

GSXR Unreal SDK

快速入门

V1.1.2

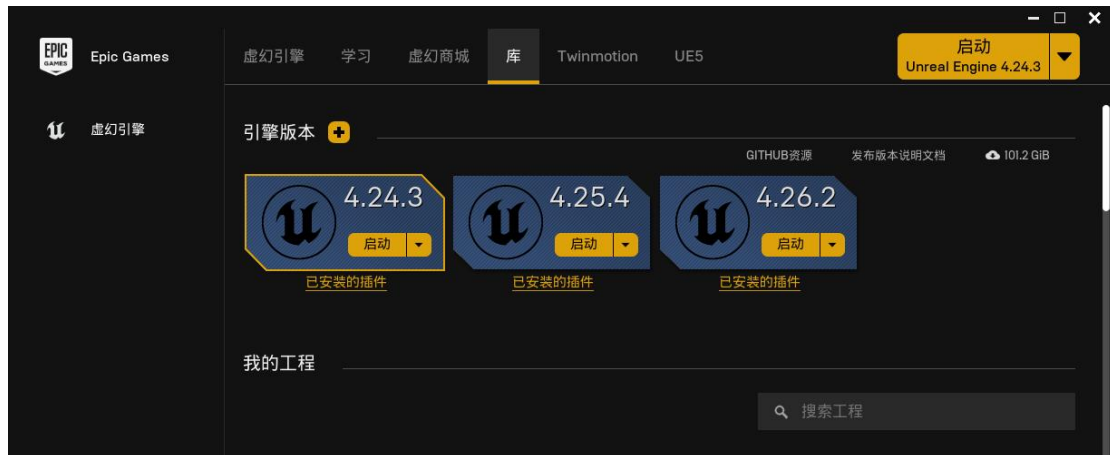
2021 年 11 月 24 日

目录

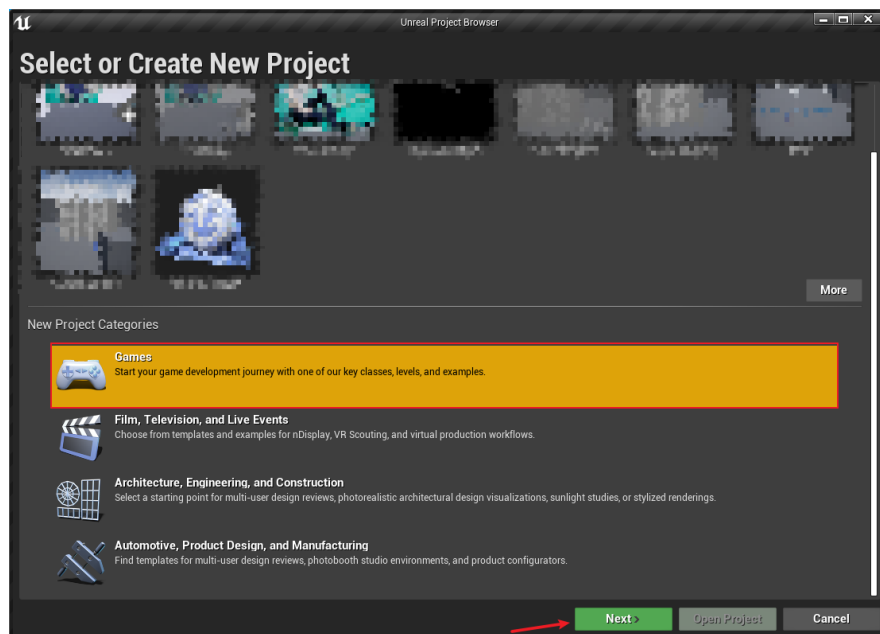
1	GSXR 项目创建	3
2	GSXR 项目设置	6
3	GSXR 按键行为映射	13
3.1	设计目的.....	13
3.2	实现流程.....	13
3.3	按键行为绑定.....	19
4	蓝图接口	21
4.1	获取设备电量状态接口.....	21
4.2	开启/关闭注视点渲染	21
4.3	获取设备连接状态.....	22
4.4	获取注视点渲染参数.....	22
4.5	获取注视点渲染功能状态.....	23
4.6	设置注视点渲染相关参数.....	23
5	动态手柄模型加载	24
6	项目打包	24
7	注意事项.....	26
7.1	设计返回逻辑.....	26
7.2	保存游戏.....	26

1 GSXR 项目创建

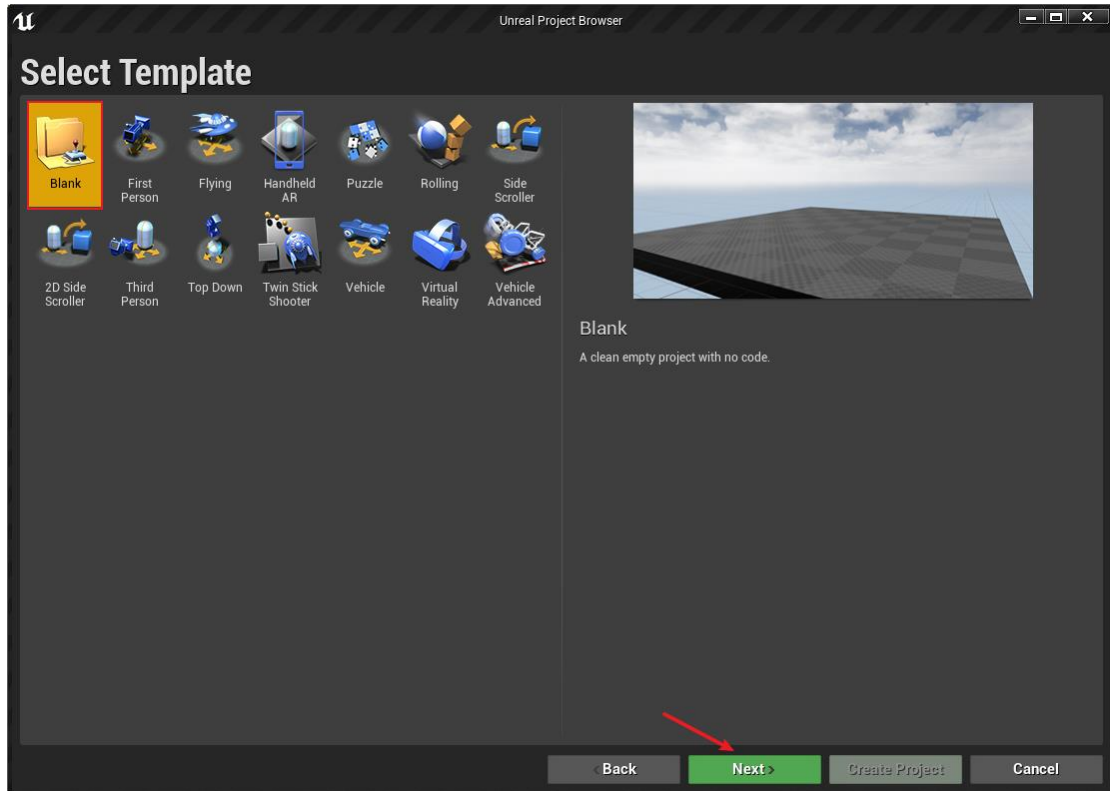
1. 用 Epic Games Launcher 下载并安装 4.24 或更新版本的虚幻引擎 4。完成此操作后，按下 启动（Launch） 按钮打开 项目浏览器。



2. 在项目浏览器中，选择合适的项目类型进行项目创建，这里以 Game 为例，选定 Game 类型，然后点击下方 Next。



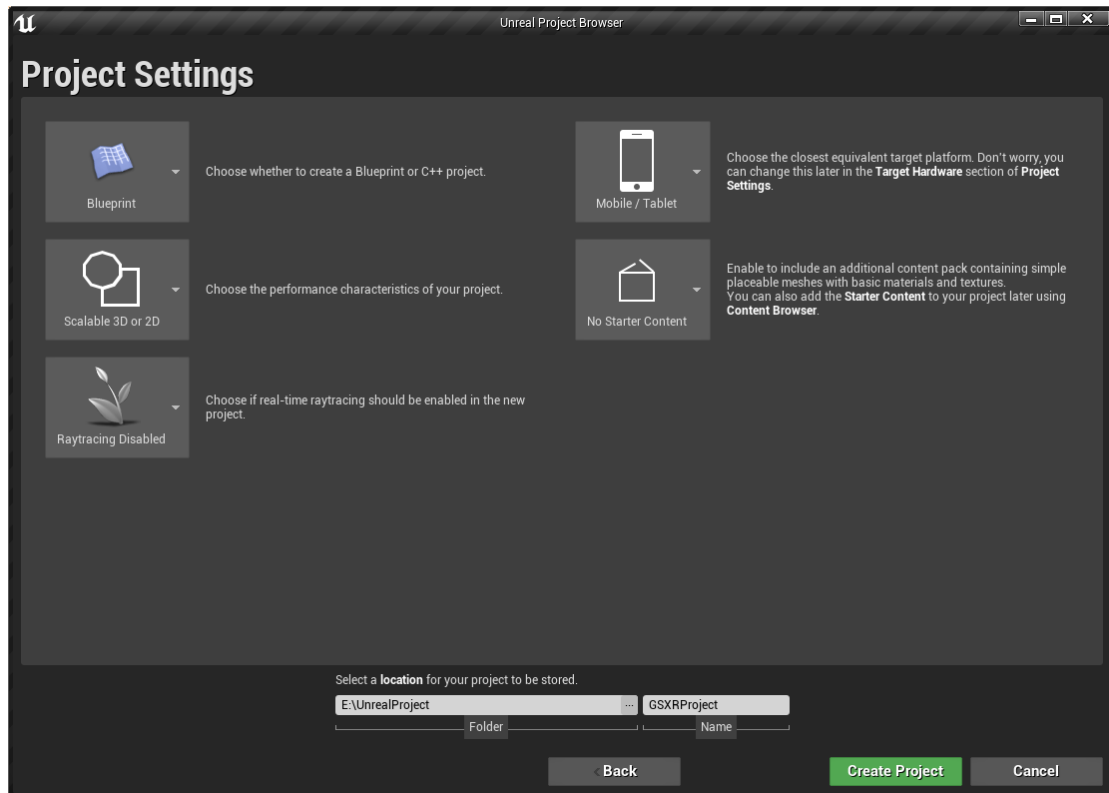
3. Unreal Engine 为开发者提供了多种工程项目模板，在这里以 Blank 项目为例进行演示，现在 Blank，点击下方 Next。



