Oculus in Unity Free

## Jonathan Pearl ( 2014-10-30 )

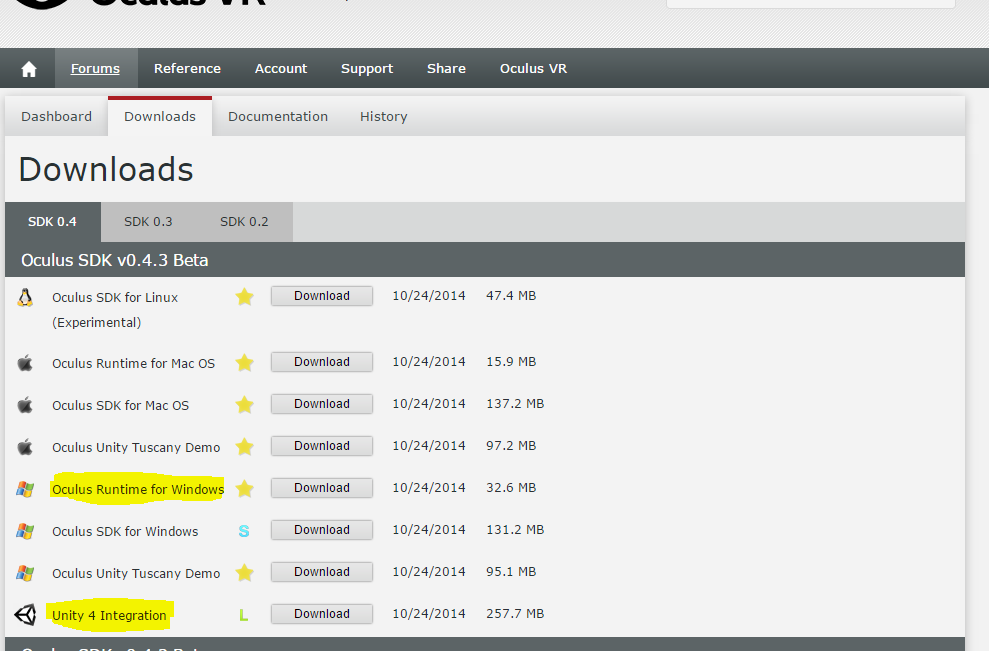
### PREFACE

Given the volatility of Oculus, this tutorial may contain elements which are no longer completely correct. Use your best judgment if problems occur when following these instructions.

### 1 – GETTING UNITY – 1

First thing is first: in order to get Oculus working inside of Unity for free, you need to update Unity. The current stable release ( 4.5.5 at the time of writing ) will work just fine. Feel free to perform the following steps while waiting for Unity to download. Whenever the download is complete, install as you normally would install unity. <http://unity3d.com/unity/download>

### 2 – GETTING THE OCULUS SDK – 2

To develop for the Oculus you will need the Oculus development kit, naturally. Head on over to <https://developer.oculusvr.com/> to create a developer account to get started. Once you have created and confirmed your account you will need to download the following from the latest revision of the SDK ( 0.43 at the time of writing ): Oculus Runtime for Windows and Unity 4 Integration.

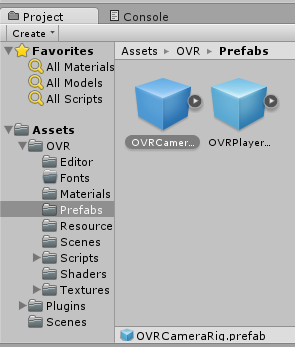
Once your downloads finish, install the Oculus runtime. Connect the Oculus to your computer fully. Once the Oculus is connected, open up the Oculus config utility and calibrate it for your own use.

### 3 – PUTTING THE TWO TOGETHER – 3

Once Unity has installed, you have installed the Oculus Runtime, and the Unity 4 Integration has finished downloading and you have unzipped it, open the Unity 4 Integration folder. Somewhere inside you should find a file along the lines of OculusUnityIntegration.unitypackage ( Not to be confused with their demo packages ). Copy this file into your installation of Unity’s standard package directory. Mine was found at the following, and yours will be located similarly.

D:\Program Files (x86)\Unity4-5-5\Editor\Standard Packages

### 4 – CREATING YOUR PROJECT – 4

Now for the fun stuff – open Unity and create a new project. Feel free to name it whatever you wish and place it wherever you would like. But before you confirm, scroll through the list of packages and find the Oculus package we copied over and check the box to import it. Now confirm. You will notice that there are a number of assets that have been included in your project by OVR. The one you’ll be working very closely to is the prefab found in OVR->Prefabs which is OVRCameraRig, it handles all of the grunt work of communicating with the Oculus as well as orienting the projections for both eyes and finally rendering the two views onto the same texture and applying the distortions needed to counteract the distortions caused by the lenses in the Oculus. Wow, what a mouthful. Feel free to slowly drag that bad boy into your scene. Congratulations, you’ve just added Oculus support to your soon-to-be game!

### 5 – RUNNING THE PROJECT – 5

Throw a few cubes and spheres into the scene away from the rig so you have something to look at when you run the project. To debug run from within Unity, feel free to press play and run the application that way. With this, it won’t display to the Oculus but will display within Unity and should read the information being sent by the Oculus. To view within the Oculus, you will need to build and run the application which will populate a new window with your game. If the application starts on your main display instead of your Oculus, I’ve resolved this by setting the Oculus as my main monitor within the Control Panel resolution change screen so that windows defaultly open inside of it. Someone on Reddit has had luck by changing the visual order of the displays within the resolution screen. Details here: <http://www.reddit.com/r/oculus/comments/2diw2a/figured_out_why_open_on_oculus_rift_wasnt_working/>

### ? – NOW WHAT – ?

Make your game. I recommend referring to documentation to figure out what’s going on behind the scenes in order to understand it better. Here is a script from my tech demo that might inspire how you would go about things like character movement.



