

秘别：

编号：



Skyworth Standalone VR SDK（Unity）

开发文档

拟制 _____ 日期 _____

审核 _____ 日期 _____

批准 _____ 日期 _____

深圳创维新世界科技有限公司

| | |
|--|----|
| 1 简介..... | 3 |
| 1.1 SDK 介绍..... | 3 |
| 1.2 开发环境..... | 3 |
| 1.3 SDK 构成..... | 3 |
| 2 SDK 的使用说明..... | 4 |
| 2.1 新建工程..... | 4 |
| 2.2 导入 SDK..... | 4 |
| 2.3 使用 SDK..... | 5 |
| 2.4 项目设置..... | 6 |
| 2.4.1 QualitySettings 设置..... | 6 |
| 2.4.2 PlayerSettings 设置..... | 7 |
| 2.4.3 Build Settings 设置..... | 9 |
| 2.4.4 AndroidManifest 设置..... | 10 |
| 2.4.5 Blit Type 设置..... | 10 |
| 2.5 导出到设备运行..... | 10 |
| 3 API 接口函数..... | 11 |
| 3.1 GvrPointerInputModule 说明..... | 11 |
| 3.2 GvrPointerPhysicsRaycaster 说明..... | 11 |
| 3.3 StereoController 说明..... | 12 |
| 3.4 GvrHead 说明..... | 12 |
| 3.5 GvrHeadset 说明..... | 12 |
| 3.6 GvrViewer 说明..... | 13 |
| 4 3DoF 手柄与头盔按键说明..... | 14 |
| 4.1 手柄或头盔的按键响应..... | 14 |
| 4.2 接口说明..... | 15 |
| 4.3 屏蔽手柄 Home 键返回到 Home..... | 15 |
| 5 凝视点击..... | 16 |
| 5.1 使用说明..... | 16 |
| 5.2 组件 UICountDown..... | 16 |
| 6 项目优化..... | 16 |
| 6.1 S801 平台优化..... | 16 |
| 6.1.1 Player Settings 设置..... | 16 |
| 6.1.2 QualitySettings 设置..... | 16 |
| 6.1.3 Camera..... | 16 |
| 6.1.4 天空盒..... | 16 |

1 简介

1.1 SDK 介绍

SDK 支持硬件设备：创维 VR 一体机 S8000,V901,S801

SDK 主要提供：3dof 手柄交互支持，Nolo6DOF，多功能交互支持，双目立体渲染等功能

1.2 开发环境

Unity:

1. Using S800 V901 Setting

支持 Unity 2017.4.6~2018.3.3f1，推荐使用 Unity2018. 4.13f1 版本，不支持 2018.1.9f2, 2018.1.2f1,2018.2.5f1

2. Using S801 Setting 和 Using S8000 V901 Legacy Setting

支持 Unity 2017.4.6~2019.2.3f1，推荐使用 Unity2017. 4.6f1 版本，不支持 2018.1.9f2, 2018.1.2f1,2018.2.5f1

