



HuaweiVR SDK2.0 开发指南

文档版本: v2.0.0



版权所有 © 华为技术有限公司 2018。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

修改记录

日期	版本	修改描述
2017-09-08	v2.0.0	第二版

目录

HuaweiVR SDK2.0 开发指南.....	1
1 概述.....	6
1.1 功能介绍.....	6
1.2 基本知识准备.....	7
1.3 支持平台.....	7
2 开发指南.....	7
2.1 开发准备.....	7
2.2 开发流程.....	8
2.2.1 新建Unity工程并导入SDK2.0.....	8
2.2.2 在场景中添加Huawei VR相机.....	10
2.2.3 控制器集成.....	11
2.2.4 应用程序的AndroidManifest配置.....	13
2.2.5 VRLauncher图标要求.....	14
2.2.6 编译配置.....	14
2.2.7 自定义Activity设置.....	15
2.2.8 2D界面提示设置.....	16
2.2.9 集成多个相机.....	16
2.2.10 集成头瞄预制件（仅用于HuaweiVR1.0）.....	17
2.2.11 在Unity Editor Play模式下预览调试.....	18
2.2.12 降功耗优化措施.....	19

1 概述

1.1 功能介绍

开发者使用HuaweiVR SDK2.0 和Unity进行华为VR设备的应用开发，HuaweiVR SDK2.0 for Unity（以下简称SDK2.0）提供的功能主要包括：

- 自由设定应用场景
通过接口函数，替换天空盒材质，实现应用场景的快速替换。
- 陀螺仪传感器精准数据捕获
华为 VR 眼镜提供更精准的陀螺仪传感器数据，可通过接口函数实时获取陀螺仪数据。
- 手柄支持
提供手柄交互功能，通过接口函数获取手柄状态、类型、姿态、触摸板和按键的状态，重置手柄姿态，预制手柄快捷键功能。
- 音量调节
提供 VR 模式下的音量调节功能，开发者无需额外集成，可通过华为 VR 眼镜的音量调节按键控制音量及显示。
- 来电接听、拒接及挂断
提供 VR 模式下的来电接听、拒接及挂断功能，开发者无需额外集成。当有来电时，启用来电模式，用户可根据弹框提示进行来电接听、拒接及挂断操作。
- 通知消息自动弹框
提供 VR 模式下的通知消息自动弹框功能，开发者无需额外集成。当有通知消息时，即显示对应弹框，通知消息种类包括：短信弹框，一般消息弹框（如微信等），低电量提示弹框，WLAN 连接变化弹框，温度告警弹框。
- 设置界面
提供 VR 模式下的设置功能，开发者无需额外集成，通过长按手柄返回键进入设置菜单界面，设置界面功能有：免打扰模式开启及关闭，重定位，亮度调节。
- 锁定及解锁视角
提供 VR 模式下的锁定及解锁视角功能，通过接口函数，实现场景内的视角锁定及解锁的功能。
- 分场景设置相机渲染分辨率
提供 VR 模式下的分场景设置相机渲染分辨率功能，通过界面选择缩放因子，修改场景的渲染分辨率的功能。
- 3D 音效

提供音频插件，实现 3D 音效功能。开发者可在 3D 虚拟场景中实现声音的方位渲染、距离模拟以及环境音效渲染，以达到更具沉浸感的声音体验。

➤ 2D 下启动

提供 2D 界面提示功能，开发者需要通过特定设置（参考开发指南 2.2.7 节），在 2D 界面下启动 VR 应用。

➤ 应用内截图

提供 VR 模式下的截图功能，开发者无需额外集成，通过手柄快捷键（Home + Trigger），在应用过程中实现截图。

➤ 自适应分辨率

提供 VR 模式下自适应调节分辨率的功能，开发者无需额外集成，当手机温度过高时，可自适应降低分辨率以降低功耗。

➤ 前向兼容

开发者在 HUAWEI VR SDK2.0 上开发的应用可以同时支持在 HUAWEI VR 1.0 和 HUAWEI VR 2.0 上使用。

1.2 基本知识准备

在使用 SDK2.0 之前，开发者需要对 Unity 平台及 C# 语言有一定基础。

1.3 支持平台

- 手机：华为旗舰手机（如：Mate10，Mate9Pro，P9 Plus 等）
- 操作系统：EMUI5.0 及以上版本（Android N 版本及以上）
- Unity 版本：5.5 及以上