4. 由于 VR 总体渲染要求较高，建议以下列设置开始 UE4 VR 新项目，确保项目从开始便以正确的帧率运行。

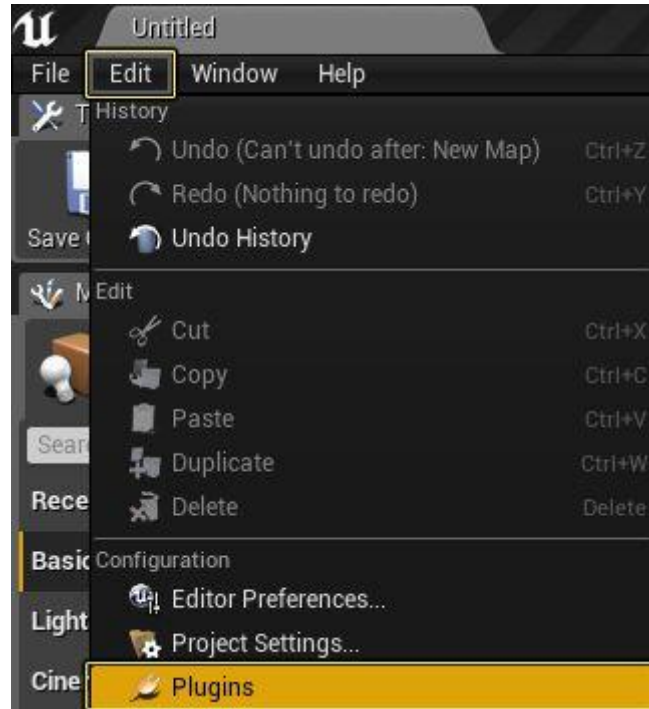
- 选择 蓝图
- 启用 移动/平板设备
- 启用 可缩放的 3D/2D
- 启用无初学者内容
- 禁用光线追踪
- 设置项目的创建路径，并为项目命名。

在此例中，项目将被命名为 **GSXRProject**,最后按下右下角的 **创建项目**按钮创建项目。

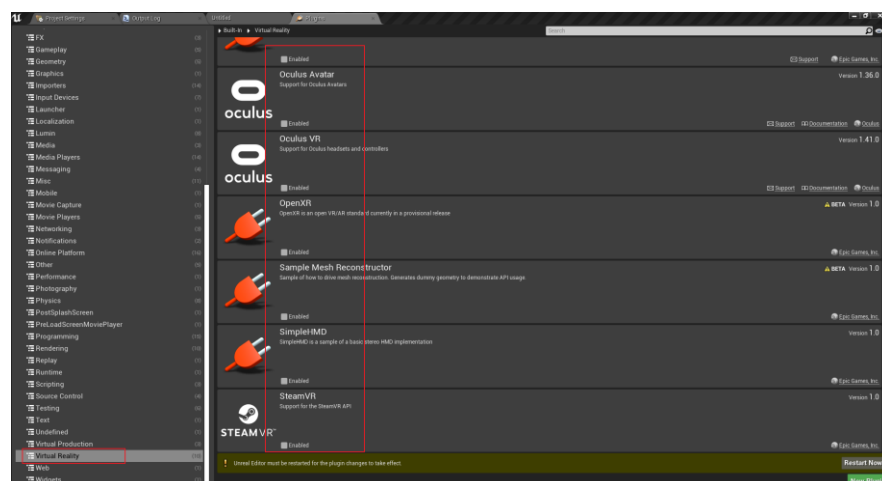


2 GSXR 项目设置

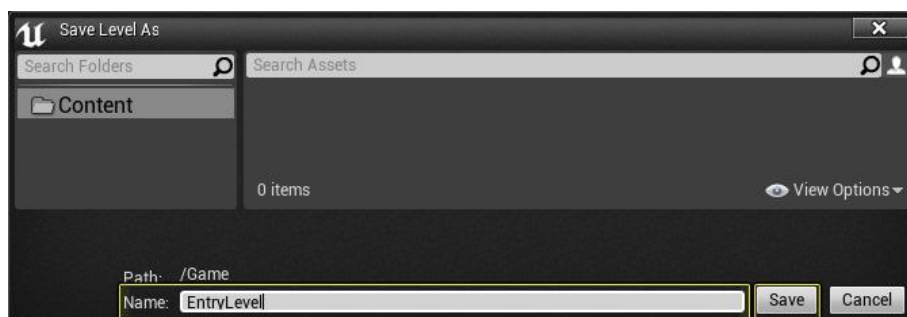
1. 虚幻引擎 4 编辑器加载后，打开 主工具栏 中的 编辑 选项卡，并选择 插件 菜单项。



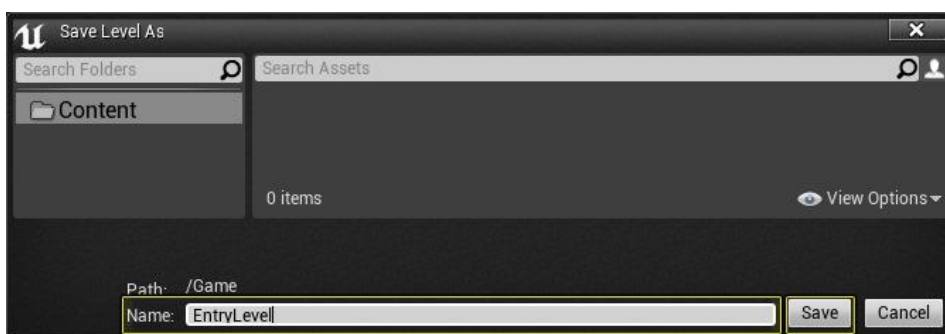
2. 前往 插件（Plugins） 菜单的 虚拟现实（Virtual Reality） 部分，关闭其中的所有插件。



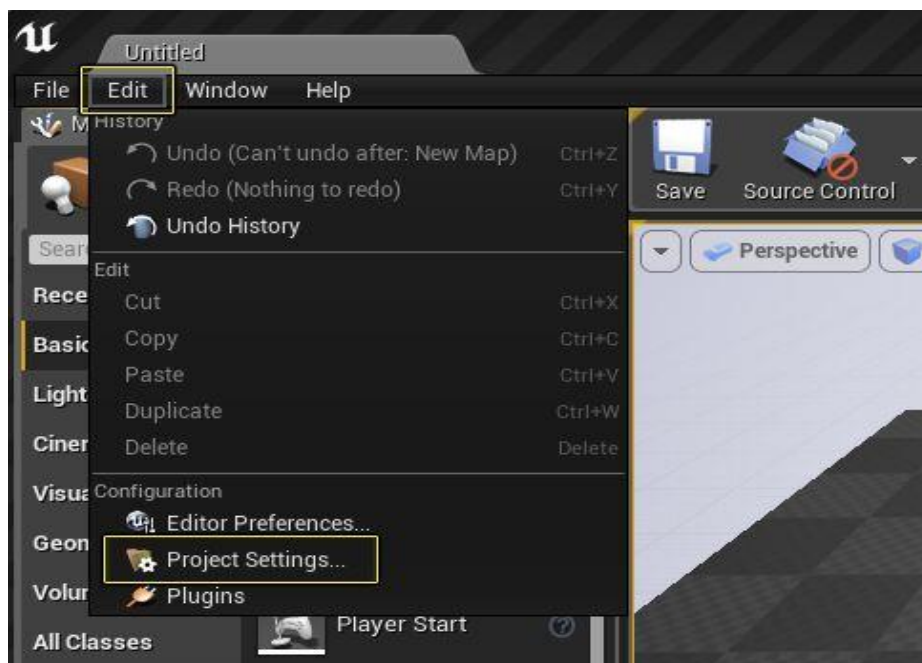
3. 选择主工具栏的 文件（File） 选项，然后点击“保存当前”选项呼出“关卡保存窗口”。