具体设置参考 2.4 节。

Android SDK: API Level 19 及以上。

JDK: jdk1.7.0_01 及以上。

1.3 SDK 构成

为方便开发者使用，SDK 通过 UnityPackage 包的形式提供，开发者导入后可见如下目录：

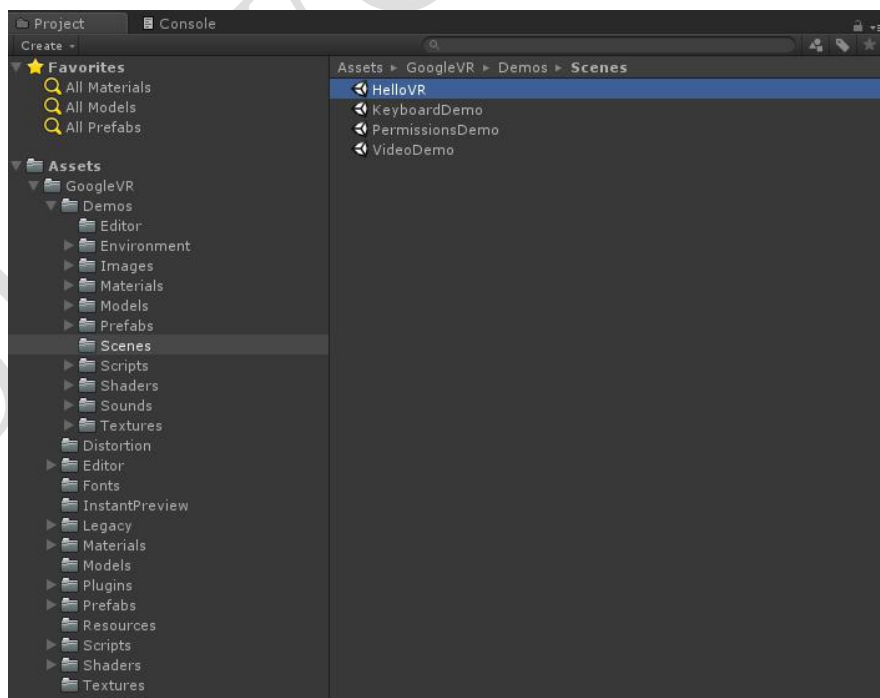


图 1.3 SDK 结构

2 SDK 的使用说明

2.1 新建工程

打开 Unity，新建工程如下：

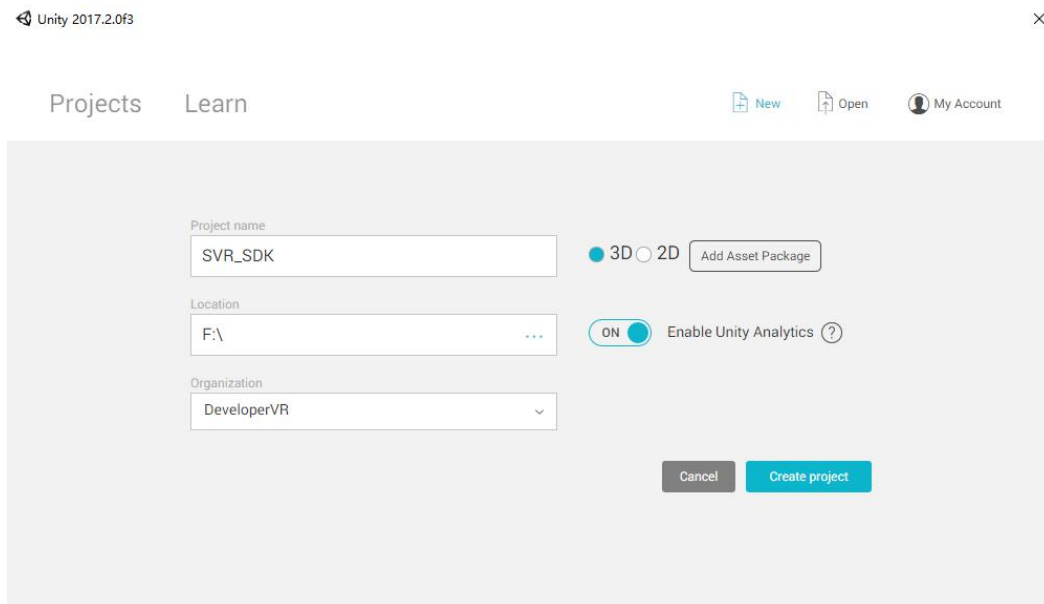


图 2.1 新建工程

2.2 导入 SDK

选择菜单 Assets->Import Package->Custom Package...如下：

