VRTRIX
无 远 弗 届

VRTRIX Data Glove UE4 SDK Tutorial

Date	Modified by	Comments
2018-10-22	Guo	Init Version.
2019-09-06	Guo	Update Manual For VRTRIX Data Glove Pro

简介

Introduction

VRTRIX™ 数据手套通过遍布全手的高性能 9 轴 MEMS 惯性传感器实时采集各指头关节运动数据，并通过反向动力学还原骨骼运动，可以在虚拟现实的场景中实现对真实手部运动的重现，并进行精细的手部运动还原和交互。每只手套上分布有 6 个传感器，双手共 12 个，可以实时高精度低延迟输出全手所有关节的运动姿态。

VRTRIX™ 数据手套提供 UE4 SDK 以供开发者在 Unreal 平台下接入数据手套硬件进行驱动。支持模型动作实时渲染，手势识别，虚拟现实支持，虚拟现实环境下的交互以及与全身动捕的整合。开发者可以通过 Unreal SDK 中开放的 API 接口和场景示例工程，与原有虚拟现实项目进行整合，或者与全身动捕设备进行整合。

VRTRIX™ 数据手套 UE4 SDK 目前包含有 4 个示例场景：

- VRTRIXGloveDataStreaming (主要展示了对数据手套硬件数据流的获取与实时渲染，同时展示了触发手套震动反馈，以及 VR 环境下如何与 HTC tracker 腕部追踪定位结合。)
- VRTRIXGloveGestureDetection (主要展示了数据手套手势识别的功能)
- VRTRIXGloveVRInteraction(主要展示了如何在 VR 环境下实现手套与虚拟物体的交互)。
- VRTRIXGloveLiveLink(主要展示了如何通过动画蓝图实现手套直接驱动骨架模型 (Skeletal Mesh))

系统要求

System Requirements

- UE 4.18 及以上版本，目前最新测试兼容 UE4.22。
- Windows 10 及以上版本。

开发准备

Prepare for Development

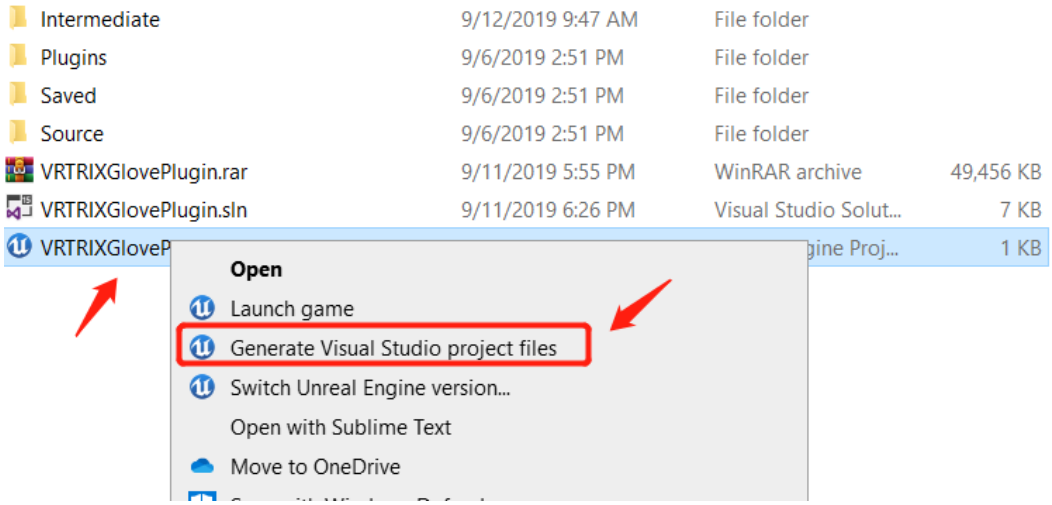
1. 请首先确保数据手套驱动软件已经成功安装，且数据手套已经成功配对，HTC Tracker 腕部追踪器也已经成功设置（可选）。如果还未进行该操作，请按数据手套操作手册安装软件和操作后再进行下面的步骤。
2. 对于希望在 VR 环境下运行数据手套的用户，需要先配置好 VR 环境，本 UE4 SDK 默认采用 HTC Vive 头盔，以及 SteamVR 开发环境。具体请见：
   https://www.vive.com/cn/support/
3. 数据手套 UE4 SDK 最新版会在 https://github.com/VRTRIX/VRTRIXGlove_UE4_SDK/releases 上进行更新，目前支持 UE4.18 及以上版本，请在 Release 页面下选择正确的版本下载。该插件所有源码均为开源，如果需要其他版本的插件可以自行编译，具体编译方法请见快速开始中的介绍。数据手套 UE4 SDK 包含三个压缩包，其中：
   VRTRIXGlove_UE4_SDK.rar 为数据手套开发者插件，可以用于获取手套数据流，并在蓝图中为 PoseableMesh 赋值驱动模型。
   VRTRIXGlove_UE4_Interaction_SDK.rar 为数据手套 VR 环境下与 HTC Tracker 腕部追踪结合的交互 Demo，该 Demo 使用了上述数据手套开发者插件，为纯蓝图工程。
   VRTRIXGloveLiveLink.rar 为数据手套 LiveLink 插件，支持将数据手套数据流通过 LiveLink 导入 UE4 引擎，并通过动画蓝图驱动模型。

快速开始

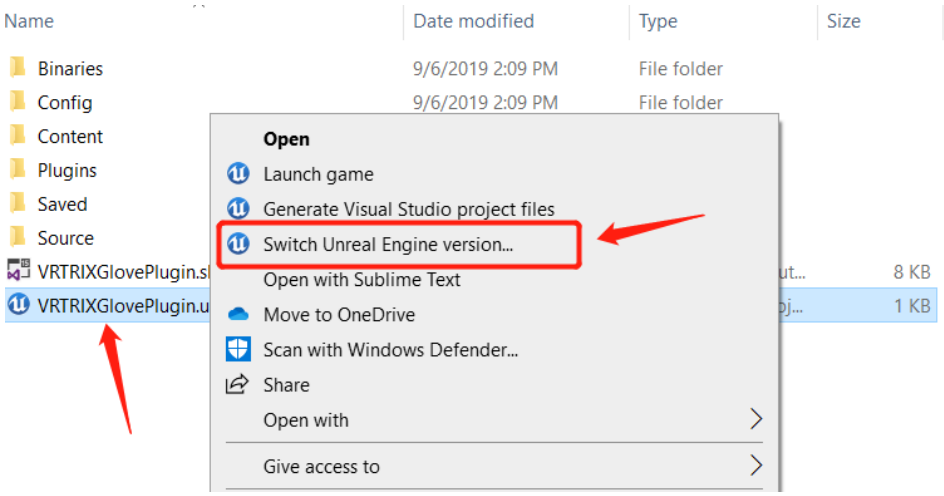
Quick Start

1. **下载手套插件：**请在 Github 代码仓库 Release 页面中下载手套插件包（VRTRIXGlove_UE4_SDK.rar）。请注意，Github 代码仓库主页中的工程为 UE4.18 平台下，代码仓库 Release 页面下提供 UE4.18-4.21 版本的带有手套插件的工程，如果需要其他版本，则如上快速开始文档中所述进行重新编译即可。