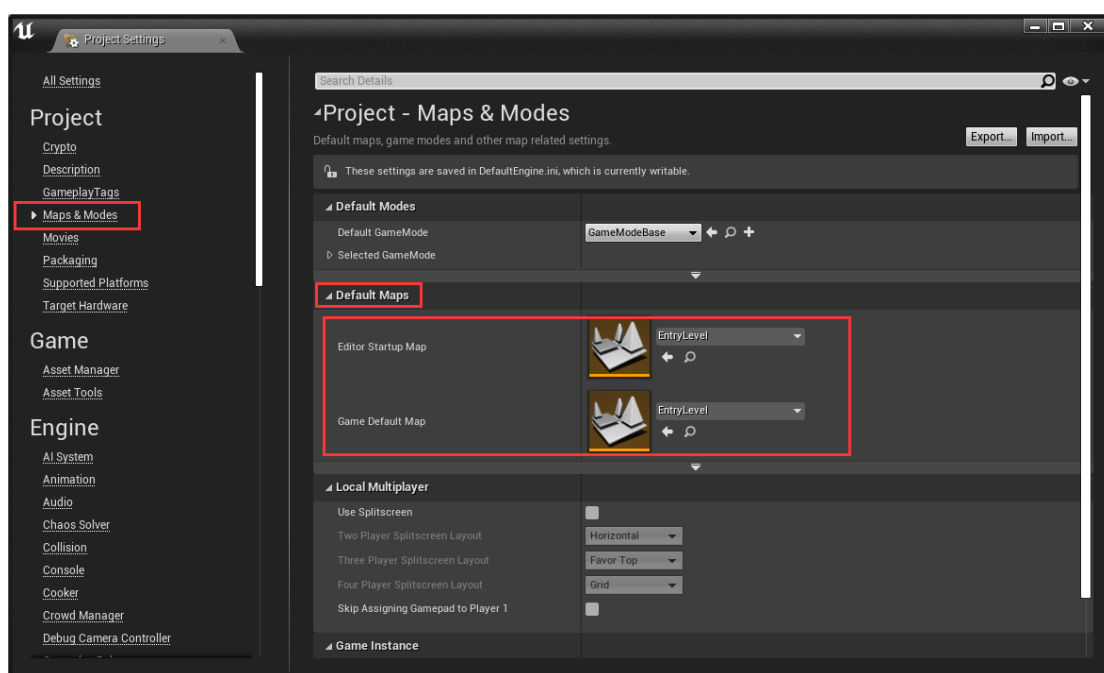
4. 在“关卡保存（Save Level As）”窗口下的 命名（Name） 部分中将关卡命名为 EntryLevel，并按下 保存（Save） 按钮。



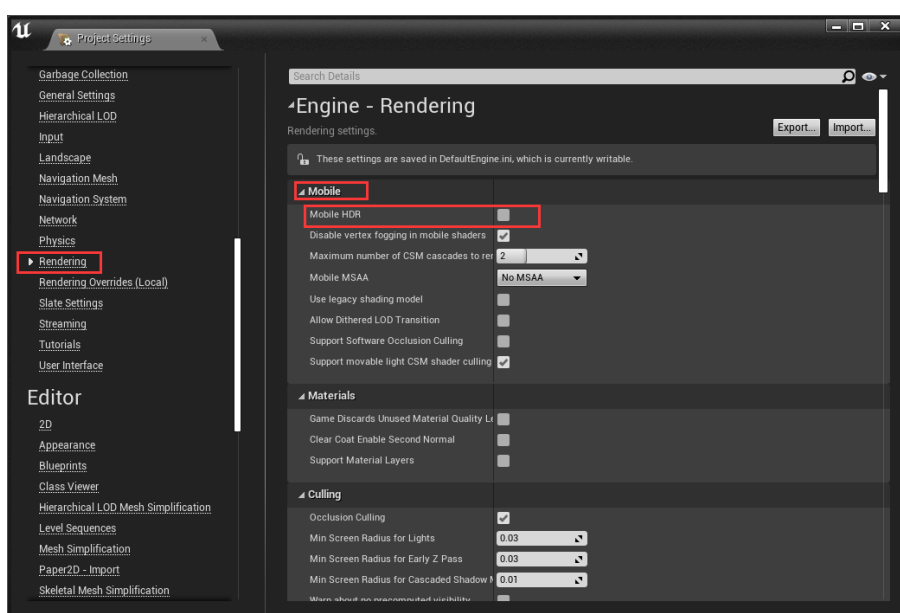
5. 返回 主工具栏 选择 编辑（Edit） 菜单项，然后选择 项目设置（Project Settings）。



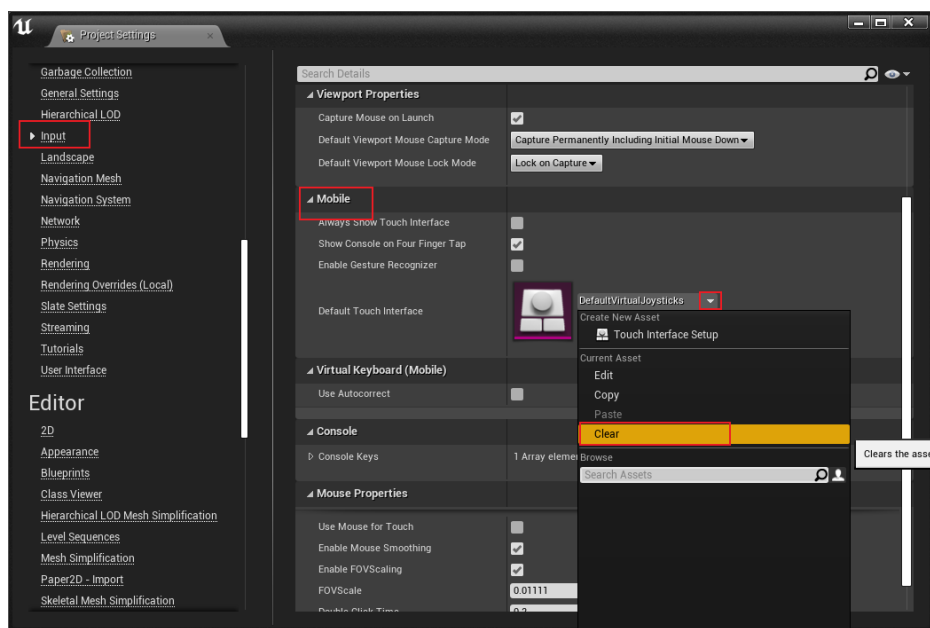
6. 在 项目设置 (Project Settings) 菜单 地图和模式 (Maps & Modes) 部分中的 默认地图 (Default Maps) 下, 将 编辑器启动地图 (Editor Startup Map) 和 游戏默认地图 (Game Default Map) 改为 EntryLevel 地图, 以便在项目运行时进行加载。



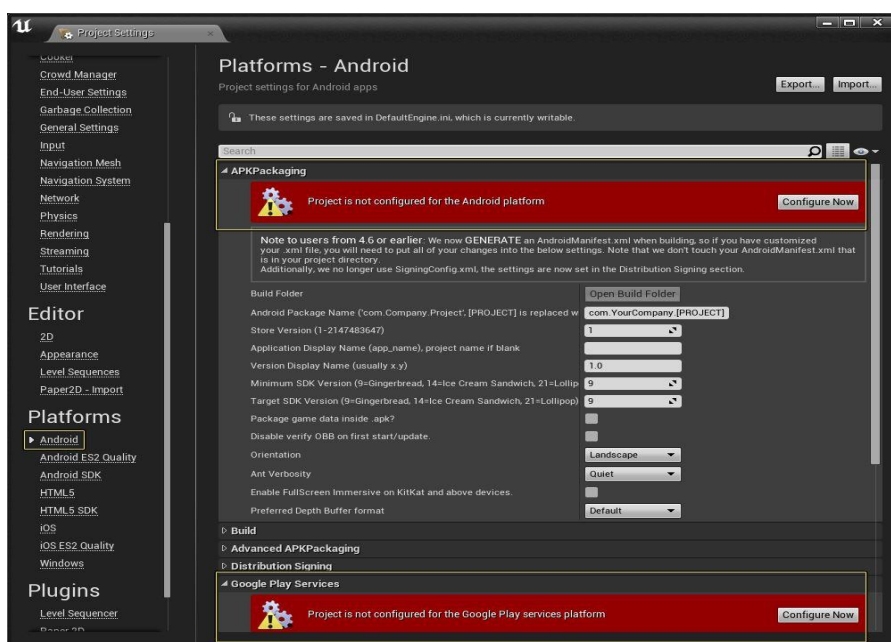
7. 将 渲染 (Rendering) 下的 移动 HDR (Mobile HDR) 禁用。这对于正确的立体渲染十分必要。

