

DPVR Unity Plugin Integration Manual

Content:

Supported Unity3D version

Import the DPVR Unity3D plugin package

Unity3D project settings

3Dof Controller integration

Supported Unity3D version

DPVR Unity plugin is verified to support below Unity versions:

Unity3D 5.3.4

Unity3D 5.4.3

Unity3D 5.5.0

Unity3D 5.6.1

Unity3D 2017

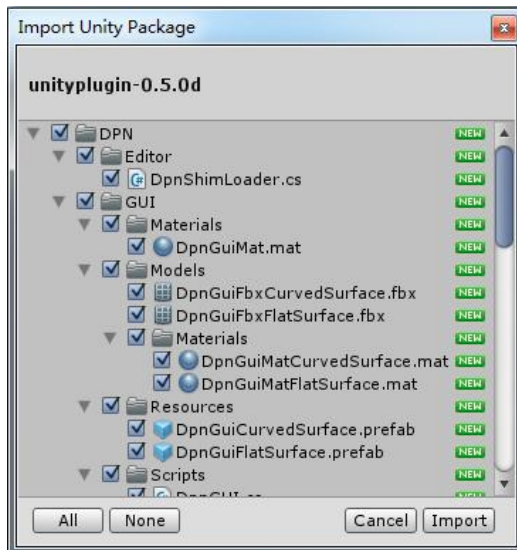
Unity3D 2018

Please try to match with above version, otherwise the plugin might not be working correctly.

Please refer to [My Very First Unity VR App](#) and [Unity Non-VR to VR Porting Tutorial](#) for step by step tutorial.

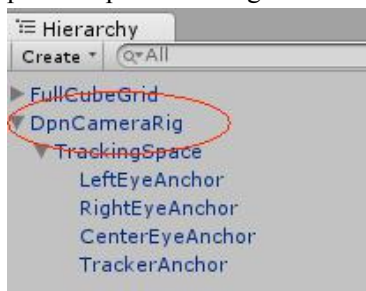
Import the DPVR Unity3D plugin package

Open Unity3D, choose Assets->Import Package->Custom Package. Select the downloaded unitypackage file and click import.



Replace Camera

Delete the original Camera, and use /Assets/DPN/Prefabs/DpnCameraRig.prefab instead. Simply drag the prefab DpnCameraRig to where the original Camera locates.



After replacing Camera, the features associated with the original Camera may require some adjustments. We recommend using Canvas in WorldSpace instead of ScreenSpace.

Modify AndroidManifest.xml (If you are developing on the Android DPVR All-In-One Device)

The AndroidManifest.xml will be replaced by default during plugin-import stage. However, if one want to use self-defined AndroidManifest.xml, below is the changes needed to make on this file to make the VR App running on DPVR All-in-One VR devices correctly.

Please refer to the plugin Assets\Plugins\Android\AndroidManifest.xml.

1) Inherit from class DpvrActivity

If the game inherits from UnityPlayerActivity, please change the inheritance to inherit from com.dpvr.sdk.DpvrActivity; If the game does not inherit from UnityPlayerActivity, please set com.dpvr.sdk.DpvrActivity as the main Activity

2) Add the following meta data to declare this application as a VR application.

```
<meta-data android:name="com.softwinner.vr.mode" android:value="vr"/>
```

3) Add a VrListener statement

for unity plugin 0.7.2d or later

```
<service android:name="com.dpvr.aw.vrsdk.VrListener"
  android:permission="android.permission.BIND_VR_LISTENER_SERVICE">
  <intent-filter>
    <action android:name="android.service.vr.VrListenerService" />
  </intent-filter>
</service>
```