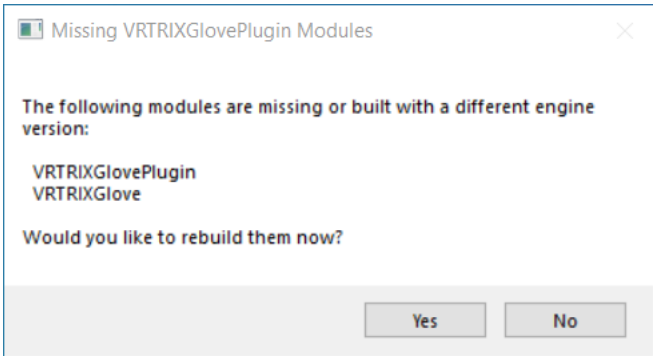
2. **VS 工程生成：**下载下来的 UE 插件包为了便于 git 管理，只包含源码和依赖库，不包含 VS 工程，可以通过 UE 自动生成 VS 工程。在下载好的 UE4 工程文件上右键点击，并选择 Generate Visual Studio project files，等待工程直接生成即可。



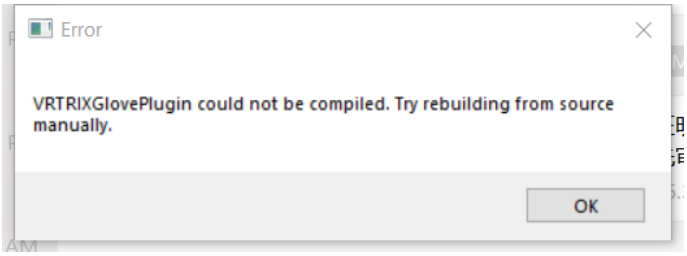
3. **UE4 版本更换及插件编译：**Github 代码仓库中提供 UE4.18-4.21 版本的 SDK 工程，如果需要其他版本，则需重新编译插件。在下载好的 UE4 工程文件上右键点击，并选择 Switch Unreal Engine version，在弹出的窗口中选择想要变更的 UE4 版本，然后点击确定。等待 UE4 自动重新生成工程文件。



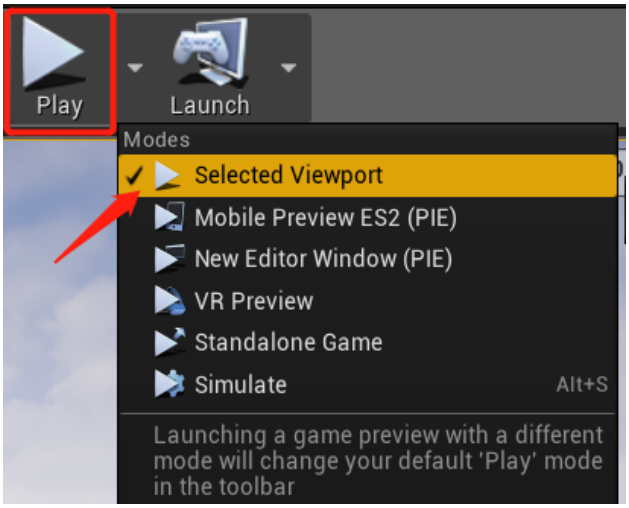
该步骤完成后双击打开 UE4 工程文件，将会提示插件需要重新编译，点击 Yes 进行插件重新编译。



如果编译成功则工程更新完成，会自动打开，此时执行下一步即可。如果报错编译失败，则需要手动打开 VS 工程，查看编译错误原因，解决后手动进行编译。

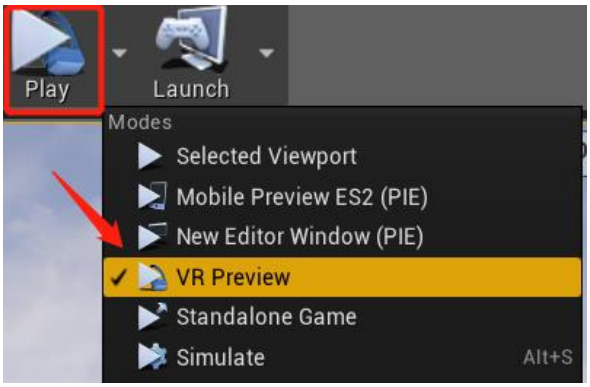


4. **运行 3D 场景：**默认场景为 VRTRIXGloveDataStreaming，点击场景右上方的 Play 按钮，确认选择的是普通运行模式，就可以开始运行。