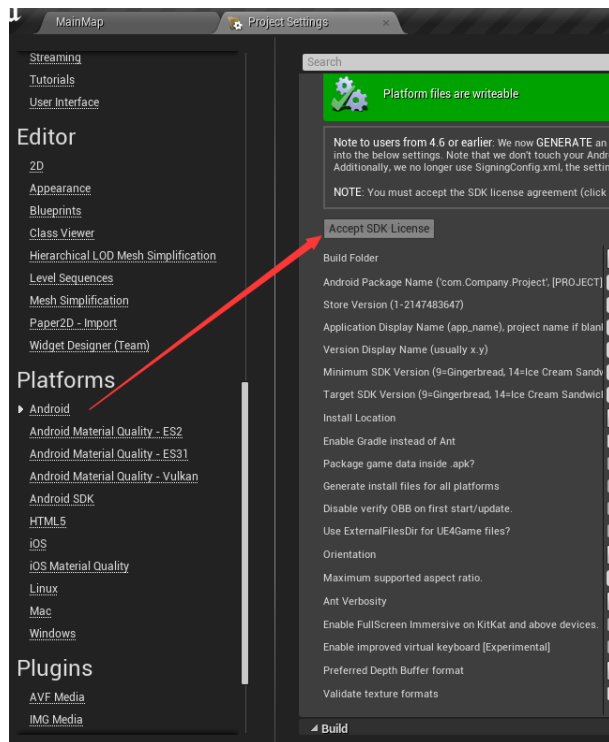


8. 在 移动平台（Mobile） 类目中的 输入（Input） 部分 ，从 默认触控界面（Default Touch Interface） 中移除 DefaultVirtualJoysticks——点击 DefaultVirtualJoysticks 旁的白色小三角形，然后从下拉菜单中选择 清除（Clear） 即可。



9. 在 APKPackaging 中, 点击 Config Now 按钮, 然后点击下方显示的 Accept SDK License 按钮, 以确保 AndroidSDK 的配置可以让你正常打包。

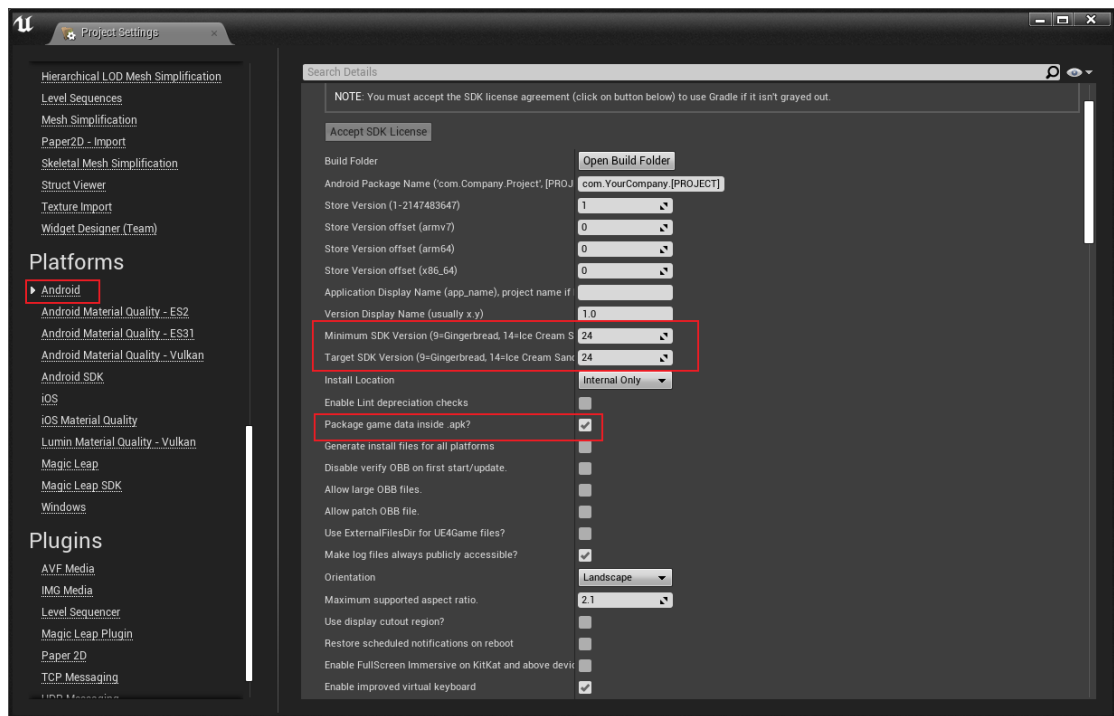




10. 关闭引擎，将 GSXR SDK 复制到项目路径下的 Plugins 文件夹下，如果没有该文件夹，需要手动创建一个。



11.在 APKPackaging 中进行以下选项的设置:



12.在 Android SDK 的 SDK Config 中, 需要下载 android SDK、NDK、

JDK 进行配置, 请参照 UE4 文档进行下载设置。

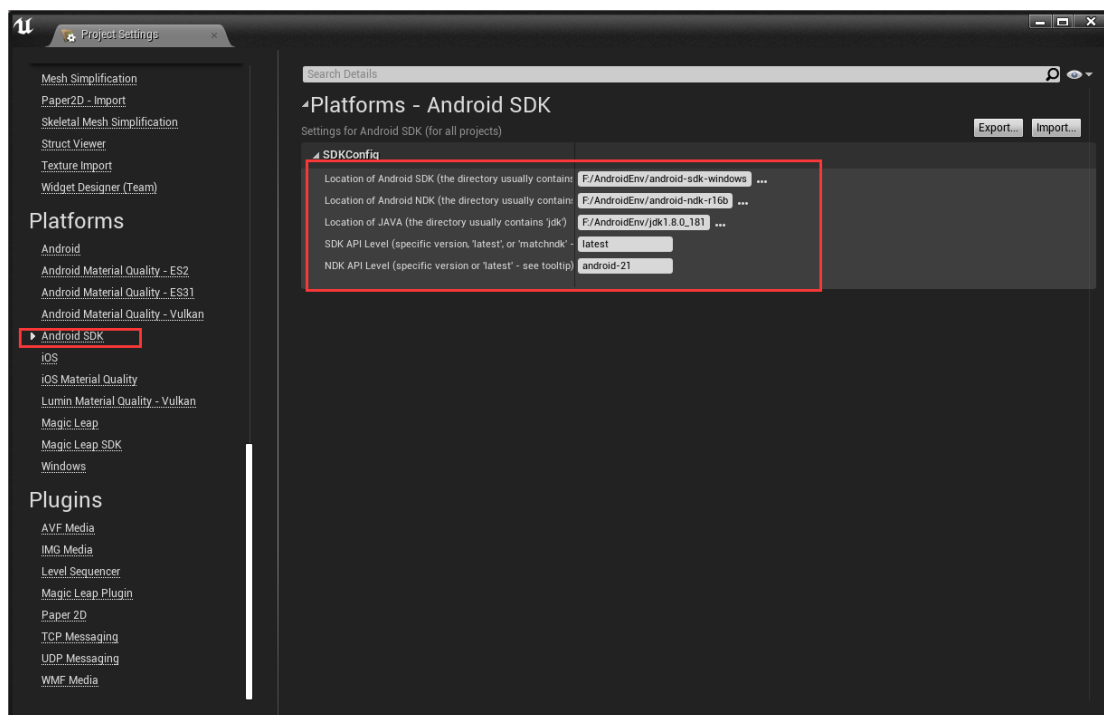
<https://docs.unrealengine.com/4.27/zh-CN/SharingAndReleasing/Mobile/Android/Setup/AndroidStudio/>

13.根据 UE4 文档操作，下载完毕 SDK、NDK、JDK 后。将 SDK API Level 设为 最新（latest）、并将 NDK API Level 设为 android-21，用最新 Android SDK 进行项目编译。关于 AndroidNDK，不同引擎版本有不同的要求，请根据实际引擎版本选择相应的 NDK 版本。

Android NDK兼容性

下表显示不同虚幻引擎版本所需或兼容的NDK版本。

虚幻引擎版本	所需的Android Studio版本	兼容NDK版本
4.26.2	Android Studio 4.0	NDK r21b
4.25	Android Studio 3.5.3	NDK r21b, NDK r20b
4.21 - 4.24		NDK r14b
4.19 - 4.20		NDK r12b



3 GSXR 按键行为映射

3.1 设计目的

近年来，伴随着社会的发展，VR、AR、MR（后续简称 XR）等高新行业也迎来了井喷时代，市面上出现了大大小小的 XR 厂商。各厂商从硬件到软件上都存在着大大小小的差异。同样，作为主要交互设备的手柄，各厂家的产品就更是五花八门了，无论是从形态上，还是从按键数量和按键类型上，可以说是各有各的特色。

因为手柄硬件设备存在着这样那样的不同，导致开发者在进行游戏开发时，需要考虑到所有想要支持的设备的手柄按键，同样需要针对每个设备出具不同的游戏使用文档，造成工作量的增长。

同时，由于现在游戏开发过程中，开发者进行游戏设计时，往往是按照惯例进行游戏开发，例如开发者会习惯性将右手柄的扳机键作为游戏内开枪行为的触发按键，但是如果对于左撇子来讲，就是特别别扭。手柄按键到游戏逻辑的绑定完全由游戏开发者决定，用户只能被动承受。