2 开发指南

使用 Unity5.5 及以上版本，构建 Android 平台的 Huawei VR 应用，本指南将展示如何设置 Unity，开发并构建 Huawei VR 应用。

2.1 开发准备

- 在 PC 端安装 Unity 平台；
- 下载 Huawei VR SDK2.0 For Unity 压缩包；
- 安装华为 VR SDK 服务 Apk 和手柄服务 Apk 至手机；
- 准备好华为手机及华为 VR 头盔。

2.2 开发流程

2.2.1 新建 Unity 工程并导入 SDK2.0

- 打开 Unity，新建 3D 工程，如图 1 所示

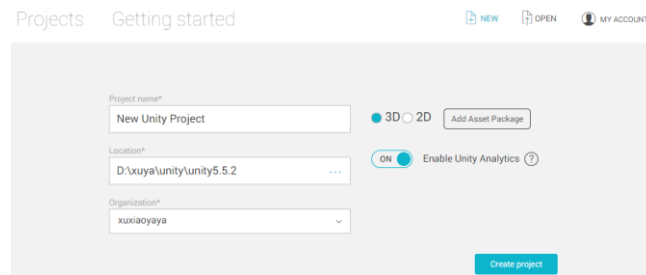


图 1 Unity 平台启动界面

- 依次点击 Assets->Import Package->Custom Package，如图 2 所示

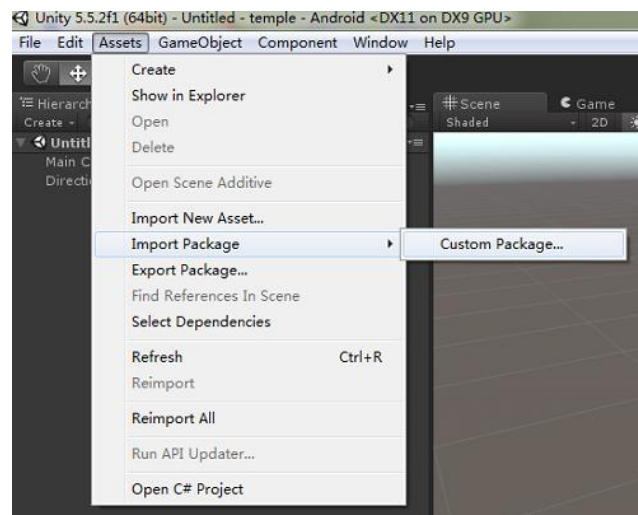


图 2 导入unitypackage操作界面

- 在弹出的对话框选择已下载的 HvrSdk.unitypackage，点击 Import，即可导入，如图 3 所示

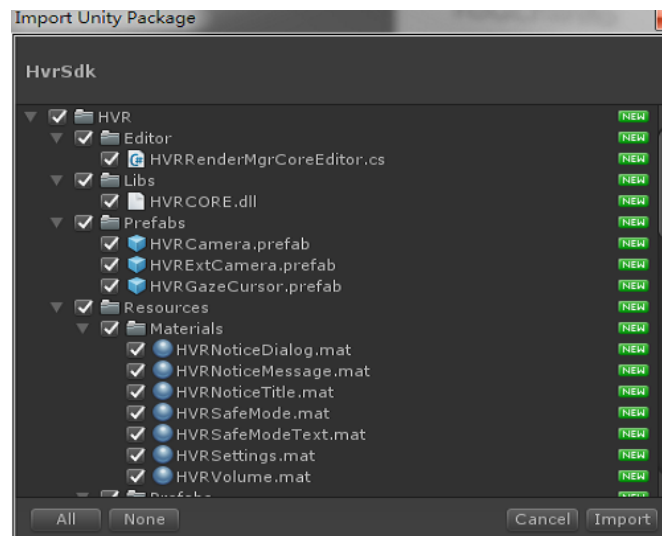


图 3 unitypackage目录界面

注意:

- HvrSdk.unitypackage 导入后, 由 Assets/HVR/Scripts/HVRAddUILayer.cs 自动添加 HVRUILayer 和 HVRVolumeUILayer 两个 UI Layer, 如图 4 所示, 分别用于绘制头瞄和音量弹框。若自动添加 UI Layer 失败, 会有弹框提示, 如图 5 所示, 需要开发者手动添加名称为 HVRUILayer 和 HVRVolumeUILayer 的 UI Layer。