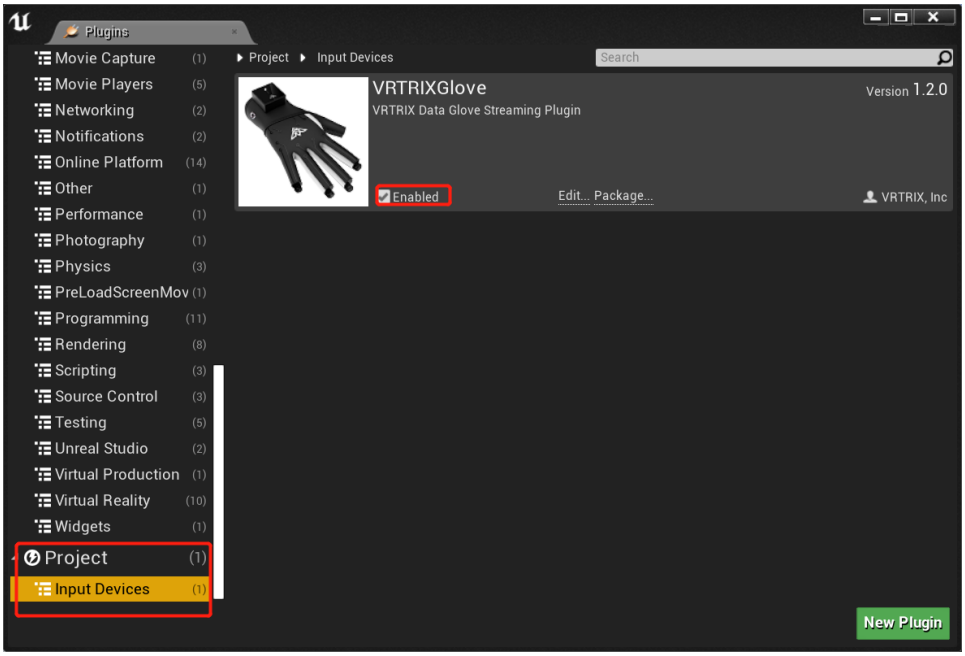
场景运行后手套会自动被连接，开始进行实时数据的传输和渲染。将左手朝前平举伸直，点击键盘“L”按键，可以将左手朝前对齐。同样的，将右手朝前平举伸直，点击键盘“R”按键，可以将右手朝前对齐。点击鼠标左键可以触发左手手心震动，点击鼠标右键可以触发右手手心震动。

5. **VR 环境使用：**首先确认 HTC tracker 已经固定好在腕部并处于打开状态，tracker 上方 LED 灯绿灯常亮。点击场景右上方的 Play 按钮，确认选择的是 VR 运行模式，就可以开始运行。



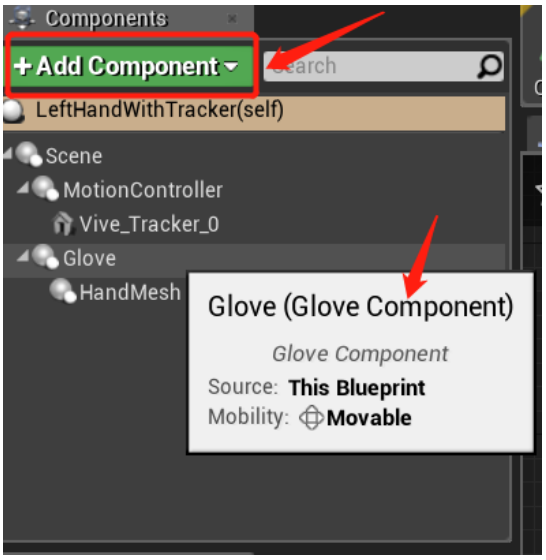
场景运行后手套会自动被连接，在 HTC Vive 头盔中就可以显示出实时渲染的手部动作，以及腕部追踪器的实时位置。将左手朝前平举伸直，点击键盘“L”按键，可以将左手朝前对齐。同样的，将右手朝前平举伸直，点击键盘“R”按键，可以将右手朝前对齐。点击鼠标左键可以触发左手手心震动，点击鼠标右键可以触发右手手心震动。

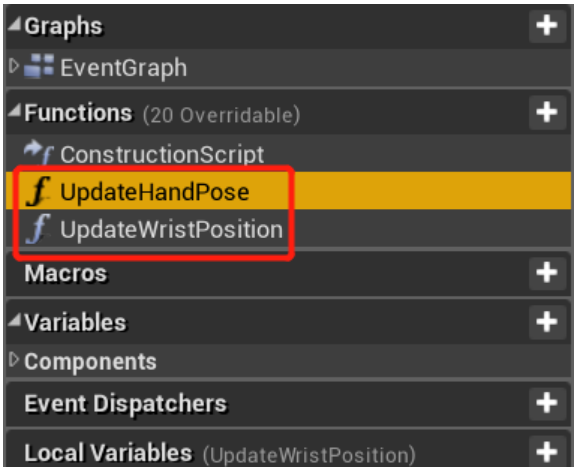
6. **向原有工程添加插件：**首先将 Github 代码仓库下载下来的 UE4 插件工程中的 Plugins 文件夹整个复制到原有工程的根目录下。如果原有工程已有 Plugins 文件夹，则选择合并即可。然后打开原有工程，UE4 编辑器会提示发现新插件，此时在编辑器窗口点击 Edit-Plugins 打开插件管理窗口，下拉至最后 Project 插件，应该看见 VRTRIXGlove 插件，并保证其左下角 Enabled 已经勾选。



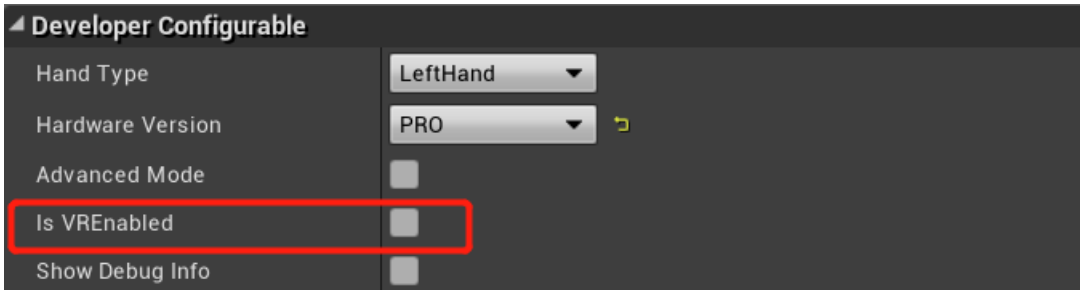
7. **使用插件驱动手套：**模仿 demo 工程场景 VRTRIXGloveDataStreaming，先创建一个 Actor 蓝图， 然后在蓝图编辑器中添加一个 Glove Component， 并将手部模型作为其子物体。从 demo 工程中将 UpdateHandPose 函数复制到该工程中，并根据手部模型替换文档部分的内容对手部模型进行匹配和调整。之后将该 Actor 蓝图拖入场景中，运行场景就可以开始手部渲染。

   如果要与 VR 项目进行整合，则需要在 Actor 蓝图中添加 MotionController 物体，同时模仿 demo 工程添加 UpdateWristPosition 函数，对手部模型腕部位置进行更新。



