

DPVR_Unity_Plugin 集成文档

Content:

支持的 Unity3D 版本

导入大朋 Unity3D 插件包

Unity3D 工程设置

3Dof 手柄适配

支持的 Unity3D 版本

目前大朋的 Unity 插件支持以下 Unity 版本：

Unity3D 5.3

Unity3D 5.4

Unity3D 5.5

Unity3D 5.6

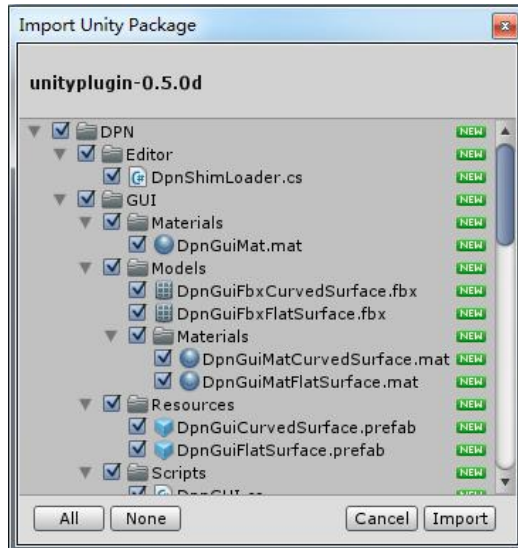
Unity3D 2017

Unity3D 2018

请尽量匹配这些版本，否则可能会出现问題。

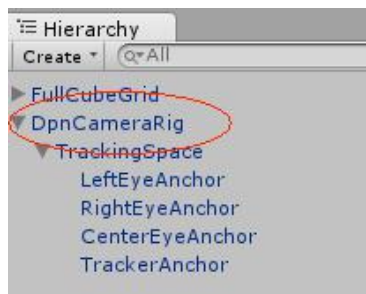
导入大朋 Unity3D 插件包

打开 Unity3D，选择 Assets->Import Package->Custom Package。选中下载好的 Unity3D 插件后导入。



替换 Camera

将场景中原来的 Camera 删除，将/Assets/DPN/Prefabs/DpnCameraRig.prefab 拖到原来 Camera 的位置即可。



替换 Camera 后，与 Camera 相关的逻辑和特效可能需要做一些调整。

在 DPVRunity 0.7.6 及之后版本，UI 互动的逻辑请使用 DpnEventSystem 替换 EventSystem 的方式（请参看[《我的第一个 VR 应用 \(Unity\)》](#)第 7 点内容）或者用 DpnStandaloneInputModule 替换 StandaloneInputModule 的方式（请参看[《现有 Unity3D 游戏的 VR 移植教程》](#)第 9 点内容）的实现示例。

修改 AndroidManifest.xml（安卓一体机开发）

对于插件版本 0.7.0 及后续版本，通常 AndroidManifest.xml 会在插件导入阶段自动覆盖项目的同名文件。开发者无需做特定修改。该版本不使用大朋证书。P1 开发者必须使用该版本开发，M2 和 M2 Pro 开发者如果不需要证书也可使用此版本。

为避免开发者有自定义该文件的需求，将对该文件所需的改动点列出如下：

相关内容请参考插件自带文件 Assets\Plugins\Android\AndroidManifest.xml

1. 继承 DpvrActivity 类

如果游戏继承了 UnityPlayerActivity，请将继承修改为继承 com.dpvr.sdk.DpvrActivity；如果游戏没有继承 UnityPlayerActivity，请将 com.dpvr.sdk.DpvrActivity 设置为主 Activity

2. 增加如下 meta data 来声明此应用为 VR 应用

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

3. 添加 VrListener 声明

```
<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>
```