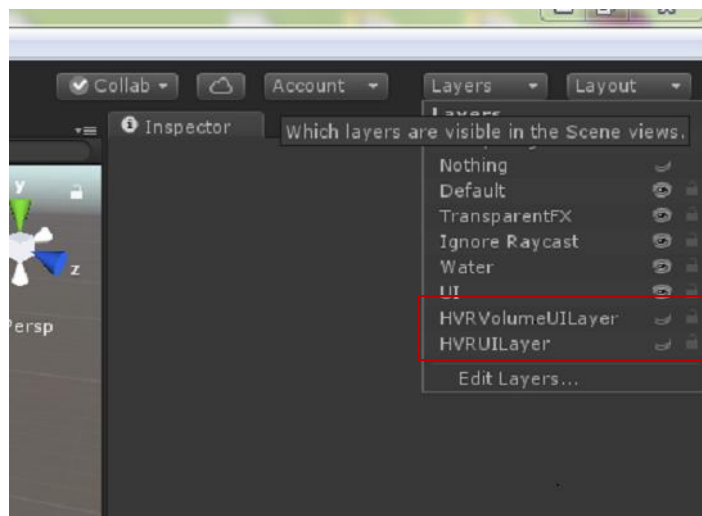


图 4 Layers 界面

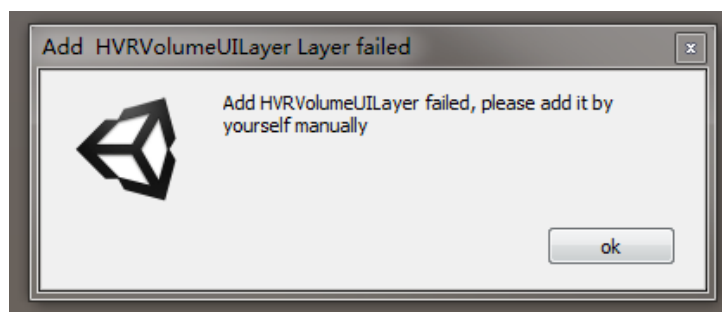


图 5 添加UI Layer 失败提示框

- 开发者不允许在这两个 UI Layer 上绘制其他内容，也不允许场景中其他 Camera 显示这两个 UI 层（例如：修改 SDK2.0 的左右 Camera 的 CullingMask，致使显示这两层内容）。

2.2.2 在场景中添加 Huawei VR 相机

为了让场景能够显示成VR模式，首先需要添加华为VR相机，即HVRCamera，目录为Assets/HVR/Prefabs，添加方法为鼠标点击HVRCamera，并按住鼠标左键拖入至 Hierarchy工作区，并且删除工程建立时自动生成的相机MainCamera，如图6所示。

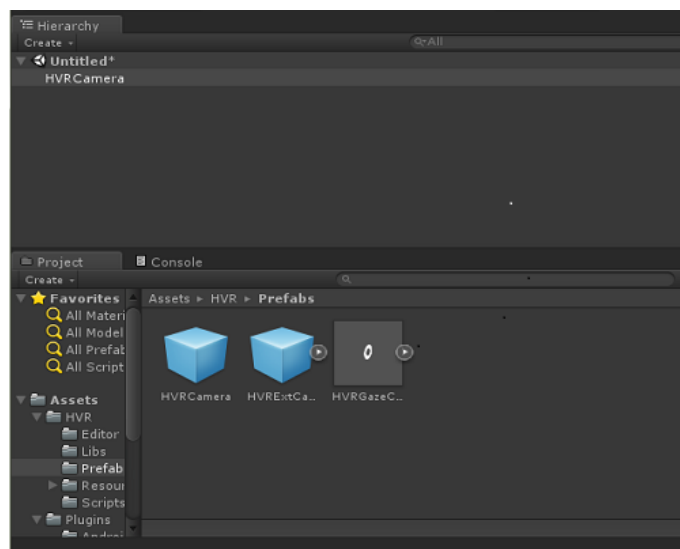


图 6 添加HVRCamera界面

添加HVRCamera之后，场景已经能够在手机上显示VR模式，单击HVRCamera，在 Inspector栏中会显示其属性，如图7示，其中：

- RenderDepthFormat 为 Unity 中初始化 RenderTexture 时的 depth 参数，默认为 24bit，可选择设置为 16bit、24bit 或 32bit；
- RenderResolutionScale 为 Unity 中初始化 RenderTexture 时的分辨率缩放倍数，默认值为 1，默认分辨率：1080*1080，开发者可根据场景复杂度设置渲染分辨率。