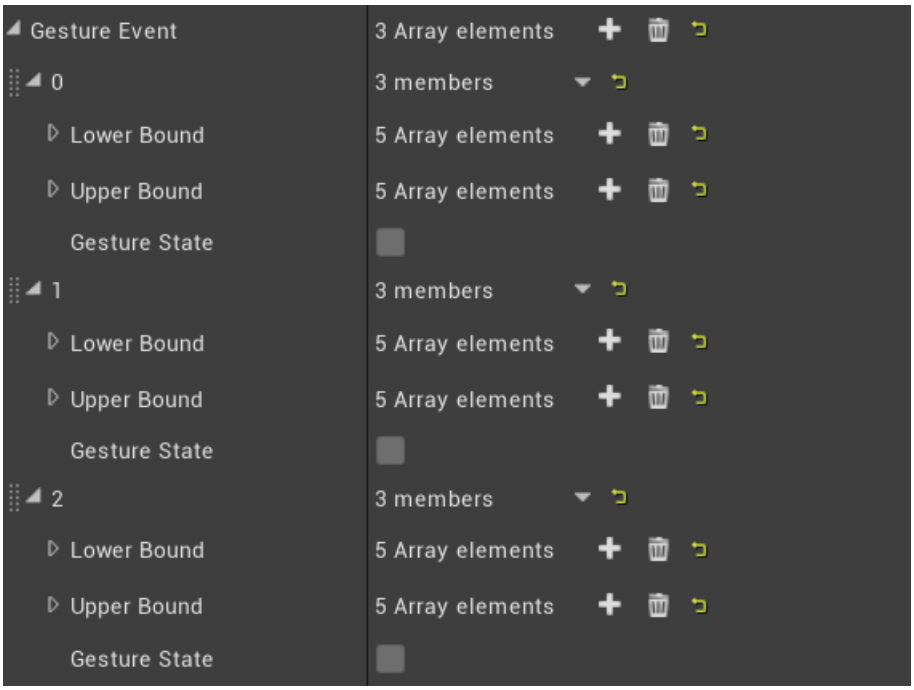


注意：如果不需要使用 VR 环境，则需要在插件选项卡里去除 VR Enabled 勾选。同样的，如果需要在 VR 环境下使用，则需要在插件选项卡里勾选 VR Enabled。

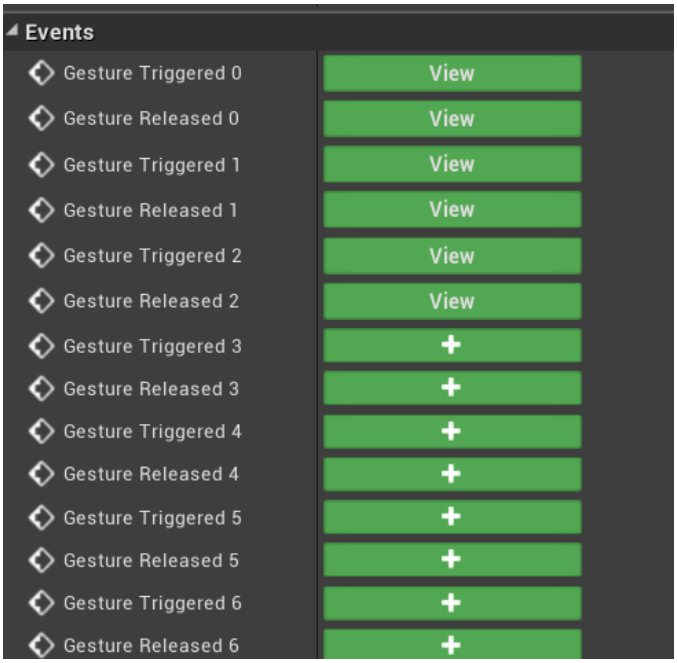


8. **使用插件自定义手势识别：**模仿 demo 工程场景 VRTRIXGloveGestureDetection，首先添加手势识别条件。插件提供的 Glove Component，右侧 detail 列表下有一栏为 Developer Configurable，此处的参数为插件对外开发的参数接口，其中 Gesture Event 为手势识别触发条件的设置，下图表示设置了三个手势的触发条件。

   手势触发条件根据五指和手背之间弯曲角度来定义，LowerBound 表示弯曲角度的下限，即该手势要求手指和手背弯曲角度大于该值，UpperBound 表示弯曲角度的上限，即该手势要求手指和手背弯曲角度小于该值。如果填写 0，则表示不关心该手指的弯曲上限/下限。



定义完手势触发条件之后，点击下方 Events 部分添加对应的事件函数，即满足该手势触发条件后调用的函数。插件默认第一个手势触发条件对应 Gesture Triggered/Released 0，以此类推。点击后蓝图中会出现一个自定义的事件，在此处实现自定义的函数就能在手势触发后被调用。



VR 环境下交互 Demo

Interaction Demo in VR

1. **下载数据手套 VR 交互 Demo：**请在 Github 代码仓库 release 中下载数据手套 VR 交互 Demo（VRTRIXGlove_UE4_Interaction_SDK.rar）。Github 代码仓库提供 UE4.18-4.21 版本工程，如果需要其他版本，则如上快速开始文档中所述进行切换即可。
2. **运行场景：**在打开工程后，请先打开双手手腕上的 tracker，并确保 tracker 上的绿灯亮起，然后在 VR Preview 下运行工程，运行时请双手朝前伸直，或者进入场景后将手朝前伸直点击 L 键对齐左手手腕位置，点击 R 键对齐右手手腕位置。场景中主要包含以下几大交互部分：

传送（Teleport）：左右手任意一只手做出传送手势后，会出现一道弧线用于提示传送目的地，选定目的地后松手进行传送。

抓取和投掷（Throwable）：左右手任意一只手触碰到可投掷物体且做出抓取手势后，可以将可投掷物体抓取在手上。松手时默认物体有重力，且以离开手瞬间的速度为初速度进行运动，即投掷行为。

