OVRVoiceMod for
Unity Integration Guide

Version 1.0.0

Copyrights and Trademarks

Contents

Overview

This guide describes how to install and use the Oculus Voice Mod Unity integration with Unity 5 for the Oculus Rift and Gear VR.

The Oculus Voice Mod integration for Unity 5 (OVRVoiceMod) is an add-on plugin and set of scripts used to modify incoming audio signals. OVRVoiceMod analyzes audio input streams from canned sources (dialogue) or a microphone input. It re-generates the audio with modified characteristics, making a person's voice sound like a robot, for example, or changing their voice from male to female, or vice-versa.

OVRVoiceMod is flexible enough to generate many different types of voices. It includes presets to help you get started. Although any sound that passes through OVRVoiceMod can be changed, it is best suited for vocal sounds.

Requirements

OVRVoiceMod requires Unity 5.x Professional or Personal or later, targeting Android or Windows platforms - see *Unity Compatibility and Requirements* for details on our recommended versions. Supports Windows 7 and later, and OS X.

Download and Setup

Note: We recommend removing any previously-imported versions of the OVRVoiceMod integration before importing a new plugin.

To download OVRVoiceMod and import it into a Unity project:

- Download the OVRVoiceMod package from the *Oculus Downloads* page.
- Extract the zip.
- Open your project in the Unity Editor, or create a new project.
- In the Unity Editor, select *Assets* > *Import Package* > *Custom Package…*.
- Select OVRVoiceMod.unitypackage in the OVRVoiceMod folder and import. When the Importing *Package* dialog opens, leave all assets selected and click *Import*.

Exploring Oculus Voice Mod with the Sample Scene

To get started, we recommend opening the supplied demonstration scene VoiceMod_Demo, located under Assets/OVRVoiceMod/Scenes.

This scene provides an introduction to OVRVoiceMod resources and examples of how the library works.

To preview the scene in the Unity Editor Game View:

1. Import and launch VoiceMod_Demo as described above.
2. Play the VoiceMod_Demo scene in *OVRVoiceMod > Scenes* in the Unity Editor Game View.
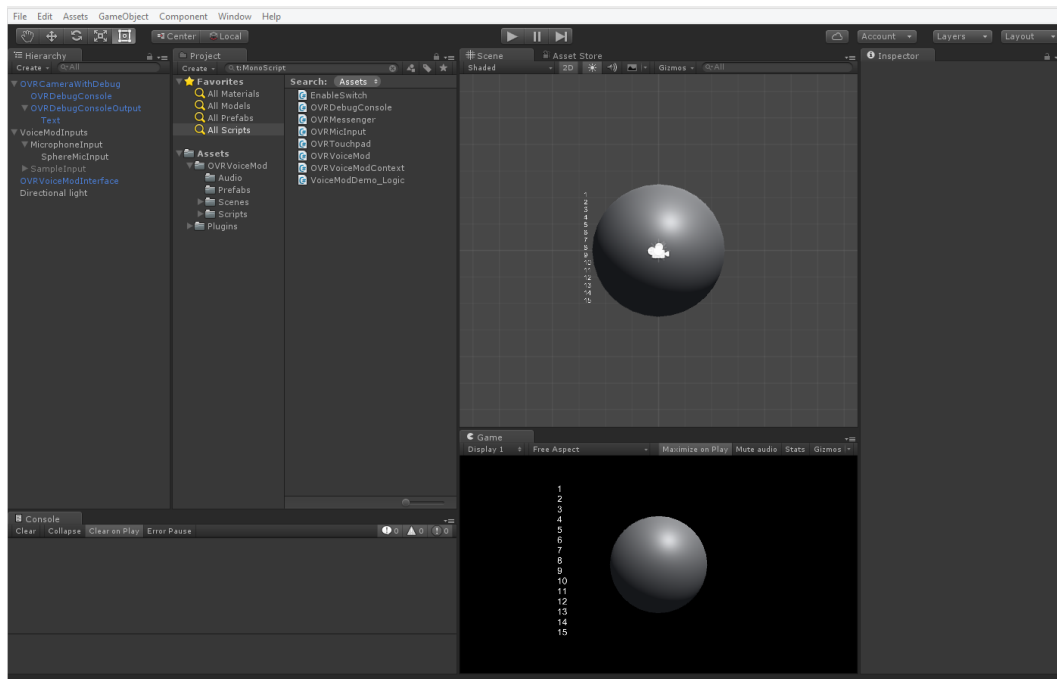
To preview the scene with a Rift:

1. Import and launch VoiceMod_Demo as described above.
2. In *Build Settings*, verify that the *PC, Mac & Linux Standalone* option is selected under *Platform*.
3. In *Player Settings*, select *Virtual Reality Supported*.
4. Preview the scene normally in the Unity Game View.

To preview the scene in Gear VR:

1. Be sure you are able to build and run projects on your Samsung phone (Debug Mode enabled, adb installed, etc.) See the *Mobile SDK Setup Guide* for more information.
2. Import and launch LipSync_Demo as described above.
3. In *Build Settings*

    a. Select *Android* under *Platform*.
    b. Select *Add Current to Scenes* in *Build*.
    c. Set *Texture Compression* to *ASTC* (recommended).
4. In *Player Settings*

     a. Select *Virtual Reality Supported*

     b. Specify the *Bundle Identifier*.

5. Copy your osig to <unity-project>/Assets/Plugins/Android/assets.

6. Build and run your project.

How to use VoiceMod_Demo

You can switch between microphone and our provided sample audio clip using the following controls.

Key	Control
Z	Set audio source to Mic input (default).
X	Set audio source to Sample input.
L or <left mouse click>	Toggle Loopback on/off. If Mic input is selected and you are not using headphones, audio may feed back from speaker to mic, so **be careful**! Default is off.
1 to 0	Various VoiceMod presets. 1 is init (no processing).
<left arrow>	Decrease linear mic./sample gain by 0.1 (min: 0.5).
<right arrow>	Increase linear mic./sample gain by 0.1 (max: 3.0).

Key	Control
Swipe Down	Select next VoiceMod preset.
Swipe Up	Select previous VoiceMod preset.
Swipe Forward / Backward	Cycle forward/backward through targets (Morph target/mic input; Flipbook target/mic input; Morph target/audio clip input; Flipbook target/audio clip input). Audio clip input autoplays.
Single Tap	Toggle mic loopback on/off to hear your voice with the mic input.

Note: Gear VR does not include the ability to adjust microphone/sample gain using the touchpad.

OVRVoiceMod Scripts In More Detail

Script	Description
VoiceModDemo_Logic.cs	This script shows how to interface with the OVRVoiceModContext script and ties the VoiceMod_Demo scene together.
OVRVoiceMod.cs	To use the Voice Mod integration, a scene must include OVRVoiceMod, the main interface to the OVRVoiceMod dll. A prefab is included in the integration for convenience.
OVRVoiceModContext.cs	OVRVoiceModContext must be added to each GameObject which has the AudioSource which you want to modify.

Script	Description
	Note: Any stereo sound passing through OVRVoiceModContext will be collapsed to a single, mono channel. To modify the audio signal, you must send parameters to the context using the function SendParameter. OVRVoiceModContext contains a list of parameters that can be set. Please refer to the script for a description of each parameter.
OVRMicInput	This script is for use with a GameObject which has an AudioSource attached to it. It takes input from any attached microphone and pipes it through the AudioSource. Note that an AudioSource must be available to use the OVRVoiceModContext script, as the system relies on the function OnAudioFilterRead to affect the audio stream.

We suggest experimenting with the different parameters to see what type of output you get. The presets within the script can give an idea of what various parameter sets sound like.

Other OVRVoiceMod Scripts

Script	Description
OVRMicInput	This script is for use with a GameObject which has an AudioSource attached to it. It takes input from any attached microphone and pipes it through the AudioSource. Note that an AudioSource must be available to use the OVRVoiceModContext script, as the system relies on the function OnAudioFilterRead to affect the audio stream.

We recommend looking at the other scripts included with this integration. They will provide more insight as to what is possible with OVRVoiceMod. We include, for example, some helper scripts to facilitate easy on-screen (in VR) debugging, as well as scripts to allow for access to the touchpad on Gear VR.