注：修改此分辨率会改变内容清晰度，同时也会影响性能和功耗。

- SinglePass仅在开启Multiview模式时勾选，开启Multiview模式具体方式请参考 2.2.12节。

注：普通模式切勿勾选，否则会出现显示问题。

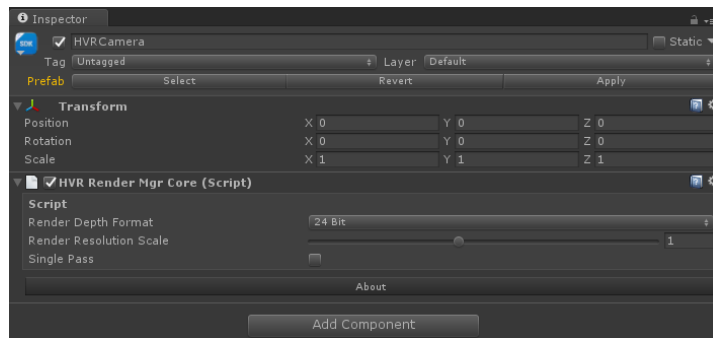


图 7 渲染位深和分辨率设置

2.2.3 控制器集成

SDK2.0需要通过控制器进行交互，具体集成方式如下：

➤ 获取控制器句柄

```
IControllerHandle controllerHandle = HvrApi.GetControllerHandle ();
```

提示：由于 HVR SDK 的初始化工作在 Awake 中完成，建议开发者在 Start 函数中调用 SDK 的接口。

➤ 获取可用的控制器序列号

```
int[] indices = controllerHandle.GetValidIndices ();
```

indices[0]为标准输入事件，用于HuaweiVR1.0头盔触摸板，indices[1]为手柄事件，indices[2]及以上为其他新增手柄。

➤ 根据序列号获取控制器对象

首先获取头盔信息，判断头盔类型（HuaweiVR1.0或HuaweiVR 2.0），参考SDK2.0 API文档4.5.10节。

开发者可通过判断头盔类型来确定交互方式，如：

若为HuaweiVR1.0头盔：

a) 无手柄连接时，使用触摸板，则获取 indices[0]的控制器对象：

```
IController controller = controllerHandle.GetControllerByIndex(indices[0]);
```

b) 有手柄连接时，使用手柄，则获取 indices[1]的控制器对象：

```
IController controller = controllerHandle.GetControllerByIndex(indices[1]);
```

若为HuaweiVR 2.0头盔，

a) 不支持触摸板，需使用手柄，则获取 indices[1]的控制器对象：

```
IController controller = controllerHandle.GetControllerByIndex(indices[1]);
```

b) 如果手柄未连接，则需等待手柄连接。

➤ 判断控制器连接状态

获取手柄对象后，需要先判断手柄是否连接，若已连接，再做后续操作。

```
bool status = controller.IsAvailable ();
```

➤ 使用控制器的接口

a) 兼容 HuaweiVR 1.0 头盔触摸板的所有接口（左、右、上、下滑动，单击，按下、抬起，返回键）；

b) HuaweiVR 2.0 提供的手柄接口包含：获取手柄姿态，手柄连接和断开通知，获取触摸板、返回键和 Trigger 键的点击事件（不含 Home 键和音量键），获取触摸板的位置和滑动事件；

c) 若要获取手柄按键的长按、短按事件，可参考示例代码 Assets\HVR\Scripts\HVRControllerUtils.cs 中的实现；

d) 默认快捷键说明：

动作	按键	按键时间	功能
单击	Home键	<1s	进入VRLauncher界面
长按	Home键	>=1s	视点重置
长按	返回键	>=3s	进入VRSetting界面
长按	Home + Trigger	>=1s	实现截图

e) 开发者如需自定义快捷键，请避免与上述快捷键组合冲突。

备注：手柄状态在Update()中更新，如果要保持应用中所有获取手柄状态的脚本每一帧获取到的手柄状态一致，则需要开发者手动调整脚本执行顺序，确保应用所有获取手柄状态的脚本都在SDK脚本之前或之后执行：

提示：手动设置脚本执行顺序的方式：Edit -> Project Settings -> Script Execution Order 菜单项中，通过设置脚本数字实现。