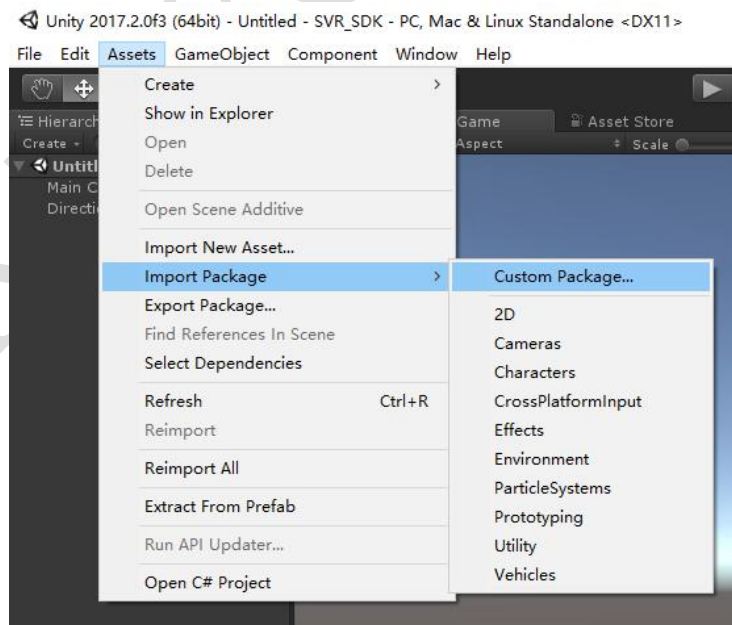


图 2.2 导入 SDK

此时 Unity 会弹出文件选择对话框，选择“svr_unity_sdk.unitypackage”后点击打开如下：

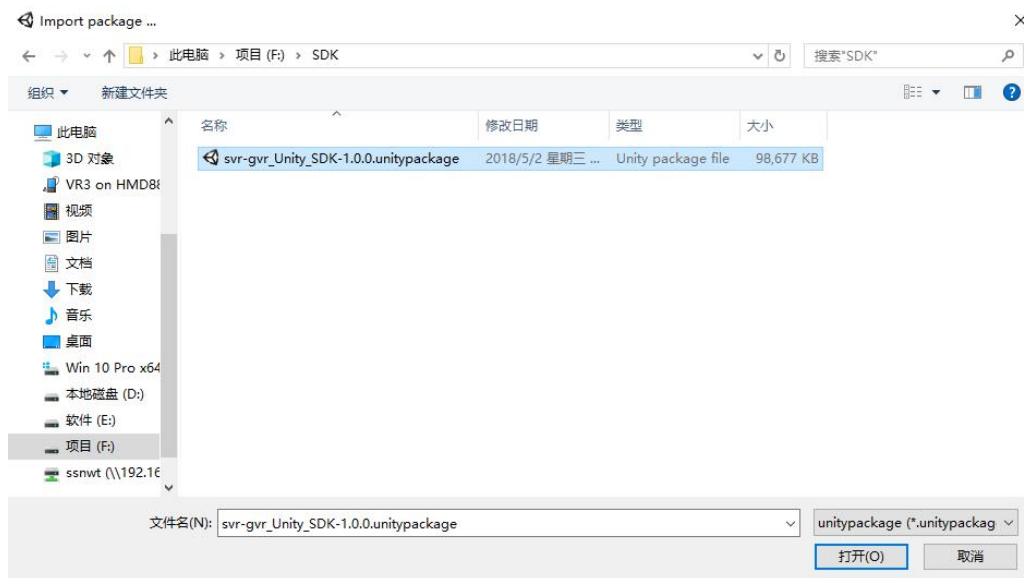


图 2.3 选择 Unity Package

点击后返回至 Unity 界面，系统弹出 SDK 目录层级，请根据需要导入：

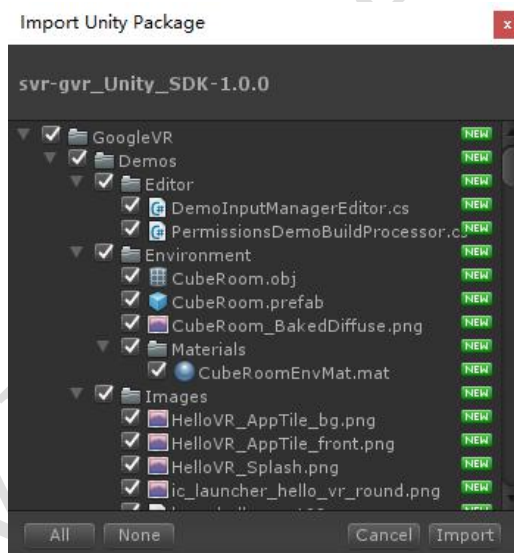


图 2.4 导入选项

2.3 使用 SDK

进入 Project 选项卡，依次展开 Assets->GoogleVR->Demos->Scenes，选择 HelloVR 场景，点击运行按钮，在 Game 窗口中可看到如下：