Unity3D project settings

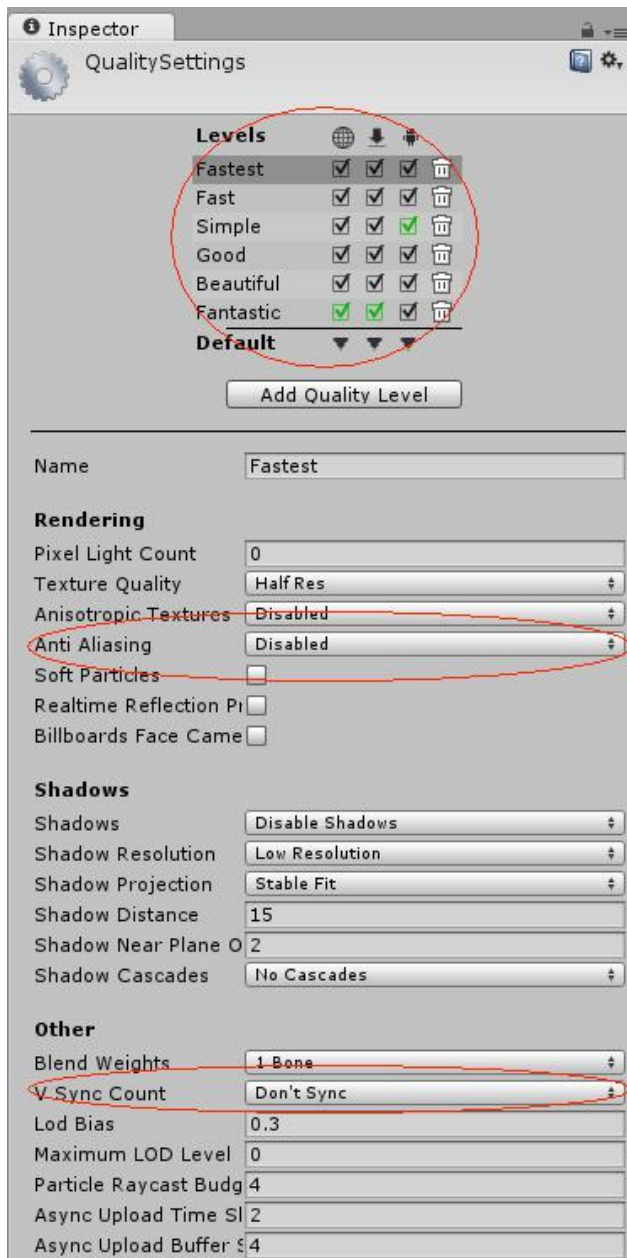
Quality settings

In Edit->Project Settings->Quality.

1) Select "Don't Sync" in the VSync option;

It will be best if the app frame-rate equals the display frame-rate. However, due to the hardware capacity and content complexity, the former frame-rate might be lower. Usually it is acceptable if the app frame-rate is greater than half of the display frame-rate. Just choose "Don't Sync" and DPVR SDK will handle the VSync signal automatically.

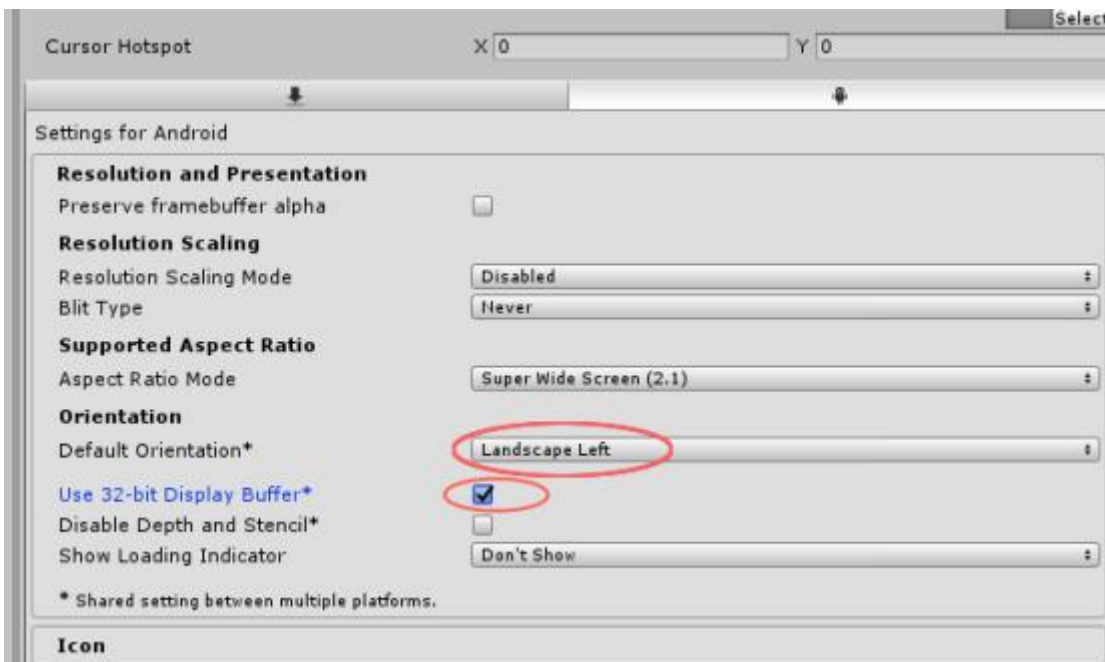
2) If you find the performance too slow, try setting "Rendering"\ "Anti Aliasing" to "Disabled"