➤ 模拟 6 自由度功能

SDK Demo中会提供模拟6自由度功能的脚本HVRArmModel.cs，开发者使用后可以获得更加真实的手部体验。

调用方式如下：（可参考demo中HvrControllerTrace.cs中调用方式）

a) 在 Start（）中调用：HVRArmModel.Instance.OnInit();

b) 在 Update（）中按如下方式更新手柄模型位置：

```
m_Controller.transform.position = HVRArmModel.Instance.mWristPosition;
```

(m_Controller为手柄模型变换组件)

c) 更新完模型位置后，调用如下接口：

```
HVRArmModel.Instance.OnUpdateController();
```

2.2.4 应用程序的 AndroidManifest 配置

应用程序的 AndroidManifest 必须符合以下内容：

- 在 application 下包含以下 meta-data，用来标明应用为华为 VR 应用

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

- 在 application 下包含以下 meta-data，为了在 Mate10 Pro 等特殊比例的屏幕上显示正常

```
<meta-data android:name="android.max_aspect" android:value="2.1" />
```

- 在 application 下包含以下 meta-data，设置应用类型："game"或"video"，默认类型为"game"，SDK 会根据类型调整温控策略。

```
<meta-data android:name="com.huawei.android.vr.application.type"android:value="game" />
```

- 将 VR Activity 中的 intent-filter 修改为以下形式，为了去除桌面图标并让 VrLauncher 可以跳转

```
<activity android:name="yourPackage.yourActivity">
    <intent-filter>
        <action android:name="com.huawei.android.vr.action.MAIN"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

- 包含以下权限，用来使用系统 VR 服务

```
<uses-permission android:name="com.huawei.android.permission.VR"/>
```

- 包含以下权限，用来连接手柄服务

```
<uses-permission android:name="com.huawei.vrhandle.permission.DEVICE_MANAGER" />
```

- 包含以下权限，用来支持截图功能

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

以上内容都可参考Assets\Plugins\Android下默认的AndroidManifest.xml

另外，如果开发者需要不连接头盔看到手机上的显示，需要在AndroidManifest的application下包含以下meta-data，但是，正式发布的应用程序中**必须去掉**该标签，此功能仅用于调试

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

2.2.5 VRLauncher 图标要求

如果开发者需要在VRLauncher中显示应用图标，则需要按照如下规格设置：

- 名称：vr_icon.png
- 大小：200KB 以内
- 格式：png
- 分辨率：576*324；
- 图标的存放路径：Unity3D 工程的 Assets/StreamingAssets/vr_icon.png

2.2.6 编译配置

- 选择 Unity 界面中 File->Build Settings...，在弹出 Build Settings 对话框中，Platform 选择 Android；
- 点击对话框中的 Player Settings...，在出现的 Inspector 页面下，选择 Other Settings，勾选 Multithreaded Rendering 选项；
- SDK2.0 暂不支持 Vulkan，建议勾选 Auto Graphics API 选项；
- 在 Package Name 设置包名，如 com.huawei.vrdemo，如图 8 所示。

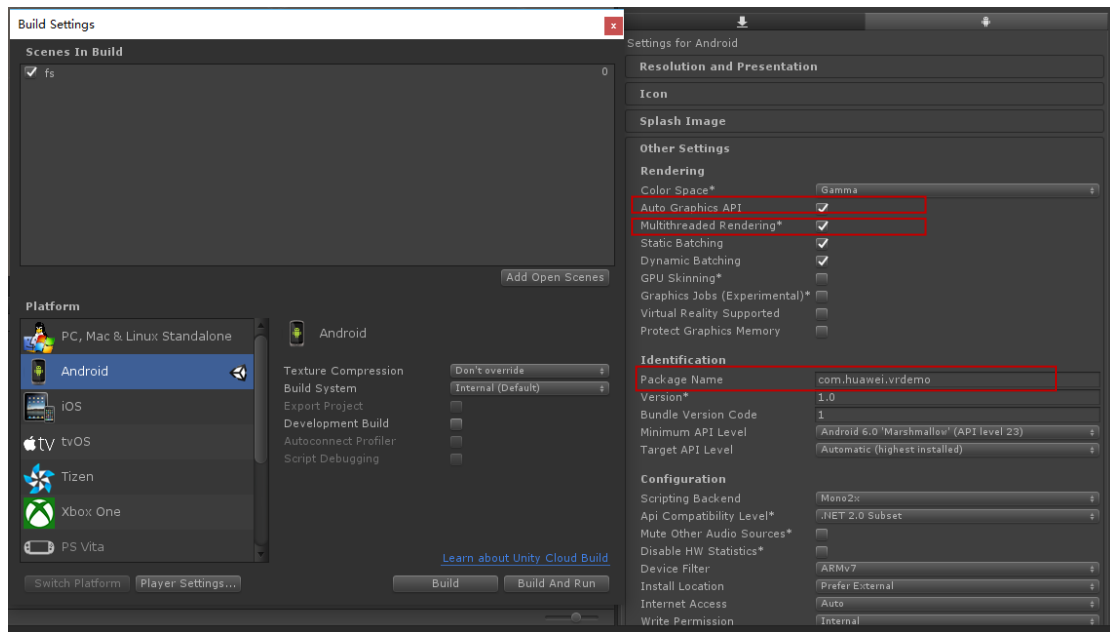


图 8 Build Settings设置界面

点击Build Settings对话框中的Build生成APK, 或者连接手机再点击Build And Run直接下载到手机运行。

2.2.7 自定义 Activity 设置

如果开发者所编译的Android应用有自定义的Activity, 这时应继承HuaweiVR Unity SDK的HVRActivity。

提示: HVRActivity定义在Assets\Plugins\Android下的hvrbridge.jar中, 继承自UnityPlayerActivity。

在Android工程中进行以下操作:

```
import com.huawei.vrlab.HVRActivity;

public class MainActivity extends HVRActivity {

    .....

}
```

同时在 AndroidManifest.xml 中将 Activity 名字修改为开发者自定义的 Activity 名, 且需注意 intent-filter 必须符合 2.2.4 章节的要求。

2.2.8 2D 界面提示设置

SDK2.0提供2D界面提示功能，该界面包含在Assets\Plugins\Android\hvrprompt.aar中，会默认打包到应用中。

只有当开发者的应用存在2D Activity，需要从2D Activity动态切换到VR Activity时才应使用该功能。

开发者只需在自己的2D Activity中，使用以下示例代码跳转到该Activity，就会弹出提示界面，插入头盔后就能跳过VrLauncher，直接启动应用自己的VR Activity。

```
Intent intent = new Intent();

intent.setPackage(getPackageName());

intent.setAction("com.huawei.android.vr.PROMPT");

startActivity(intent);
```

2.2.9 集成多个相机

如果需要集成多个Unity相机的功能，可以按如下方式添加：

将Assets/HVR/Prefabs目录下的HVRExtCamera拖入至Hierarchy工作区中，我们可以看到HVRExtCamera下挂载了两个相机，如图9所示，分别为左眼相机LeftCamera和右眼相机RightCamera，对应华为VR头盔中的左眼视图和右眼视图，在这里可拖入多组HVRExtCamera，然后对每一组相机的Clear Flags、Culling Mask、Depth等参数进行修改，实现多相机功能。

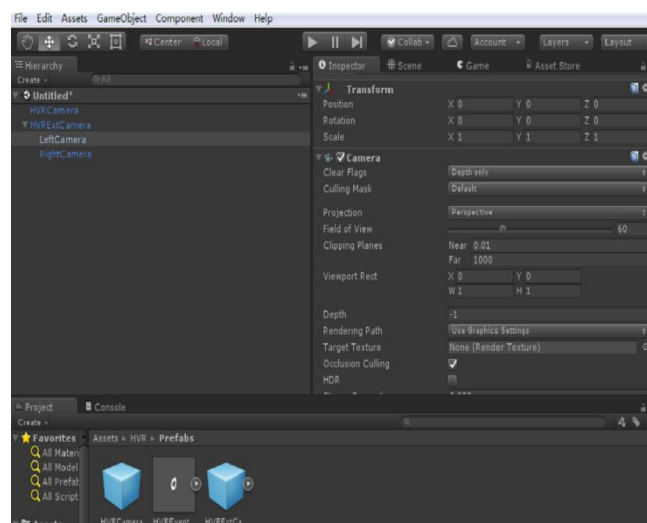


图 9 多相机预制件设置界面

2.2.10 集成头瞄预制件（仅用于 HuaweiVR1.0）