从上述内容可以总结出出现在 XR 手柄使用过程中的两大问题：

- 1、XR 手柄类型繁多，开发者不可能考虑到所有类型手柄的按键
- 2、XR 用户对游戏的按键和行为绑定没有干预能力。

为解决上述问题，故设计该按键行为映射方案。

3.2 实现流程

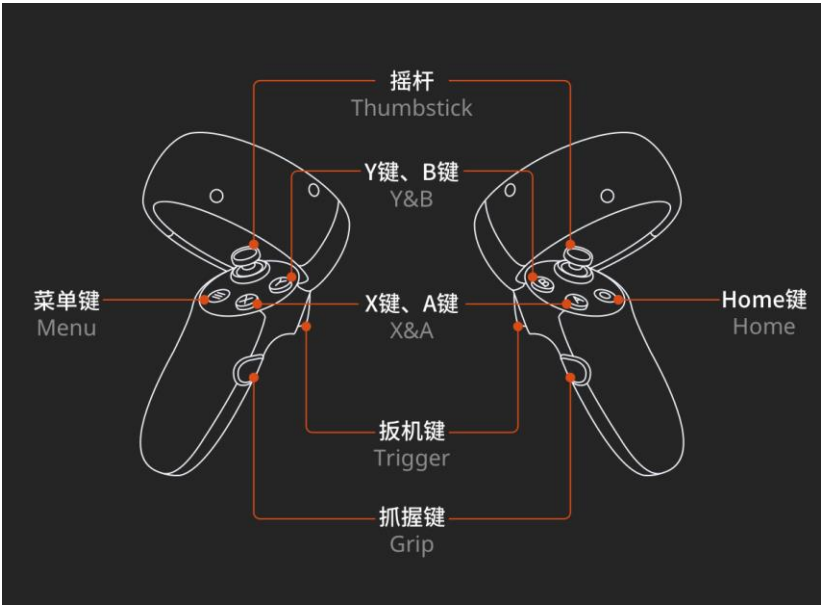
整个按键行为映射方案需要 Unreal SDK 与 GSXR Runtime 协作实现。SDK 会根据开发者定义的行为事件和按键的关系，生成一个配置文件。然后再将该文件打包入 APK 中，当游戏安装之后，SDK 将配置文件复制到特定路径下，这样 Runtime 层就可以访问到开发者定义的行为事件和按键的绑定关系了。然后 SDK 层会每帧通过行为事件的名字向 Runtime 层查询行为事件的状态，然后通

过行为事件的状态来驱动游戏逻辑，从而实现 **Runtime** 可以对游戏内逻辑的按键和行为事件的重载。

整个实施方案分为以下六个部分：

- 1、SDK 注册 **GSXR** 标准按键到 **UE4** 引擎。
- 2、开发者进行按键和行为事件的映射绑定
- 3、SDK 根据开发者配置的绑定生成对应的配置文件
- 4、SDK 将配置文件打包入 **APK**
- 5、应用安装后 **SDK** 将配置文件复制到特定路径下
- 6、SDK 通过行为名称向 **Runtime** 查询事件状态

Joystick 类型手柄



SDK 已经将该类型按钮注册入引擎，可见下表参考：

按钮名称	触发类型	引擎键值
左手 X 键	点击	GSXRJoystick(L) X Click
左手 X 键	触摸	GSXRJoystick(L) X Touch
左手 Y 键	点击	GSXRJoystick(L) Y Click
左手 Y 键	触摸	GSXRJoystick(L) Y Touch
左手抓握键	点击	GSXRJoystick(L) Grip Click
左手菜单键	点击	GSXRJoystick(L) Menu Click
左手扳机键	点击	GSXRJoystick(L) Trigger Click
左手扳机键	触摸	GSXRJoystick(L) Trigger Touch
左手摇杆键	点击	GSXRJoystick(L) Thumbstick Click
左手摇杆键	触摸	GSXRJoystick(L) Thumbstick Touch
左手摇杆键	上推	GSXRJoystick(L) ThumbstickUp Click
左手摇杆键	下推	GSXRJoystick(L) ThumbstickDown Click
左手摇杆键	左推	GSXRJoystick(L) ThumbstickLeft Click
左手摇杆键	右推	GSXRJoystick(L) ThumbstickRight Click
左手抓握键	键程	GSXRJoystick(L) Grip Axis
左手扳机键	键程	GSXRJoystick(L) Trigger Axis
左手摇杆横向 X 轴	键程	GSXRJoystick(L) ThumbstickX Axis
左手摇杆竖向 Y 轴	键程	GSXRJoystick(L) ThumbstickY Axis
右手 A 键	点击	GSXRJoystick(R) A Click
右手 A 键	触摸	GSXRJoystick(R) A Touch
右手 B 键	点击	GSXRJoystick(R) B Click

右手 B 键	触摸	GSXRJoystick(R) B Touch
右手抓握键	点击	GSXRJoystick(R) Grip Click
右手 Home 键	点击	GSXRJoystick(R) System Click
右手扳机键	点击	GSXRJoystick(R) Trigger Click
右手扳机键	触摸	GSXRJoystick(R) Trigger Touch
右手摇杆键	点击	GSXRJoystick(R) Thumbstick Click
右手摇杆键	触摸	GSXRJoystick(R) Thumbstick Touch
右手摇杆键	上推	GSXRJoystick(R) ThumbstickUp Click
右手摇杆键	下推	GSXRJoystick(R) ThumbstickDown Click
右手摇杆键	左推	GSXRJoystick(R) ThumbstickLeft Click
右手摇杆键	右推	GSXRJoystick(R) ThumbstickRight Click
右手抓握键	键程	GSXRJoystick(R) Grip Axis
右手扳机键	键程	GSXRJoystick(R) Trigger Axis
右手摇杆横向 X 轴	键程	GSXRJoystick(R) ThumbstickX Axis
右手摇杆竖向 Y 轴	键程	GSXRJoystick(R) ThumbstickY Axis

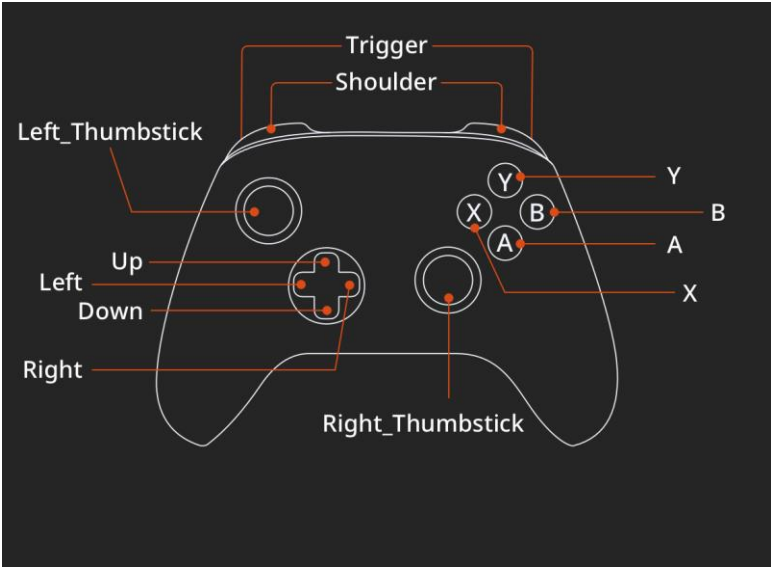
Touch 类型手柄



下表为 Touch 类型手柄按键与引擎键值的对应关系：

按键名称	触发类型	引擎键值
左手触摸板键	点击	GSXRTouch(L) Trackpad Click
左手触摸板键	触摸	GSXRTouch(L) Trackpad Touch
左手板机键	点击	GSXRTouch(L) Trigger Click
左手抓握键	点击	GSXRTouch(L) Grip Click
左手菜单键	点击	GSXRTouch(L) Menu Click
左手电源键	点击	GSXRTouch(L) System Click
左手触摸板键	上推	GSXRTouch(L) TrackpadUp Click
左手触摸板键	下推	GSXRTouch(L) TrackpadDown Click
左手触摸板键	左推	GSXRTouch(L) TrackpadLeft Click
左手触摸板键	右推	GSXRTouch(L) TrackpadRight Click
左手触摸板键横向 X 轴	键程	GSXRTouch(L) TrackpadX Axis
左手触摸板键竖向 Y 轴	键程	GSXRTouch(L) TrackpadY Axis
左手板机键	键程	GSXRTouch(L) Trigger Axis
右手触摸板键	点击	GSXRTouch(R) Trackpad Click
右手触摸板键	触摸	GSXRTouch(R) Trackpad Touch
右手板机键	点击	GSXRTouch(R) Trigger Click
右手抓握键	点击	GSXRTouch(R) Grip Click
右手菜单键	点击	GSXRTouch(R) Menu Click
右手电源键	点击	GSXRTouch(R) System Click
右手触摸板键	上推	GSXRTouch(R) TrackpadUp Click
右手触摸板键	下推	GSXRTouch(R) TrackpadDown Click
右手触摸板键	左推	GSXRTouch(R) TrackpadLeft Click
右手触摸板键	右推	GSXRTouch(R) TrackpadRight Click
右手触摸板键横向 X 轴	键程	GSXRTouch(R) TrackpadX Axis
右手触摸板键竖向 Y 轴	键程	GSXRTouch(R) TrackpadY Axis
右手板机键	键程	GSXRTouch(R) Trigger Axis