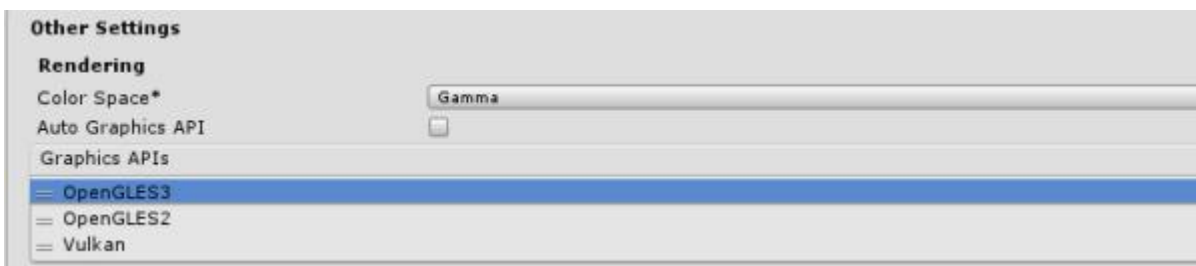
Please be noted that above setting should go through all the levels (from Fastest to Fantastic) so that there will be no missing point.

Player settings (If you are developing on the Android DPVR All-In-One Device)

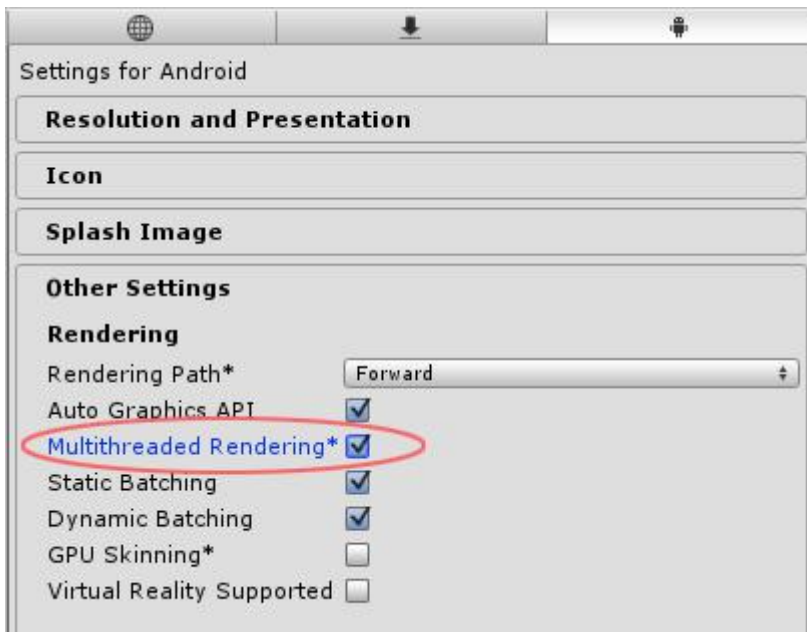
- 1) In Edit->Project Settings->Player->Settings for Android->Resolution and Presentation, please change "Default Orientation" to "Landscape Left";
- 2) In Edit->Project Settings->Player->Settings for Android->Resolution and Presentation, please check "Use 32-bit Display Buffer". (This is a must for P1 & P2 Lite, it is optional for M2/M2 Pro or P2).



- 3) Current DPVRUnity Rendering SDK does not support Vulkan yet. As a result, we recommend check "Auto Graphics API" option in Edit->Project Settings->Player->Settings for Android->Other Settings. Or, OpenGL3 is recommended on top of the Graphics API list.



4) In Edit->Project Settings->Player->Settings for Android->Other Settings, "Multithreaded Rendering" can be checked; However please be caution to turn it on because this option was reported unstable in some Unity versions.