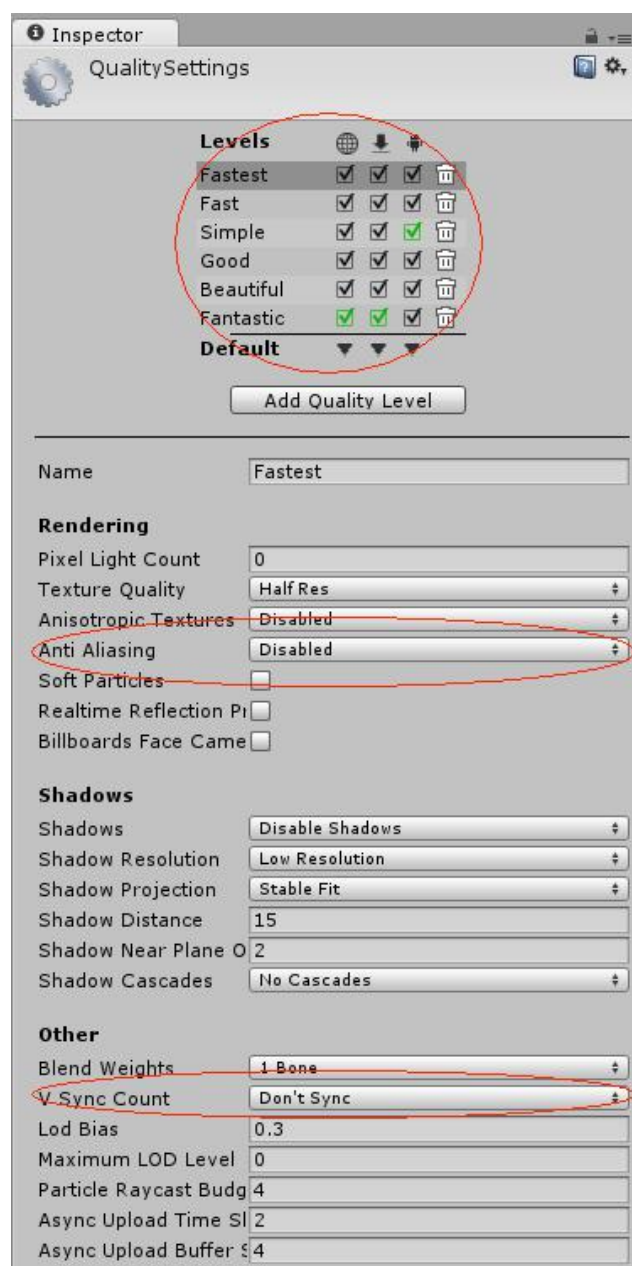
Unity3D 工程设置

Quality 设置

打开 Edit->Project Settings->Quality。

1 将 VSync 设置成“Don't Sync”。应用的帧率与屏幕刷新率相同为最佳，但受到性能影响，往往会低于屏幕刷新率。一般在刷新率的一半以上还是可以接受的。SDK 会自动处理垂直同步，所以这里需要关闭。

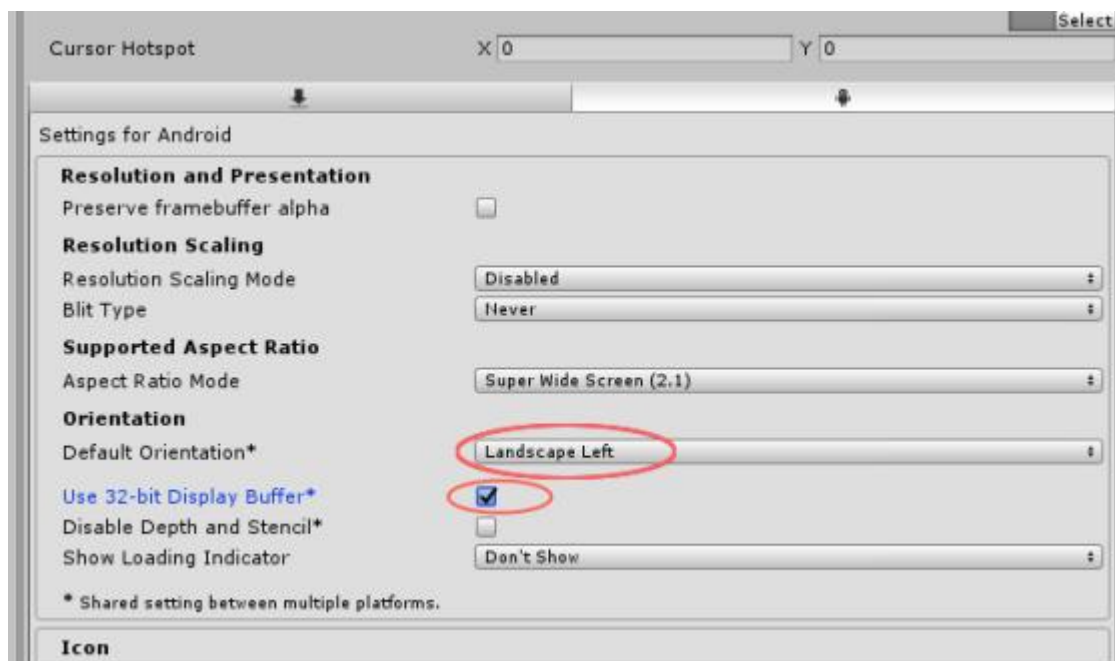
2 将 Anti Aliasing 设置成“Disabled”。SDK 会自动处理抗锯齿，所以这里需要关闭。