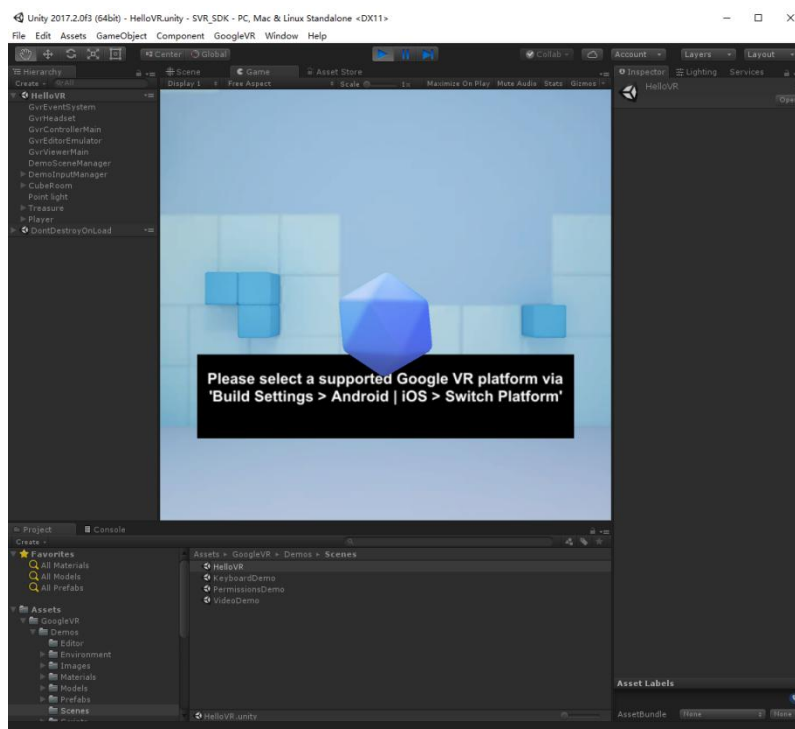


图 2.5 模拟运行

按住 **Alt** 移动鼠标，可上下左右转动画面；按住 **Ctrl** 移动鼠标，可上下翻转画面；按住 **Shift** 移动鼠标，可模拟手柄操作。

2.4 项目设置

我们需要对 Unity 进行一些设置才能正常的运行在设备上。最新的 SDK 支持三种设置，使用后会自动去设置后面所提到的这些项目，比如 **QualitySettings** 设置，**PlayerSettings** 设置，**Blit Type** 设置等。但是最新的 Unity 可能存在兼容性问题，使用自动设置后建议还是对照 2.4.1, 2.4.2, 2.4.3, 2.4.4, 2.4.5 逐一检查。

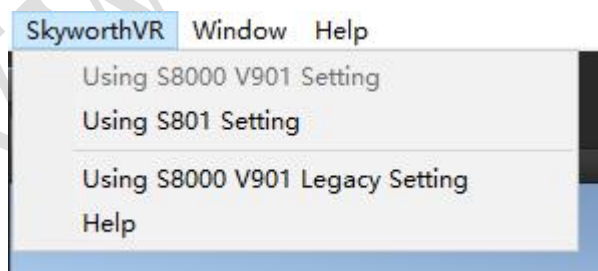


图 2.4 sdk 设置

2.4.1 QualitySettings 设置

如图 2.6 中 **Levels** 选择安卓平台绿色勾选项的“**Medium**”等级。具体参数如下图 2.6 所示：



图 2.6 QualitySettings 设置示意图

注意：Levels 设置中要先让灰色条选中到 **Android** 平台一列打绿色勾的行，如图 2.6 所示意。然后再设置 **Anti Aliasing** 和 **V Sync Count** 两项。如果不先选择到 **Android** 平台，这两项目设置就不会在 **Android** 导出包中生效，导致应用运行是黑屏的。