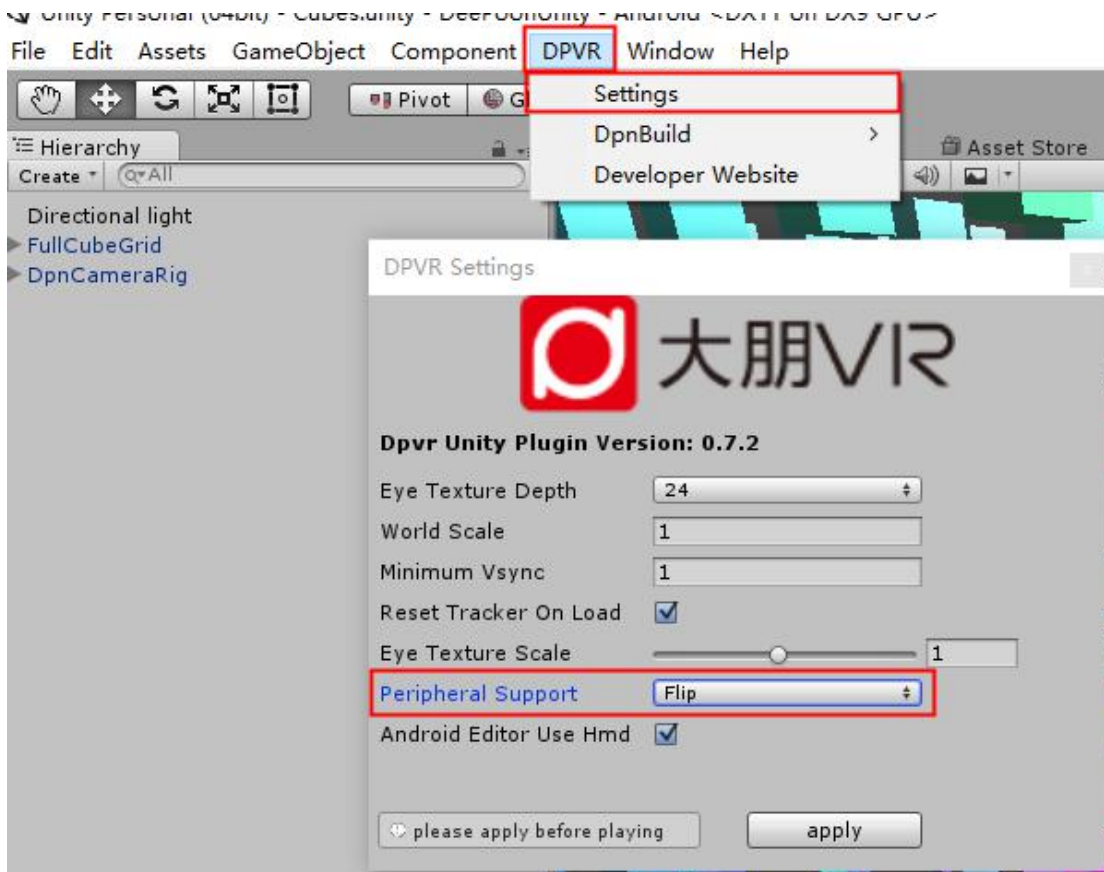


Time setting (Recommended, not a must)

Open Edit->Project Settings->Time and set the "Fixed Timestep" to 0.01.

3Dof Controller

- 1) Import the DPVR unity plugin (version 0.7.0 and above);
- 2) In the menu DPVR->Settings, please set the "Peripheral Support" option to "Flip";
- 3) DpnCameraRig.prefab will automatically generate controller-related prefabs, including code and models.



How to get the controller information:

1) If you want to get the pose and button information of the controller, you can refer to the Daydream controller and use the according properties on the DpnDaydreamController.cs script. For more information, please refer to the comment in the script. Compared to the Daydream controller, DPVR flip controller adds a TRIGGER button, removes the VOLUMEUP and VOLUMEDOWN buttons, changes the APP button to a BACK button and keeps the HOME button. The BACK and HOME button behavior are handled automatically, no application processing is required.

The current system default behavior for the 2 buttons is as follows:

BACK: The function is the same as the BACK key on the HMD;

HOME: Short press returns to launcher, long press recenter calibration;

2) The controller comes with a ray collision algorithm by default. If the menu->DPVR->Settings->Use DPVRPointer option option is checked, DPVR collision algorithm is used (default behavior); Otherwise user defined collision algorithm is taking effect;



3) If you set the "Peripheral Support" option to "None". The app would use HMD control as gaze cursor. However in this case, the 3Dof controller can still work as a controller and report its touch and button event as long as it is paired to the All-in-One HMD. Similarly, if you set `Dpnmanager.DPVRPointer` to true, the DPVR default collision algorithm will take effect;

4) On DPVRunity 0.7.3b and above, the 3Dof controller supports switching between left-hand and right-hand mode in unity plug-in. One can use Interface name:
`DpnDaydreamController.interactiveHand { set, get }`
where 0 represents right-hand mode and 1 represents left-hand mode. Please update to the latest image so as to enable this feature;

5) For UI interaction, On DPVRunity 0.7.6 and above, replace `EventSystem` with `DpnEventSystem` (please refer to [My Very First Unity VR App](#), section 7), or, alternatively, replace the `StandaloneInputModule` component under `EventSystem` with `DpnStandaloneInputModule` (please refer to [Unity Non-VR to VR Porting Tutorial](#), section 9).

Look Around in Editor Mode

In DPVRUnity plugin 0.7.3b or newer version, the plugin provided an auxiliary camera mover script in Editor mode.

Hold down the Alt key, move the mouse around with the right mouse button pressed in Editor window to look around.