GamePad 类型手柄



下表为 GamePad 类型手柄按键与引擎键值的对应关系：

按键名称	触发类型	引擎键值
A 键	点击	GSXRGamePad A Click
B 键	点击	GSXRGamePad B Click
X 键	点击	GSXRGamePad X Click
Y 键	点击	GSXRGamePad Y Click
Up 键	点击	GSXRGamePad DpadUp Click
Down 键	点击	GSXRGamePad DpadDown Click
Left 键	点击	GSXRGamePad DpadLeft Click
Right 键	点击	GSXRGamePad DpadRight Click
Left_Thumbstick 键	点击	GSXRGamePad LeftThumbstick Click
Left_Thumbstick 键	上推	GSXRGamePad LeftThumbstickUp Click
Left_Thumbstick 键	下推	GSXRGamePad LeftThumbstickDown Click
Left_Thumbstick 键	左推	GSXRGamePad LeftThumbstickLeft Click
Left_Thumbstick 键	右推	GSXRGamePad LeftThumbstickRight Click
Right_Thumbstick 键	点击	GSXRGamePad RightThumbstick Click
Right_Thumbstick 键	上推	GSXRGamePad RightThumbstickUp Click
Right_Thumbstick 键	下推	GSXRGamePad RightThumbstickDown Click
Right_Thumbstick 键	左推	GSXRGamePad RightThumbstickLeft Click
Right_Thumbstick 键	右推	GSXRGamePad RightThumbstickRight Click
左 Shoulder	点击	GSXRGamePad LeftShoulder Click
右 Shoulder	点击	GSXRGamePad RightShoulder Click
左 Trigger	点击	GSXRGamePad LefTrigger Click
右 Trigger	点击	GSXRGamePad RightTrigger Click

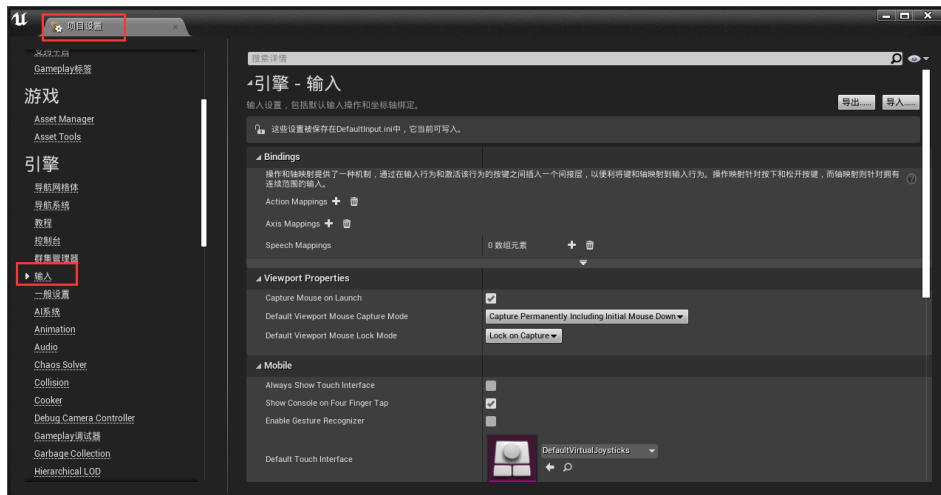
3.3 按键行为绑定

Unreal 引擎提供了行为事件和按键的绑定功能。开发者可以通过引擎提供的行为事件和按键的绑定功能进行行为事件和按键绑定。

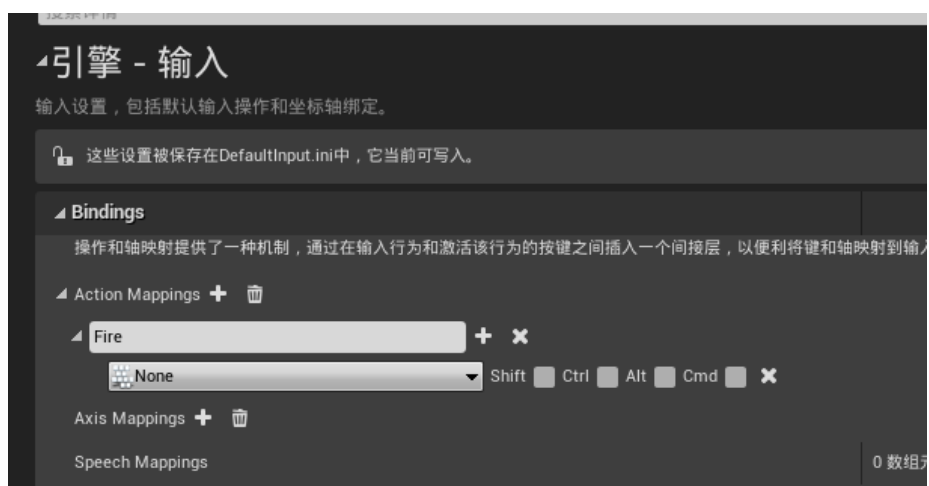
由于 **GSXR** 设计理念是一次开发多设备适配，建议开发者绑定输入按键时同时绑定三种输入设备对应按键。

具体设置如下：

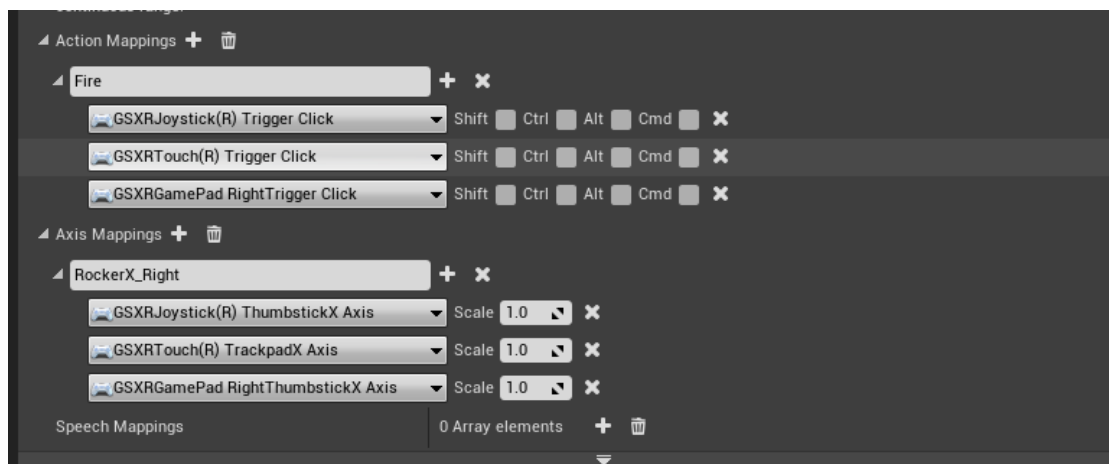
- 1、 打开 Unreal 引擎
- 2、 进入项目设置—引擎—输入



- 3、 点击 Action Mapping 或 Axis Mappings 右侧的加号定义行为事件



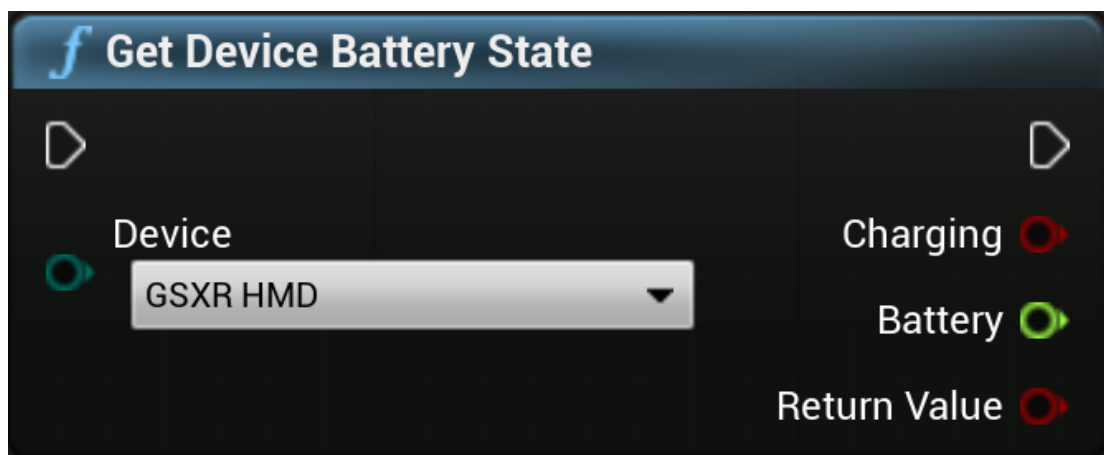
4、 在行为事件下绑定想要绑定的 GSXR 标准按键



在这里说明一下，**Action Mappings** 中进行绑定布尔类型的按键，例如按键点击事件或触摸事件，非此即彼的类型。**Axis Mappings** 中绑定带有 **float** 类型的按键，例如手柄扳机键扣动的进度，手指在触摸板水平 **X** 轴上的位置等。同一个行为事件可以绑定多个按键。