Anti Aliasing：可以根据需要进行调整，推荐使用 4 倍抗锯齿。

V Sync Count：一定要设置为 Don't Sync。

2.4.2 PlayerSettings 设置

1. Resolution and Presentation 选项卡中，导出设置为横屏。具体参数如下图 2.7 所示：

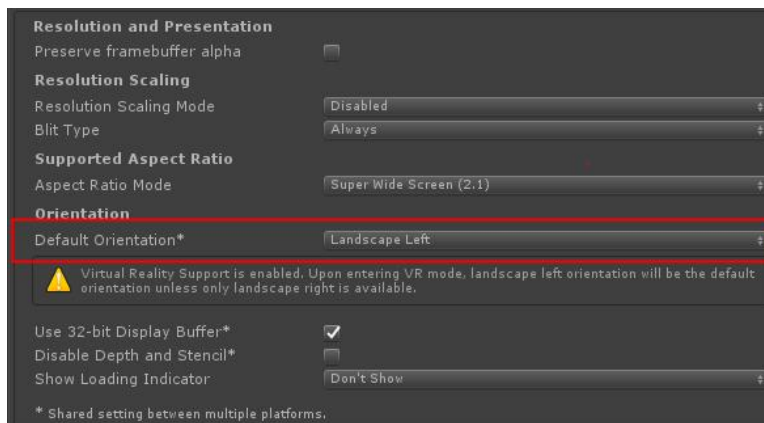


图 2.7 导出设置示意图

2. Other Settings 选项卡中, 不要勾选 GPU Skinning, API Level 为 Android 7.1, 读写权限根据需要选择 SDCard。具体参数如下图 2.8 所示:

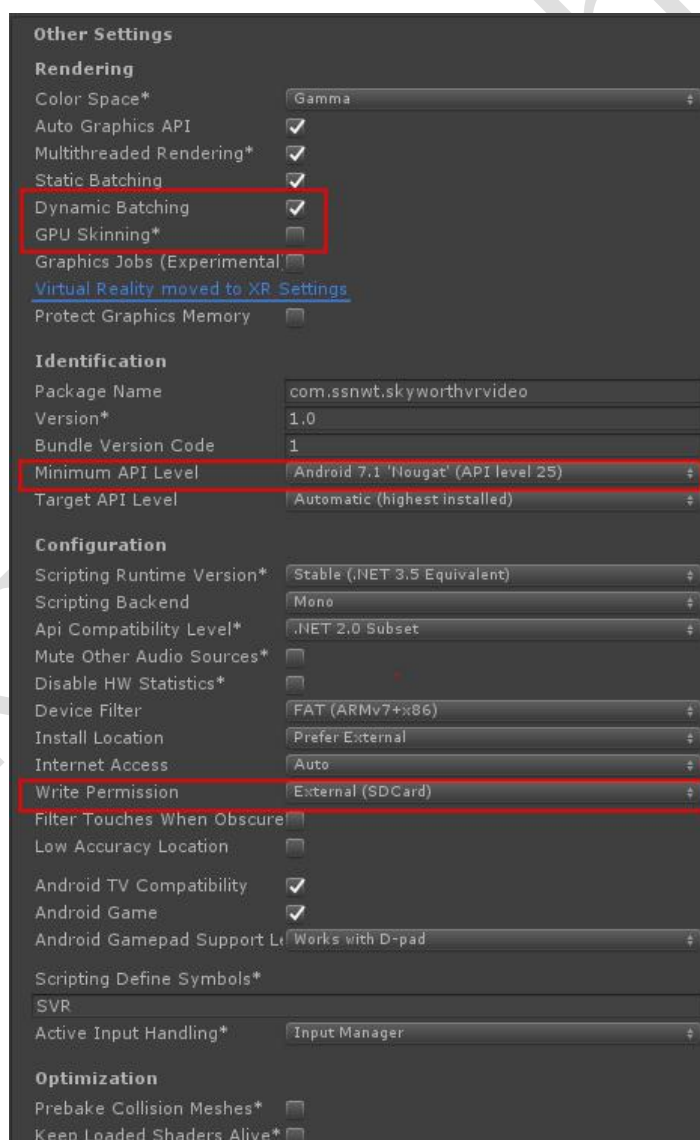


图 2.8 PlayerSettings 设置示意图

3. XR Settings 选项卡中, 勾选 Virtual Reality Supported, 选择“None”, 将 Stereo Rendering Methods 设置为“Single Pass(Preview)”。具体参数如下图 2.9 所示:

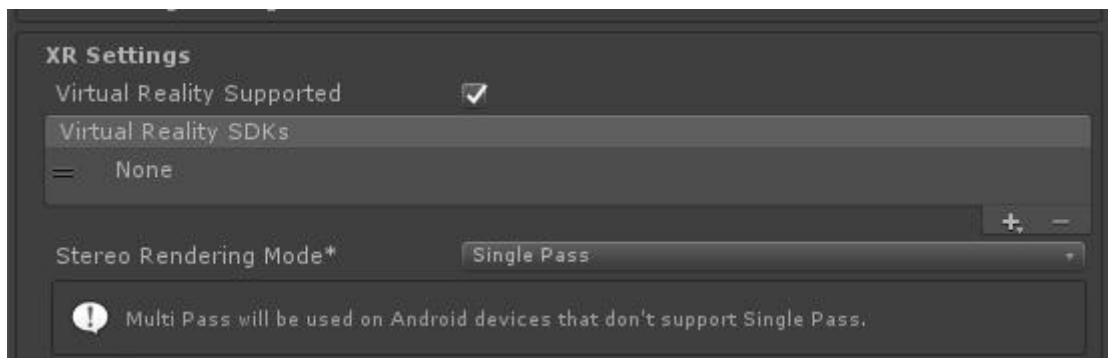


图 2.9 XR Settings 设置示意图

2.4.3 Build Settings 设置

选择默认平台“Android”, 构建系统(Build System)选择“Internal”编译方式。具体参数如下图 2.10 所示:

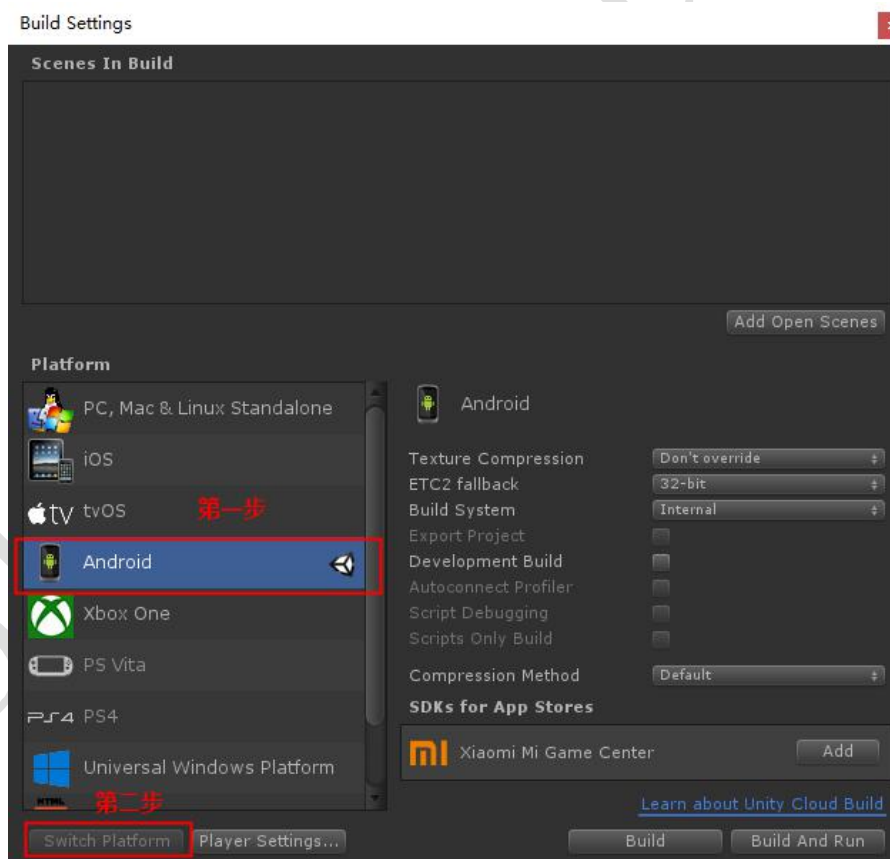


图 2.10 Build Settings 设置示意图

2.4.4 AndroidManifest 设置

将 Activity 配置为 “com.ssnwt.sdk.MainActivity”

1. 如果你的工程里使用了自己的 Manifest 文件并也配置了自己的 Activity, 那么需要将你的 Activity 继承自 MainActivity。
2. 如果你是新建的工程, 需要在工程里创建 Plugins/Android/ 目录, 然后将 GoogleVR/Plugins/Android/AndroidManifest.xml 文件移动到刚才创建的 Plugins/Android/目录。

2.4.5 Blit Type 设置

在 Unity2017.3 以上的版本才会有 Blit Type 的设置项。这里需要手动设置, S801 平台需要设置成 Never, S8000 和 901 平台需要设置成 Always。

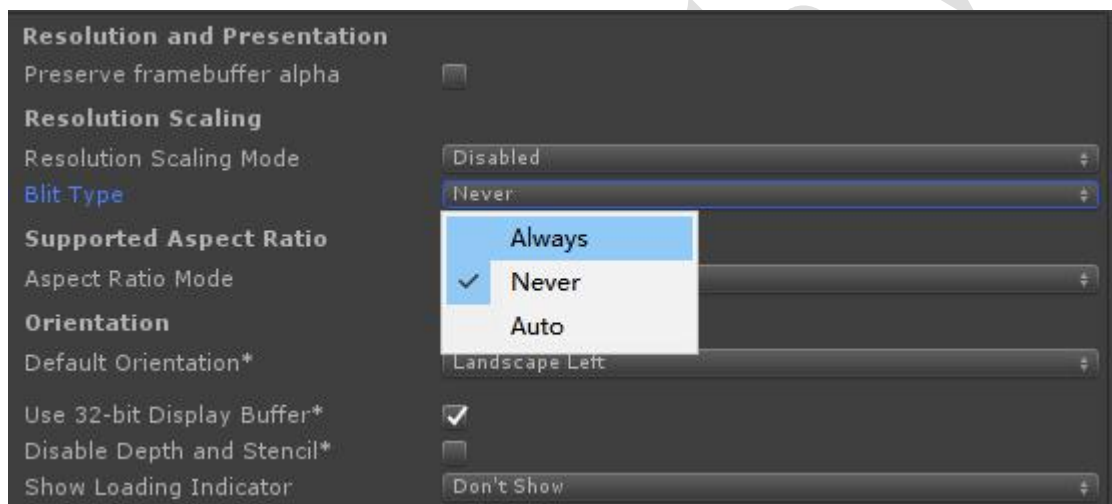


图 2.11 Blit Type 设置示意图

2.5 导出到设备运行

- 1.将设备通过 USB 连接到电脑, 当连上电脑之后, Launcher 界面的电池图标会变成充电状态。
- 2.点击 File->Build & Run, 等待进度条结束就 OK 了!