点击（Click）：左右手任意一只手触碰到 UI 按钮且做出点击手势，UI 按钮的功能就会被触发。

开枪（Gun）:左右手任意一只手做出开枪手势，弯曲拇指则为抠动扳机。

以上所述的手势定义如下：

传送（Teleport）

抓取（Grab）

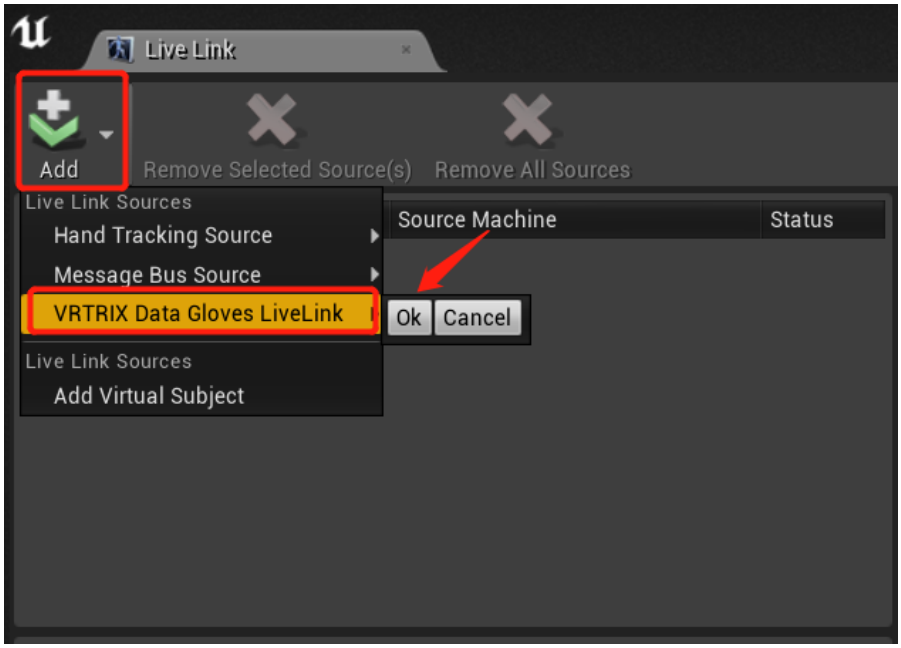
点击（Click）

开 枪 （ Gun ）

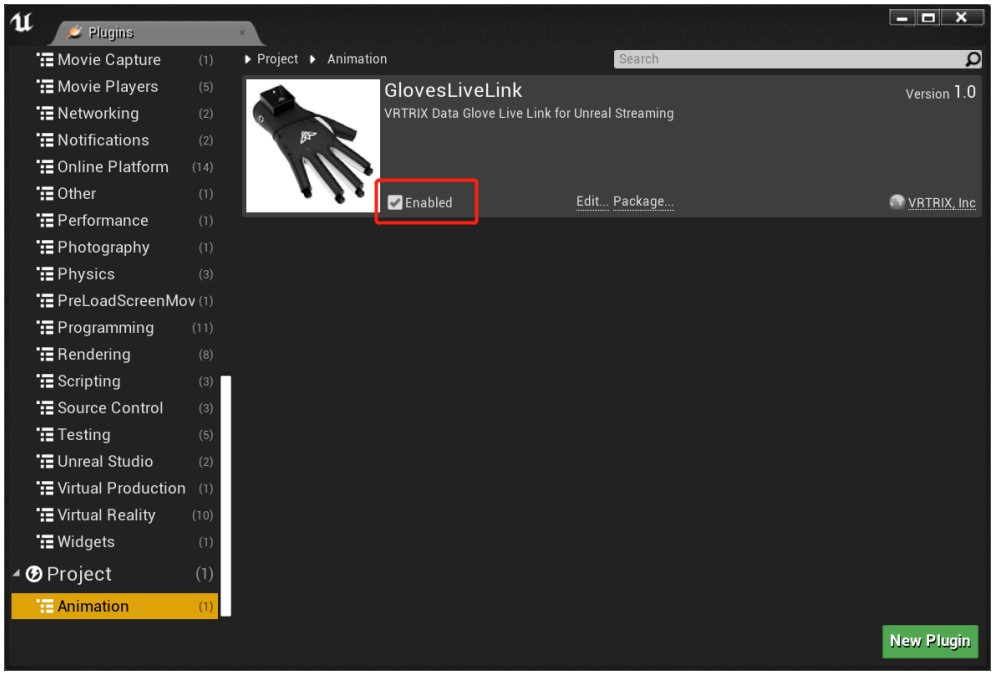
Live Link 驱动动画蓝图

Live Link Support with Animation Blueprint

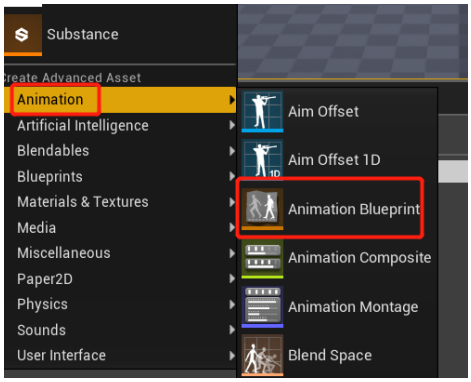
1. **下载 Live Link 插件：**请在 Github 代码仓库 release 中下载支持 UE4 Live Link 的手套插件包。Github 代码仓库提供 UE4.18-4.21 版本的带有手套 LiveLink 插件的工程，如果需要其他版本，则如上快速开始文档中所述进行重新编译即可。
2. **添加 Live Link 源：**在 UE4 编辑器中打开含有手套 LiveLink 插件的工程，点击 Window-Live Link 进入 Live Link 窗口。点击 Add 然后选择 VRTRIX Data Gloves LiveLink，点击 Ok 确定。

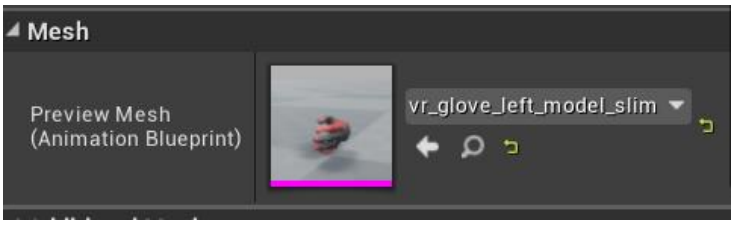


3. **观看模型渲染动画及运行场景：**在添加成功 Live Link 源之后，手套就会自动被连接上，可以先打开左右手的动画蓝图 ANIM_Gloves_L 和 ANIM_Gloves_G 观看手部模型渲染动画是否正常，然后运行默认场景，手部模型就会在动画蓝图的驱动下在场景中实时渲染。
4. **向原有工程添加 Live Link 插件和动画蓝图：**首先将 Github 代码仓库下载下来的 LiveLink 插件工程中的 Plugins 文件夹整个复制到原有工程的根目录下。如果原有工程已有 Plugins 文件夹，则选择合并即可。然后打开原有工程，UE4 编辑器会提示发现新插件，此时在编辑器窗口点击 Edit-Plugins 打开插件管理窗口，下拉至最后 Project 插件，应该看见 GlovesLiveLink 插件，并保证其左下角 Enabled 已经勾选。