4 蓝图接口

4.1 获取设备电量状态接口

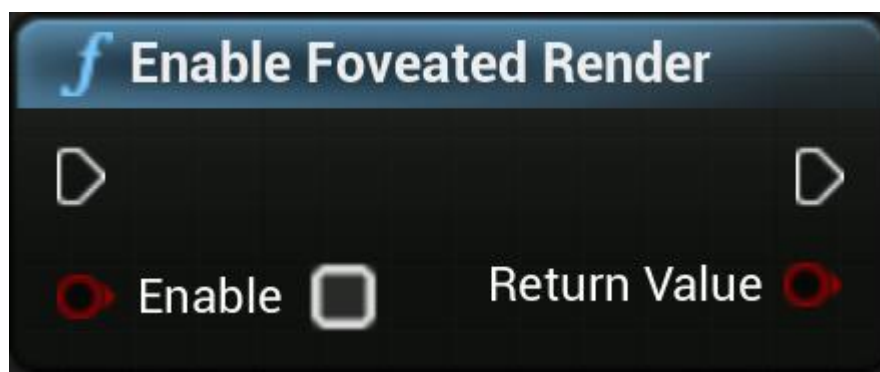


图例 1 获取设备电量状态

输入参数：Device，可输入 GSXR HMD，GSXR Left Hand，GSXR Right Hand，通过 Device 不同，获取不同设备的电量状态。

输出参数：Charging，设备是否在充电状态；
Battery，设备电量
Return Value，接口是否调用成功。

4.2 开启/关闭注视点渲染



图例 2 开启/关闭注视点渲染

输入参数：Enable，开启或关闭注视点渲染。

输出参数：Return Value，接口是否调用成功。

4.3 获取设备连接状态



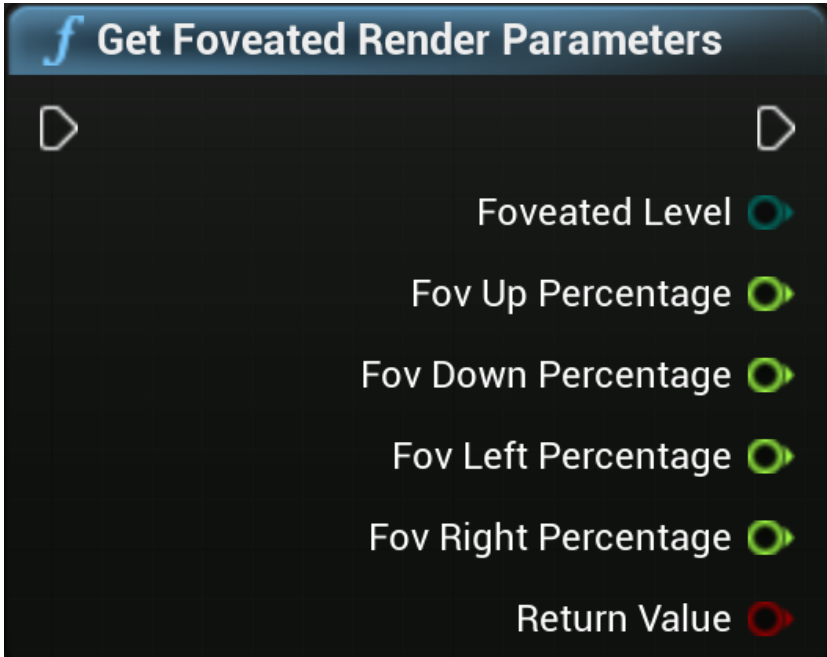
图例 3 获取设备连接状态

输入参数：Device，可输入 GSXR HMD，GSXR Left Hand，GSXR Right Hand，通过 Device 不同，获取不同设备的连接状态。

输出参数：Connected，设备连接状态

Return Value，接口是否调用成功。

4.4 获取注视点渲染参数



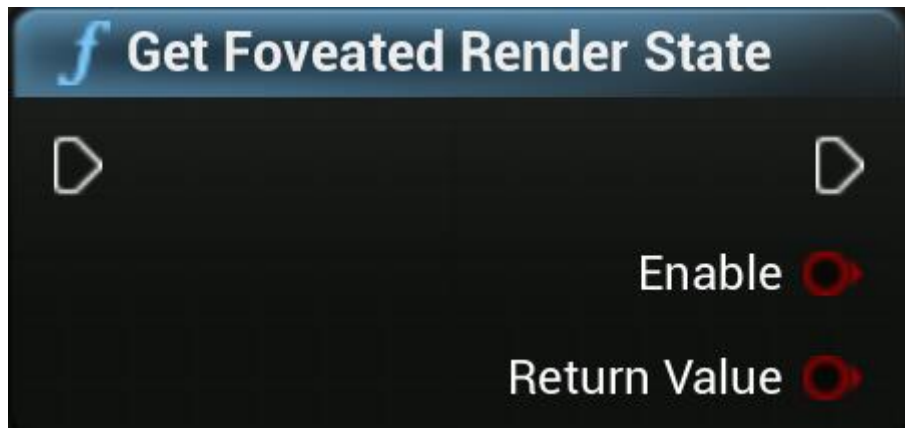
图例 4 获取注视点渲染参数

输入参数：无

输出参数：注视点渲染相关参数。

Return Value，接口是否调用成功。

4.5 获取注视点渲染功能状态



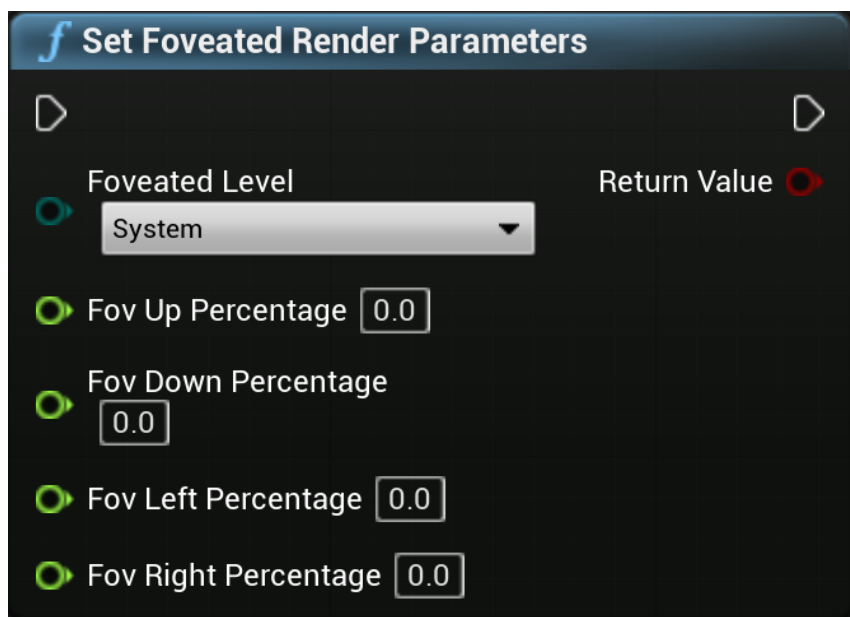
图例 5 获取注视点渲染功能状态

输入参数：无

输出参数： Enable，是否开启。

Return Value，接口是否调用成功。

4.6 设置注视点渲染相关参数



图例 6 设置注视点渲染相关参数

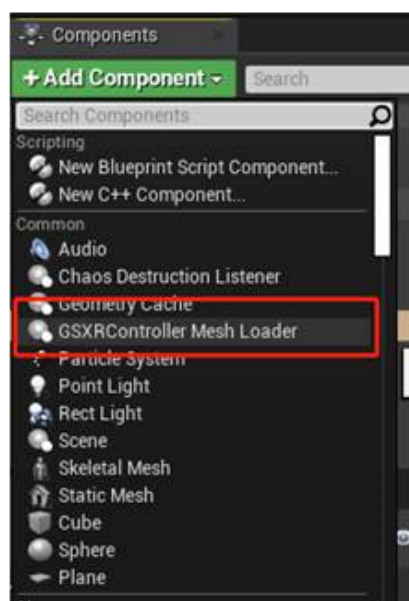
输入参数：注视点渲染相关参数

输出参数： Return Value，接口是否调用成功。

5 动态手柄模型加载

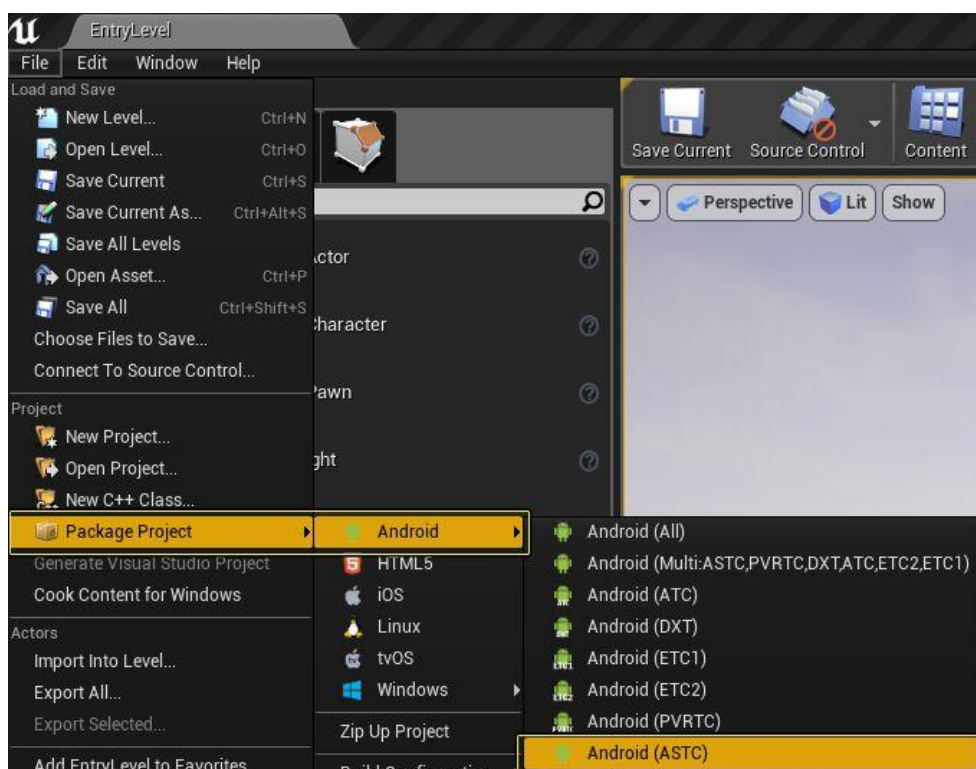
当开发者开发过程中需要显示手柄模型时，可以在蓝图中加入 GsxrControllerMeshLoader 组件，然后选择该组件为左手或则右手。