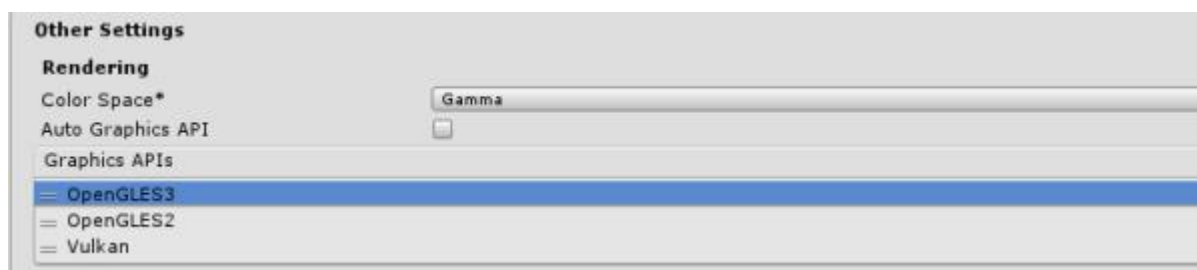
注意所有的 Level 都必须进行一次设置，确保全都按照要求设置。

Player 设置（安卓一体机开发）

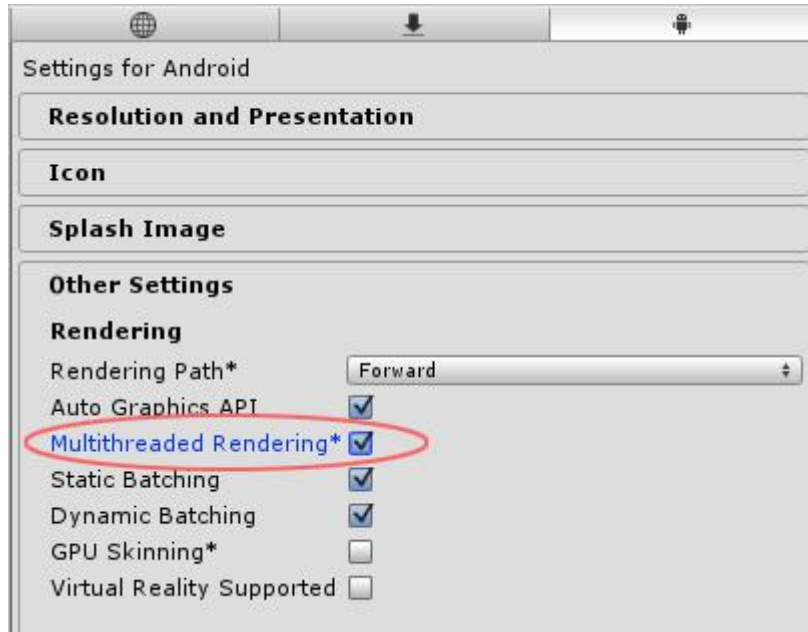
- 1) 在 Edit->Project Settings->Player->Settings for Android->Resolution and Presentation 中，将 "Default Orientation" 改成 "Landscape Left"。
- 2) 在 Edit->Project Settings->Player->Settings for Android->Resolution and Presentation 中，请勾选 "Use 32-bit Display Buffer"。（此项要求仅在 P1、P2 Lite 上是必须的，M2/M2 Pro、P1Pro、P2 上非强制）



- 3) 目前的 DPVRUnity Rendering SDK 不支持 Vulkan 的 Graphics API。因此，在 Edit->Project Settings->Player->Settings for Android->Other Settings 中，可勾选 Auto Graphics API，若不勾选，推荐将 OpenGL ES3 置于 Graphics API 列表顶层（不可将 Vulkan 置于顶层）。



- 4) 在 Edit->Project Settings->Player->Settings for Android->Other Settings 中，可以开启 "Multithreaded Rendering"，这可以提升一定的应用的性能，但打开此选项时请小心，在某些 Unity 版本上勾选该选项可能导致设备死机或者应用闪退。

