VIVE MIXED REALITY PLUGIN

FOR UNREAL
VERSION 0.8 BETA

FOO HOANG / HTC VIVE © 2018 (Original code collaboration with Scott Hedstrom @ Vision Agency)

CHANGELOG

0.8	Initial plugin and documentation release.

INTRODUCTION

This standard Unreal Engine Plugin can be simply added to any Unreal 4.16+ Blueprint/C++ project to enable the Mixed Reality Compositing standard based on SteamVR's Unity implementation of pre-composited quadrant view. This enables any user that's already set up for Mixed Reality to leverage their existing setups to composite your Unreal based experience without any additional effort.

| FOREGROUND (FG)	FG ALPHA CHANNEL
BACKGROUND (BG)	USER VIEW (HMD)

REQUIREMENTS

- UE 4.16+ (tested on 4.16, 4.18)
- Project support for Blueprints & C++

INSTRUCTIONS

1. **Install**
   Place the **ViveMixedReality** folder (the parent folder that this document lives in) into the **Plugins** directory of your project.
   Re-open your project so that it scans the **Plugins** folder correctly.

2. **Enable Plugin**
   Navigate to the file menu item under **Edit > Plugins**.
   Under **Installed** should be **Vive Mixed Reality**.
   Click **Enable**.

3. **Add Hooks**
   You'll need hooks in your Blueprints to add the component, and then enable a way for the user to turn it on and off. The overhead for adding the component is low so performance shouldn't take a hit. Adding it on **Begin Play** on your Level Blueprint should be fine. Then what you'll want to do is add keys to enable/disable the component.

   Ex.
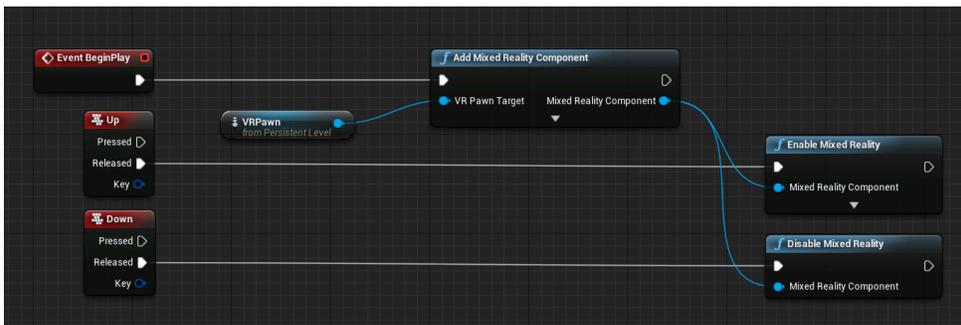
In this example, **Add Mixed Reality Component** is called on **Event BeginPlay**. The instance reference is passed to **Enable Mixed Reality** and **Disable Mixed Reality**. Which are activated only on key release events **Up** and **Down** respectively.

Full information on available Blueprint components below.

TESTING

Verification of the functionality of the plugin can be done by adding an **externalcamera.cfg** file in the package root, the WindowsNoEditor root, or if you're testing in the editor, the project folder root (the same location as your **uproject/sln** files.) An example cfg file can be found in the package root as **externalcamera.cfg.sample**. Just be sure to rename it.

Add and enable the VMR component via the method you used in Blueprints and move the tracker/controller around to verify that the **FG** (top left), **FG Alpha** (top right), **BG** (bottom left) quadrants are following your tracker. Also, move the headset around and be sure that the only quadrant that is updating positionally is the **User View** (bottom right) quadrant.

If you would like to test your experience with full composited output, we recommend checking out the many great resources out there to help with this.

SUPPORT

Please visit Vive Community Forums in the Developer Forums if you have any issues. http://community.vive.com/

BLUEPRINT COMPONENTS
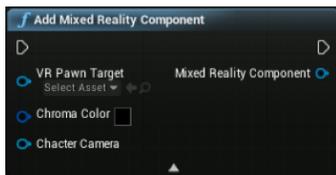
IN

VR Pawn Target
The main VR Actor attached to OpenVR/SteamVR hooks.

Expanded:

Chroma Color
The chroma color for the FG (top-left) quadrant. Generally, you want this to be black so that your transparencies don't grab on to another color.

(Default: #000000)

Character Camera
The gameplay set character camera. Only change this if you really need to.

(Default: Index 0 of the VR Pawn Target World)

OUT

Mixed Reality Component
The reference to the created instance of the VMR component.

IN

Mixed Reality Component
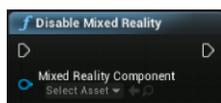A reference to an instance of the VMR component.

Expanded:

Camera Tracker
The SteamVR tracked device to be used as attached to the IRL camera. For example, every time you call enable, you can search for the first attached tracker, or the second attached controller.

(Default: Looks for the last attached tracker and then the third attached controller.)

OUT

IN

Mixed Reality Component
A reference to an instance of the VMR component.

OUT