# What is Juniper?

Juniper is a suite of configuration management tools, documentation, source code, graphics, and audio assets for developing XR applications. These components are managed in a source control repository for complete history tracking and traceability.

## Configuration Management Tools

Juniper includes a series of scripts, programs, and Unity Editor operations for managing XR applications in repeatable ways. By using the included tools, developers and designers are quickly guided through many repetitive, mundane tasks for creating high-quality Unity3D projects.

## Documentation

We leverage documentation generation tools to build professional, easily browsable, and searchable documentation.

## Source Code

Juniper leverages the composability of .NET code to provide high-level operations for common application development tasks, with applicability to standard Desktop apps as well as XR applications.

## Graphics

Common user interface elements are included in the Juniper package, to enable rapid iteration with a level of graphics quality above the default Unity3D UI assets.

## Audio

Audio is unfortunately the most neglected part of many projects, as simple interface reaction clips and spatialized audio can greatly enhance the XR experience at a very low cost of development. Juniper includes easily-customizable audio interactions by default, making sure every project has a rich audio component from the very beginning.