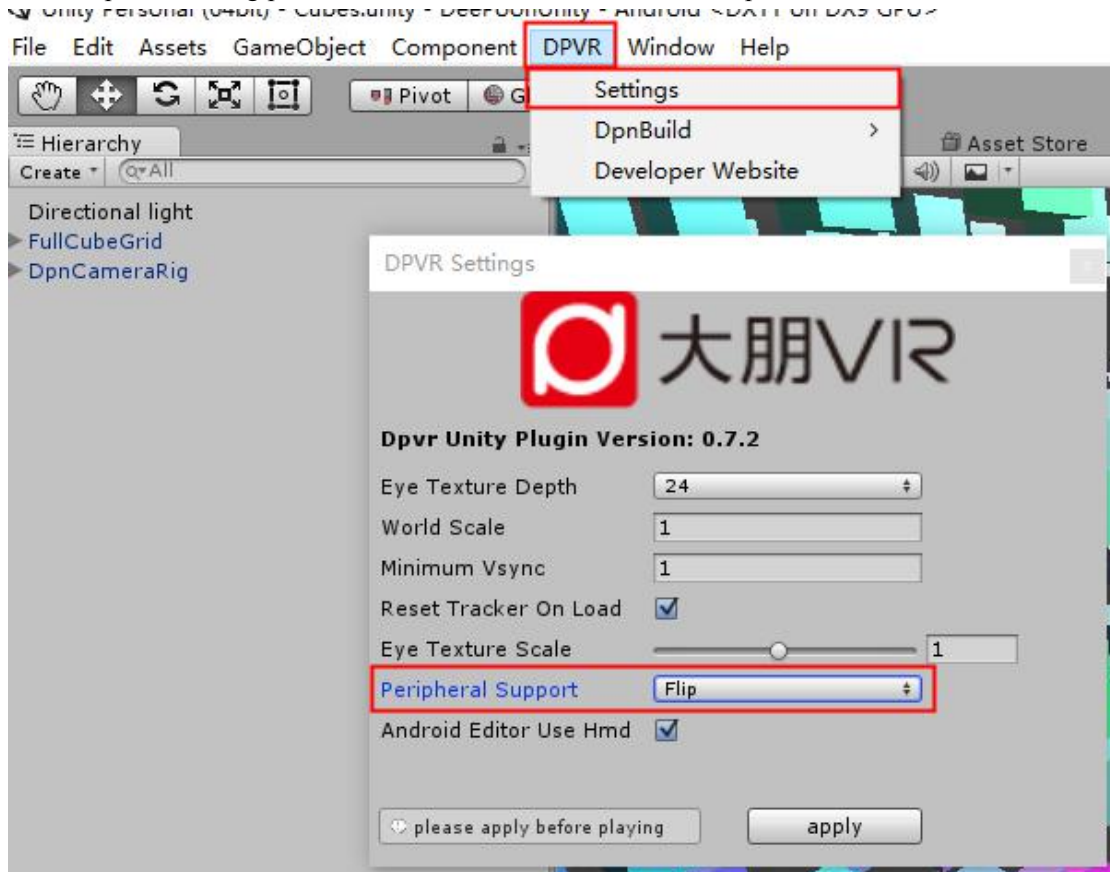


Time 设置（推荐而非强制）

打开 Edit->Project Settings->Time, 设置“Fixed Timestep”为 0.01。

3Dof 手柄适配

- 1、正常导入 DPVRunity 插件。
- 2、将菜单栏->DPVR->Settings 中，Peripheral Support 选项设为 Flip。
- 3、将 DpnCameraRig.prefab 会在运行时自动生成手柄相关的 prefab，包含代码和模型。



如何获取手柄信息：

- 1、如果要获取手柄的姿态和按键信息，可以参考 Daydream 手柄，在 DpnDaydreamController.cs 脚本上使用相关属性来获取，属性说明请参考脚本注释。相比 Daydream 手柄，我们的手柄添加了 trigger 按键，删除了 VolumeUp 和 VolumeDown 两个按键，App 键被修改为 Back 键，Home 键保持不变。其中 Back 键和 Home 键有系统默认行为，无需应用特殊处理；

当前系统默认实现如下：

Back: 功能同系统返回键；

Home: 短按功能同 android 手机的 Home 键，长按 recenter 校准；

- 2、手柄默认自带射线碰撞系统，若将菜单栏->DPVR->Settings 中 Use DPVRPointer 选项勾选，此时将开启大朋的碰撞系统（默认行为）。若不勾选该选项，用户可使用自定义的碰撞

检测逻辑：



3、如果将菜单栏->DPVR->Settings 中 Peripheral Support 选项设为 None，则应用默认使用头控，光标始终位于正前方，用户通过头盔右侧触摸板与 VR 内容交互。当然如果此时系统已经和一个手柄配对成功，用户仍然可以通过操作手柄与 VR 内容交互，仍会发送触摸信息、扳机信息（等价于触摸事件）和按键（back/home）消息给头盔；

4、DPVRunity 0.7.3b 及之后的版本中，手柄支持左右手模式切换，设置接口为 Interface name : DpnDaydreamController.interactiveHand { set, get }。这个值 0 代表右手，1 代表左手。（需配合最新的 Image 使用）；

5、在 DPVRunity 0.7.6 及之后版本，UI 互动的逻辑请使用 DpnEventSystem 替换 EventSystem 的方式(请参看[《我的第一个 VR 应用\(Unity\)》](#)第 7 点内容)或者用 DpnStandaloneInputModule 替换 StandaloneInputModule 的方式（请参看[《现有 Unity3D 游戏的 VR 移植教程》](#)第 9 点内容）的实现示例。

Editor 模式辅助头控

在 DPVRunity plugin 0.7.3b 及之后的版本中，增加了一个简单的设置用来在 Editor 模式下辅助头控朝向以进行简单的效果预览。

Editor 模式运行时，同时按住（Alt 键 + 鼠标右键）并移动鼠标，完成旋转相机观察场景