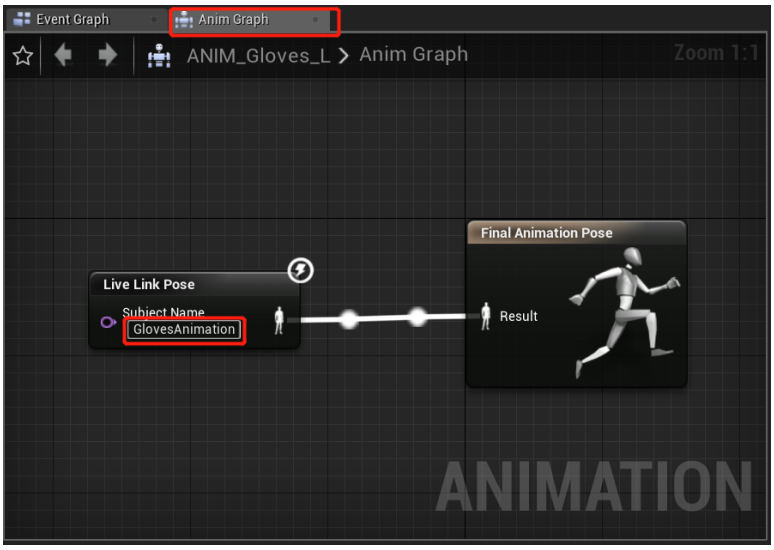


5. **在工程中添加动画蓝图：**首先创建一个动画蓝图，并在动画蓝图的模型选择框中选择要使用的模型，注意该模型暂时无法提供更换功能，需要更换 Live Link 驱动的模型需要自行修改插件代码。





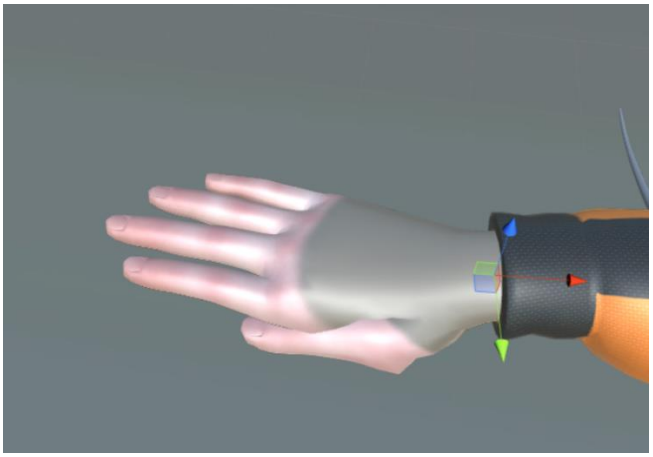
6. **在动画蓝图中添加节点：**首先添加 live link pose 节点，然后在 SubjectName 中填入 GlovesAnimation，然后将该节点连接到 Animation Pose 节点，随后重复第二步的流程添加 Live Link 源，模型就可以开始实时渲染了。



手部模型替换

Use Custom Models

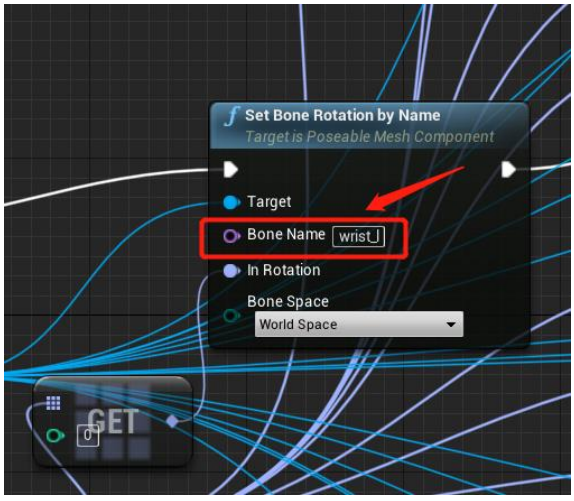
1. **全身/手部模型要求：**要替换手部模型，第一步就是将数据手套的手部骨骼定义与新模型手部骨骼定义相匹配，为了实现最佳效果，要替换的模型手部骨骼及模型需满足以下要求：
   - 手腕关节及以下所有子关节骨骼轴向一致（即选中每个关节时，坐标系方向相同）。
     - 手腕关节及以下所有子关节局部旋转为 0 时，模型姿态应该呈现五指并拢，平伸朝前，且与手背平齐，五指的指甲盖朝上（包含大拇指在内）如下图所示：

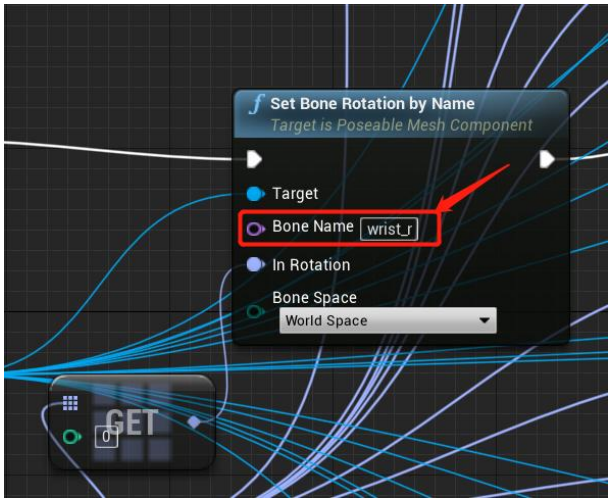


2. 骨骼名称匹配：

将数据手套定义的手部骨骼名称与全身模型的手部骨骼名称匹配可以通过 LeftHandWithTracker 和 RightHandWithTracker 蓝图中 UpdateHandPose 蓝图函数直接在蓝图中对骨骼名称进行匹配。注意将左手的骨骼名称替换至 LeftHandWithTracker 蓝图中的 UpdateHandPose 函数下，将右手的骨骼名称替换至 RightHandWithTracker 蓝图中的 UpdateHandPose 函数下。