该头瞄预制件仅用于HuaweiVR1.0，不支持在HuaweiVR2.0中显示，HuaweiVR2.0不建议使用头瞄交互方式，建议使用手柄交互方式。

如果需要集成头瞄功能，可以按如下方式添加。

具体使用方法：

- 将 HvrInputModule.cs 脚本绑定到 EventSystem，并取消勾选原来的 Standalone Input Module.cs 脚本，如图 10 所示；

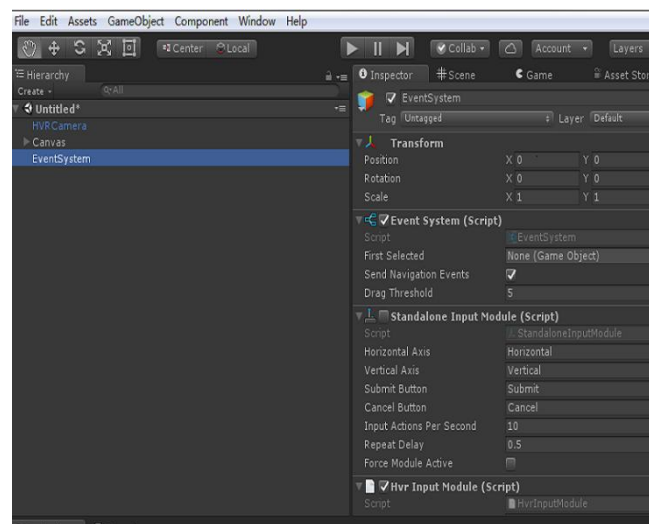


图 10 绑定 HvrInputModule.cs 界面

- 将 Assets/HVR/Prefabs 目录下的 HVRGazeCursor 拖入至 Hierarchy 工作区中，如图 11 所示，可以看到该相机下挂载了 HVREventCamera 和 gaze_cursor 头瞄图片；
提示：开发者可以替换该图片。

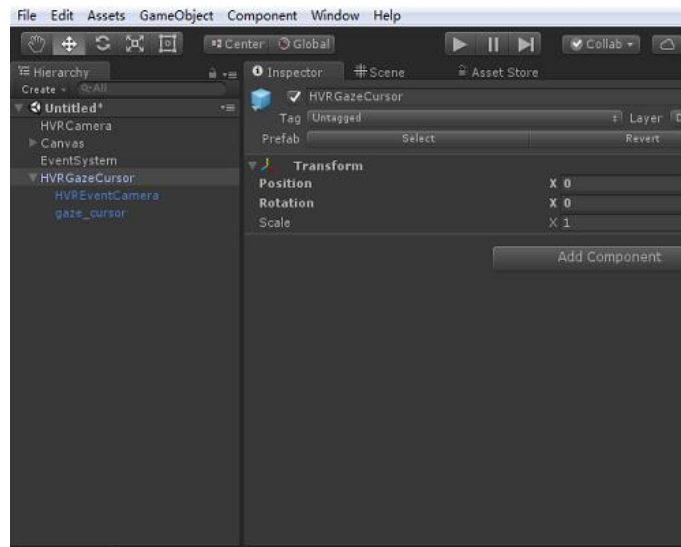


图 11 HVRGazeCursor设置界面

- HVRGazeCursor 用于监听 UI 事件，使用时，
- 将 Canvas 属性中 Render Mode 设置为 WorldSpace;
 - 将 HVRGazeCursor 赋予 Canvas 的 Event Camera;
 - 给 Canvas 绑定 HvrEventListener.cs 脚本(该脚本提供了静态委托变量: onEnter, onExit, onDown, onUp, onClick, 用来传递开发者需要监听的头瞄注视到的 UI 输入事件)，如图 12 所示。

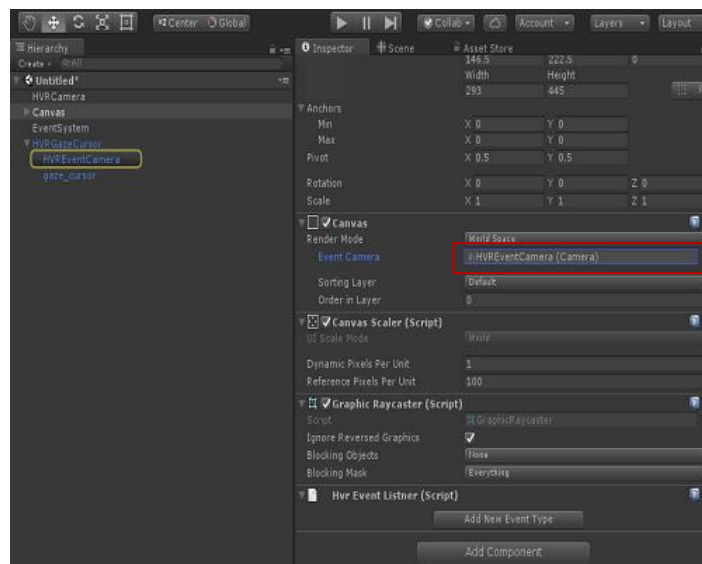


图 12 Canvas设置界面

2.2.11 在 Unity Editor Play 模式下预览调试

开发者可以在Unity Editor Play模式下，利用键盘鼠标控件模拟头部运动及手柄按键

状态，进行预览调试，对应操作如下表所示：

头部运动：

模拟动作	操作方法
转动偏航角（yaw）	按住Alt + 移动鼠标左右移动
转动俯仰角（pitch）	按住Alt + 移动鼠标上下移动
转动翻滚角（roll）	按住Ctrl + 移动鼠标左右移动
重置偏航角（resetYaw）	Alt + 点击‘Z’键