这个组件会根据当前应用运行设备自动识别对应手柄进行显示。

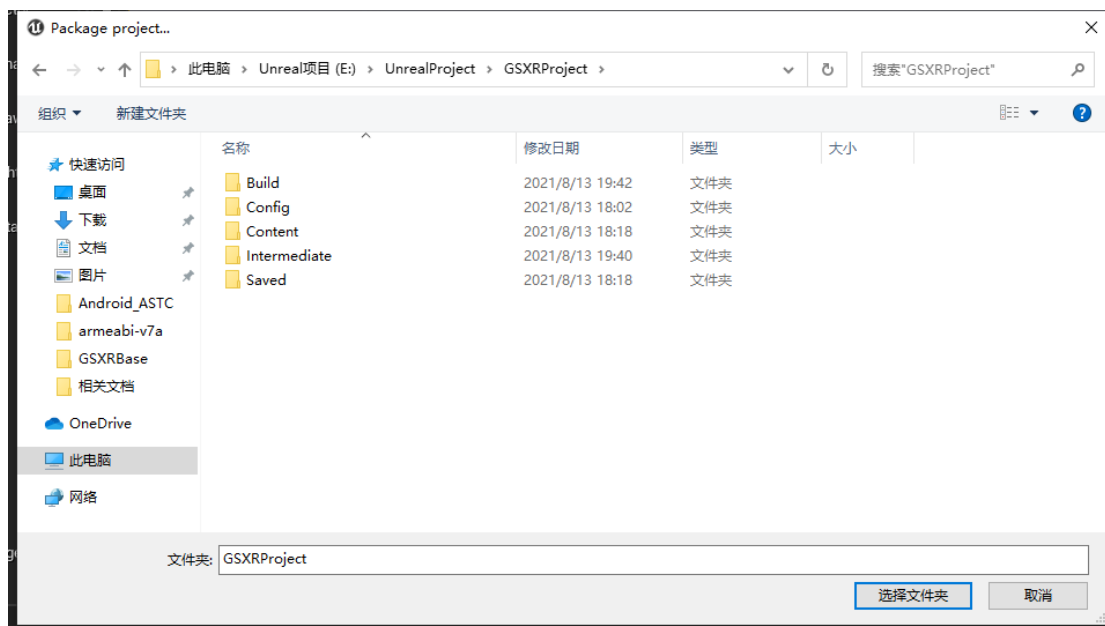


6 项目打包

1. 前往 主工具栏 的 文件（File） 菜单，选择 打包项目（Package Project） > Android > Android (ASTC) 开始打包。



2. 选择 **Android (ASTC)** 格式后将出现一个 **Windows** 对话框。选择 **APK** 的目标路径，然后点击 **选择文件夹** 开始打包进程。



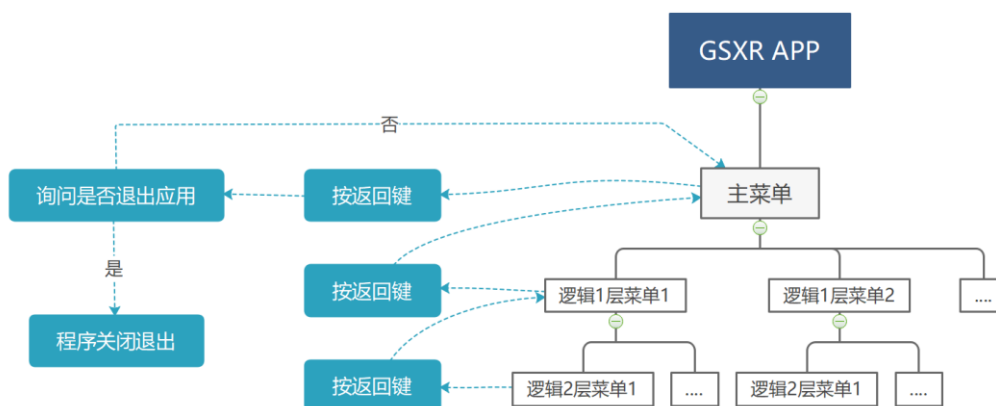
3. 项目打包完成后，会生成一个 **APK** 安装包以及两个批处理文件，可以通过运行批处理文件对应用进行安装或卸载。

 GSXRBase-armv7-es2.apk	2021/8/31 12:32	APK 文件	128,300 KB
 Install_GSXRBase-armv7-es2.bat	2021/8/31 12:32	Windows 批处理...	2 KB
 Uninstall_GSXRBase-armv7-es2.bat	2021/8/31 12:32	Windows 批处理...	1 KB

7 注意事项

7.1 设计返回逻辑

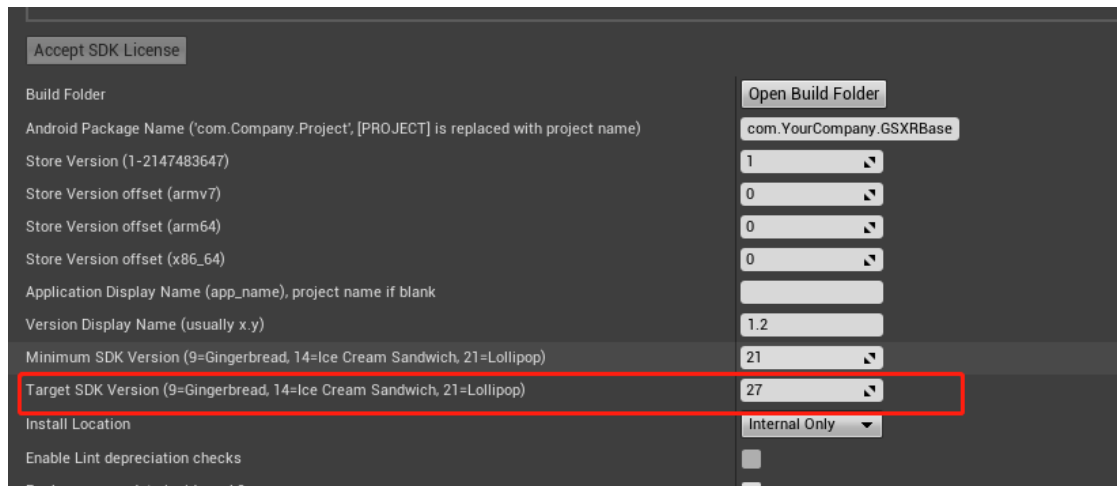
GSXR 内容上线 GSXR 平台需设计对应的返回键退出应用逻辑，如在业务设计中存在业务返回逻辑，当业务处于最外层时退出应用。



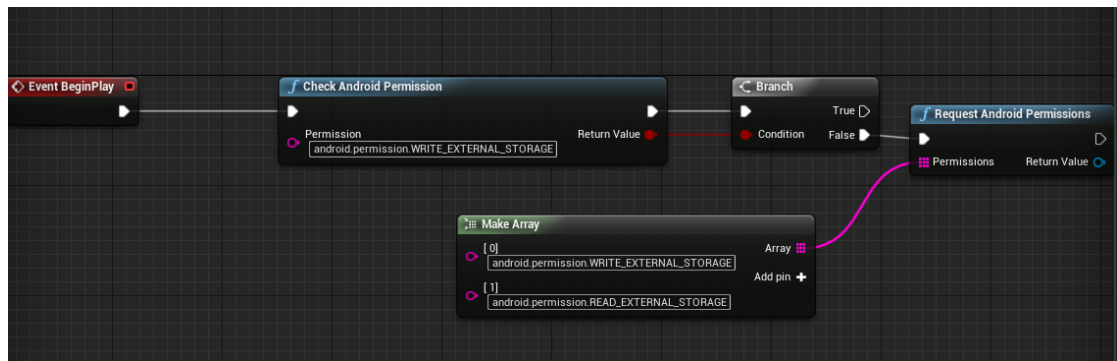
GSXR 应用返回逻辑参考图

7.2 保存游戏

安卓 6.0（API Level 23）后增加了权限管理之后，不仅需要应用申请权限，还需要用户在弹窗提示中允许之后才生效。



如果 target SDK version 选择了 22 以上的数字，则会导致 savegame 失败，解决办法如下。



这样会在应用运行时弹出窗口申请权限，需要用户同意，如果不同意则仍不能保存游戏。