将模型左右手手部骨骼关节名称一一复制到对应的配置框中，即可完成名称匹配。

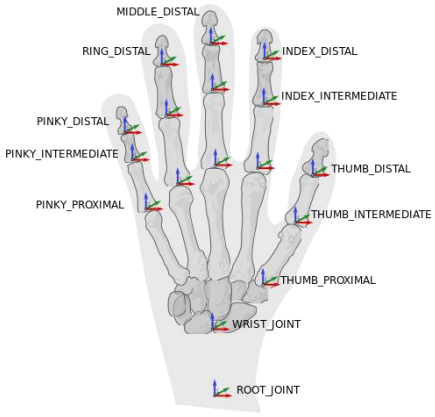




匹配时的顺序如下：

右手手腕-右手拇指近端关节-右手拇指中端关节-右手拇指远端关节-右手食指近端关节-右手食指中端关节-右手食指远端关节-右手中指近端关节-右手中指中端关节-右手中指远端关节-右手无名指近端关节-右手无名指中端关节-右手无名指远端关节-右手小指中端关节-右手小指中端关节-右手小指远端关节

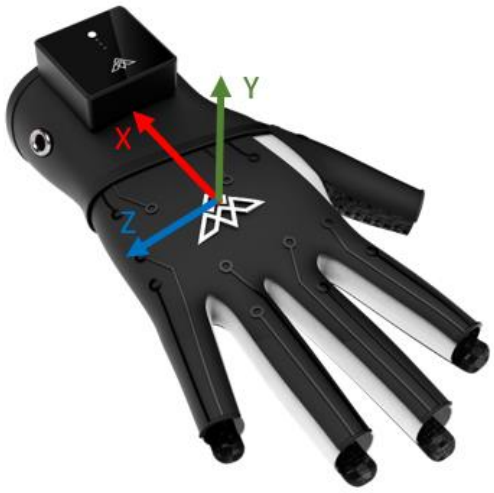
左手手腕-左手拇指近端关节-左手拇指中端关节-左手拇指远端关节-左手食指近端关节-左手食指中端关节-左手食指远端关节-左手中指近端关节-左手中指中端关节-左手中指远端关节-左手无名指近端关节-左手无名指中端关节-左手无名指远端关节-左手小指中端关节-左手小指中端关节-左手小指远端关节



3. 骨骼轴向匹配：

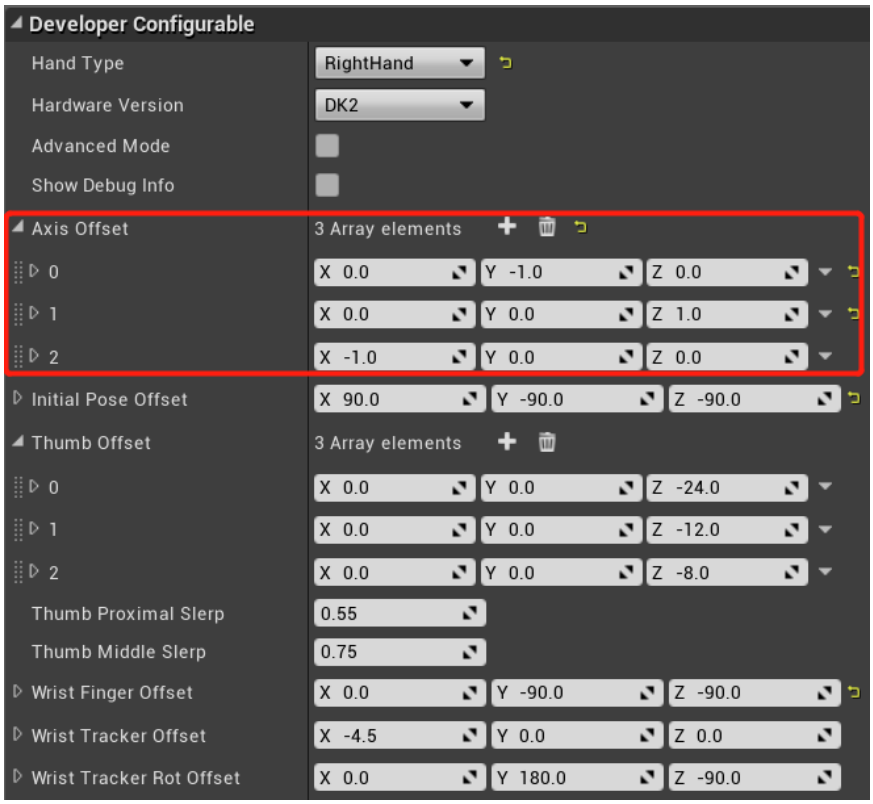
将数据手套定义的手部骨骼轴向与新模型手部骨骼轴向匹配。此处需要通过手套 UE4 插件开放的外部接口对手部骨骼轴向进行匹配。点击 LeftHandWithTracker 和 RightHandWithTracker 蓝图中插件提供的 Glove Component，右侧 detail 列表下有一栏为 Developer Configurable，此处的参数为插件对外开发的参数接口。

在手套硬件的坐标系定义如下，图中红色坐标轴为 x 轴，绿色坐标轴为 y 轴，蓝色坐标轴为 z 轴。



模型骨骼的坐标系定义在 Unity 中选中进行查看：

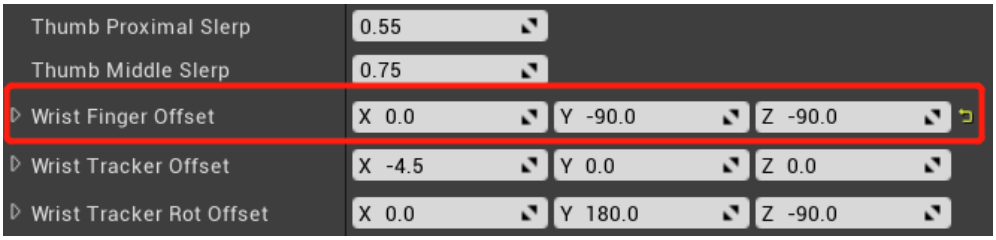
两者对比可以看出模型的 x 轴（红色轴）对应为手套硬件定义下的-y 轴，即（0，-1，0），模型的 y 轴（绿色轴）对应为手套硬件定义下的 z 轴，即（0，0，1），模型的 z 轴（蓝色轴）对应为手套硬件定义下的-x 轴，即（-1，0，0）。所以我们在脚本的配置 AxisOffset 中填入上述矩阵：