手柄按键：

模拟动作		操作方法
触摸板Confirm 按键	pressed	按住Shift + 点击鼠标左键
	click	点击鼠标左键
返回键	pressed	按住Shift + 鼠标右键
	click	点击鼠标右键
Trigger键	pressed	按住Shift + 鼠标滑轮键
	click	点击鼠标滑轮键
触摸板上滑		按住Shift + 点击方向上键
触摸板下滑		按住Shift + 点击方向下键
触摸板左滑		按住Shift + 点击方向左键
触摸板右滑		按住Shift + 点击方向右键
触摸板触摸		按住Shift 键

2.2.12 降功耗优化措施

HuaweiVR SDK2.0针对不同VR应用场景提供Multiview和单目渲染两种降功耗优化措施，需要开发者基于新增接口选择适配。

场景与接口的匹配建议如下：

- ✧ 视频类：适用单目渲染
- ✧ 游戏类：适用 Multiview
- ✧ 其他类：需要根据具体应用类型区分，普通工具类（如 VR 虚拟桌面）适用单目渲染接口，强 3D 类（如 VR 三维绘画）适用 Multiview

➤ Multiview 模式开启方式:

- 1) 勾选 SinglePass 选项, 如图 13 所示:

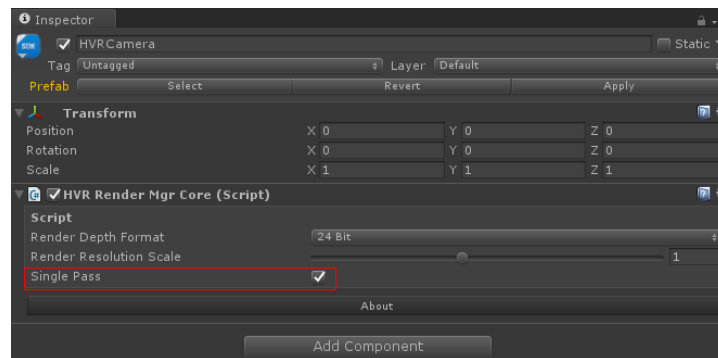


图 13 勾选SinglePass选项

- 2) 勾选 PlayerSettings 中的 Virtual Reality Supported, 清空 VirtualReality SDKs 中的选项, Stereo Rendering Method 选择 Single Pass (Preview), 如图 14 所示:

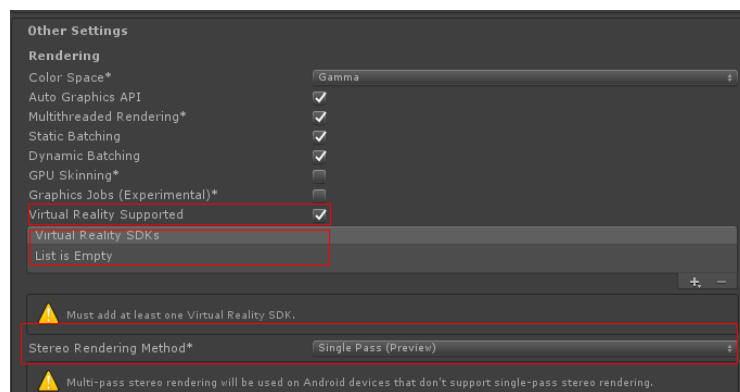


图 14 设置Stereo Rendering Method

注: Multiview模式开启时, 不支持集成HVRExtCamera预制件。

➤ 单目模式开启方式:

调用接口: `void EnableMonocularDisplay(bool enable)`, 调用方式参考SDK2.0 API文档4.8.4节。

注: Multiview和单目渲染两种方式不能同时开启