3 API 接口函数

该 SDK 以 GVR SDK v1.40.0 为基础拓展，下面介绍部分接口，其余接口请前往官方网站 (<https://developers.google.com/vr/unity/reference/>) 获取详情。

3.1 GvrPointerInputModule 说明

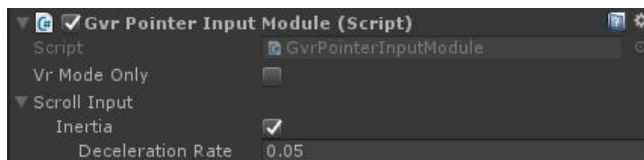


图 3.1 GvrPointerInputModule 设置示意图

功能:

GvrPointerInputModule 继承自 BaseInputModule, 使用此脚本可以让基于 Canvas 的(UGUI)UI 元素和 3D 场景对象在应用程序中进行交互。所以 UI 元素在被 Pointer 选中或 Trigger, Touching 触发的事件, 可以传递出来。

使用:

进入 Project 选项卡, 依次展开 Assets->GoogleVR->Prefabs->UI, 将 GvrEventSystem 预制体放入场景替换原有的 EventSystem。

3.2 GvrPointerPhysicsRaycaster 说明



图 3.2 GvrPointerPhysicsRaycaster 设置示意图

功能:

GvrPointerPhysicsRaycaster 继承自 GvrBasePointerRaycaster, 提供了一个用于 GvrPointerInputModule 的碰撞检测。

使用:

在场景中新建一个空对象, 为其命名为 Player。将场景原有 Main Camera 拖动至 Player 之下。为 Main Camera 添加 GvrPointerPhysicsRaycaster 脚本。

3.3 StereoController 说明

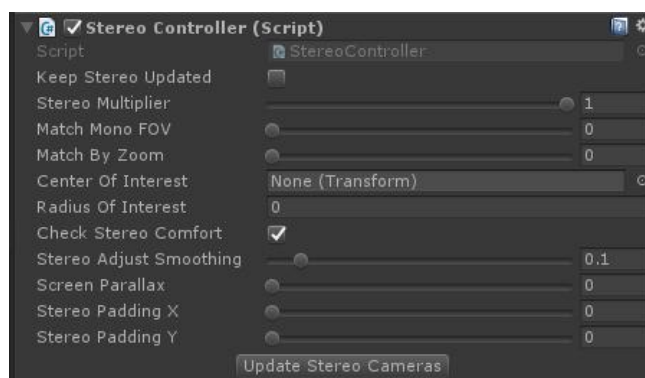


图 3.3 StereoController 设置示意图

功能:

绑定在 MainCamera 上, 两个 GvrEye 渲染 stereo view 内容的控制器, 这个脚本需要绑定到做 VR 渲染的摄像机上。

使用:

为 Main Camera 添加 StereoController 脚本。

3.4 GvrHead 说明

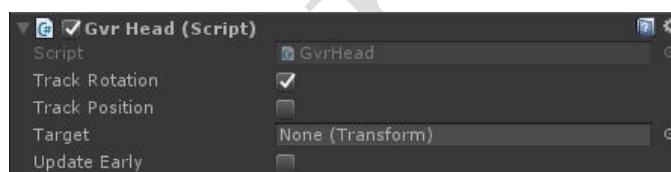


图 3.4 GvrHead 设置示意图

功能:

用户头部跟踪模拟, 提供头部跟踪的数据给 Camera, 在头部运动过程中它附着的 Transform 属性也会同步发生改变。

使用:

为 Main Camera 添加 GvrHead 脚本。

3.5 GvrHeadset 说明

功能:

GvrHeadset 是 VR 一体机耳机 API 的主要接口。一个场景中有且只能有一个 GvrHeadset 预制体。

使用:

进入 Project 选项卡, 依次展开 Assets->GoogleVR->Prefabs->Headset, 将 GvrHeadset 预制体放入场景即可。

3.6 GvrViewer 说明

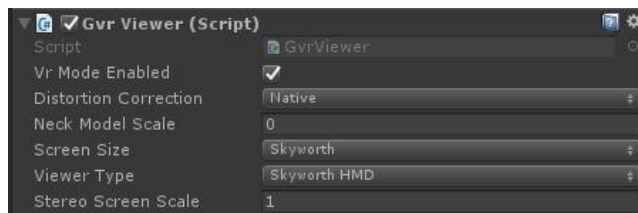


图 3.5 GvrViewer 设置示意图

功能:

用于初始化 Devices,在编辑器运行的时候会初始化 Editor 的 Devices。一个场景中有且只能有一个 GvrViewerMain 预制体。

使用:

进入 Project 选项卡,依次展开 Assets->GoogleVR->Prefabs,将 GvrViewerMain 预制体放入场景即可。

4 3DoF 手柄与头盔按键说明

4.1 手柄或头盔的按键响应

1. 确认键 (头盔和手柄)

GvrControllerInput.ClickButton

GvrControllerInput.ClickButtonDown

GvrControllerInput.ClickButtonUp

2. 返回键 (头盔)

Input.GetKey(KeyCode.Escape)

Input.GetKeyDown(KeyCode.Escape)

Input.GetKeyUp(KeyCode.Escape)

3. Home 键 (头盔和手柄)

GvrControllerInput.HomeButton

GvrControllerInput.HomeButtonDown

GvrControllerInput.HomeButtonUp

4. Trigger 键 (手柄)

GvrControllerInput.TriggerButton

GvrControllerInput.TriggerButtonDown

GvrControllerInput.TriggerButtonUp

5. App 键 (手柄)

GvrControllerInput.AppButton

GvrControllerInput.AppButtonDown

GvrControllerInput.AppButtonUp

4.2 接口说明

SDK 提供 3DoF 手柄控制器的按键与触摸板响应事件，调用手柄相关接口请参照第三章 API 接口函数——GvrControllerInput 说明，这里介绍接口参数与物理手柄按键的对应关系。

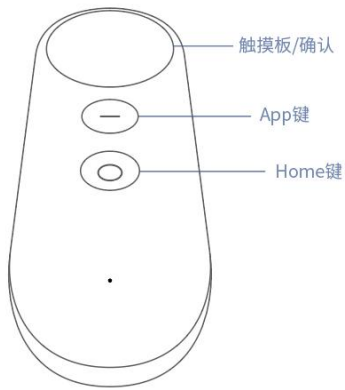


图 5.1 3dof 手柄示意图

手柄物理按键与 API 接口函数中参数的对应关系如下：

| 物理按键 | API 相关接口 |
|---------|---|
| 触摸板/确认 | ClickButton ClickButtonDown ClickButtonUp |
| App 键 | AppButton AppButtonDown AppButtonUp |
| Home 键 | HomeButtonDown HomeButtonState |
| Tigger键 | TriggerButton TriggerButtonDown TriggerButtonUp |

4.3 屏蔽手柄 Home 键返回到 Home

当连接手柄后点了 Home 按键会返回的 Home 界面，如果想实现点 Home 键不返回到 Home 需要在 AndroidManifest.xml 添加一下代码。

```
<uses-feature android:name="android.software.vr.ignore.home" android:required="false" />
```

“true”表示点击 Home 不返回到 Home，“false”表示点击 Home 键会返回到 Home，默认是 false。

5 凝视点击

SDK 中提供了倒计时出发点击的功能, 当不需要手柄操作时可以使用凝视倒计时来触发点击事件。该功能需要和 `GvrReticlePointer` 一起使用。(注: 当期 3DoF 手柄已成为 VR 一体机标准交互方式, 如特定的原因, 请默认使用 3DoF 手柄作为交互方式。)

5.1 使用说明

在 `Assets->GoogleVR->Prefabs->UI` 中将 `SvrReticleDownClick` 预制体放在 `GvrReticlePointer` 下面, 然后在场景中创建一个 `Button`, 点击运行。当锚点 `Hover` 到 `Button` 上时会出现倒计时效果。

`SvrReticleDownClick` 只会检查实现了 `PointerClick` 的物体才起作用, 如果你不是使用的 `PointerClick` 将没有任何效果。

默认情况下倒计时是 1 秒, 我们也提供了 `UICountDown` 组件来控制倒计时时间。

5.2 组件 `UICountDown`

当需要控制按钮倒计时的时间时可以将 `UICountDown.cs` 脚本挂在相应 `PointerClick` 的物体上, 然后设置 `Count` 的值, 单位是秒。

6 项目优化

6.1 S801 平台优化

6.1.1 `Player Settings` 设置

`Blit Type`: 设置为 `Never`。

6.1.2 `QualitySettings` 设置

`Anti Aliasing`: 需要设置成 `disable`。

6.1.3 `Camera`

`Clean Flags`: 需要设置成 `Solid Color` 或者 `Depth only`, 以节省不必要的绘制操作

6.1.4 天空盒

不要使用 Unity 内置的 `skybox` 或者 `sphere` 来实现 360 全景背景, 因为 unity 内置的模型和 `shader` 都比较复杂。请自己建一个 3D 球体模型并贴图显示, `shader` 使用 `Utile/Texture`, 顶点可以控制在 3000 个左右。