同理，左手模型也与硬件定义的坐标系做对比，得到左手的矩阵。

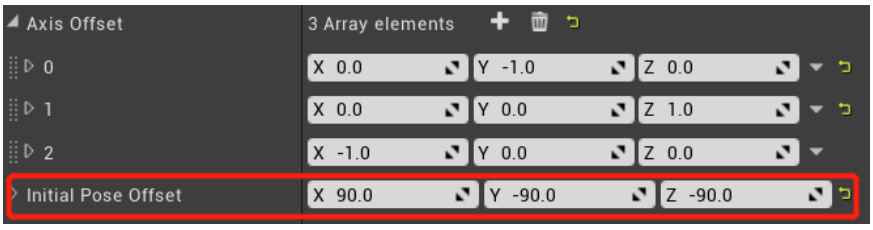
如果遇到有的模型手指的轴向和手腕不一致，可以进一步通过配置里的 WristFingerOffset 来调整。例如该模型手指和手腕坐标系如下：

即将手指坐标系绕 y 轴旋转-90，再绕 z 轴旋转-90 可以得到腕部坐标系，所以 WristFingerOffset 填写（0， -90， -90）。如果腕部轴向和手指各关节轴向一致，则此处填写（0，0，0）即可。



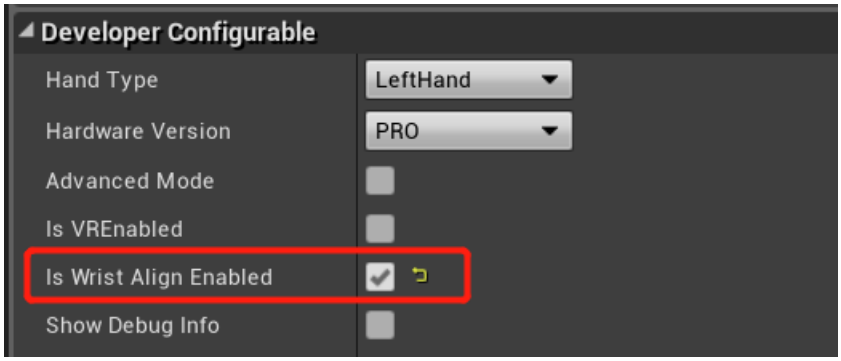
4. 腕部骨骼初始朝向匹配：

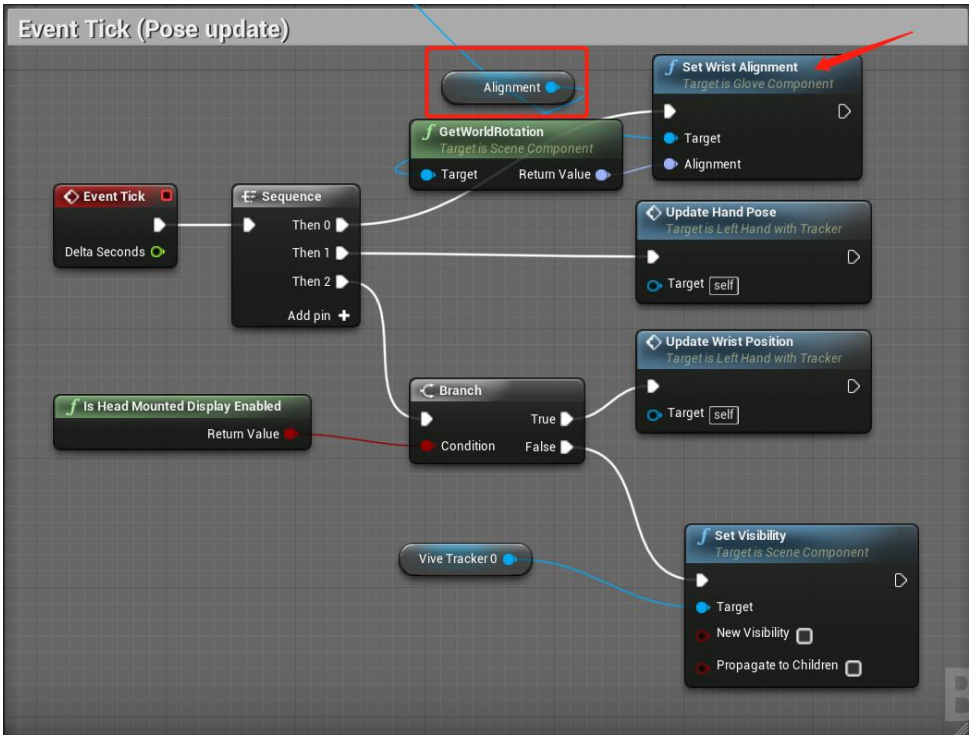
- 3D 模式下，需要指定腕部骨骼初始朝向，在如下参数框中适当修改即可，如果是全身模型，也可以将此值设置为模型上手腕关节默认状态下（如 T-Pose）的全局旋转。该值规定了点击“L”或者“R”对手腕方向进行复位时手腕的最终方向。



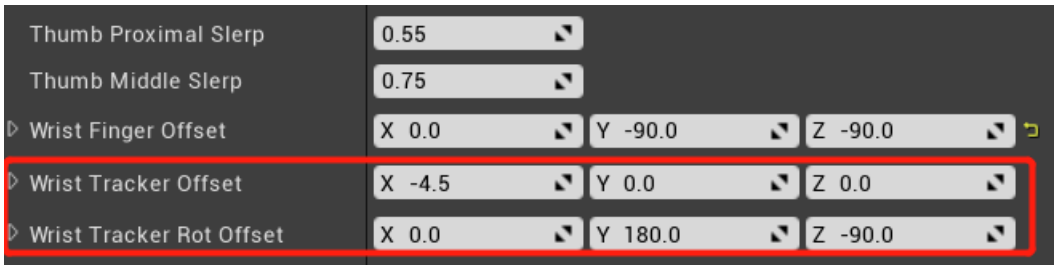
在手套与动捕设备结合的应用中，常常希望手腕的朝向始终与动捕设备手腕朝向或者手臂朝向一致，则可以勾选 IsWristAlignEnabled，然后在包含手部插件组件蓝图

的 Event Graph 中找到调用 SetWristAlignment 函数的地方，将目标改为动捕设备的手腕关节或者手臂关节，这样就可以使得动捕设备的手腕方向一直与手套的方向一致，无需进行 T-Pose 校准。





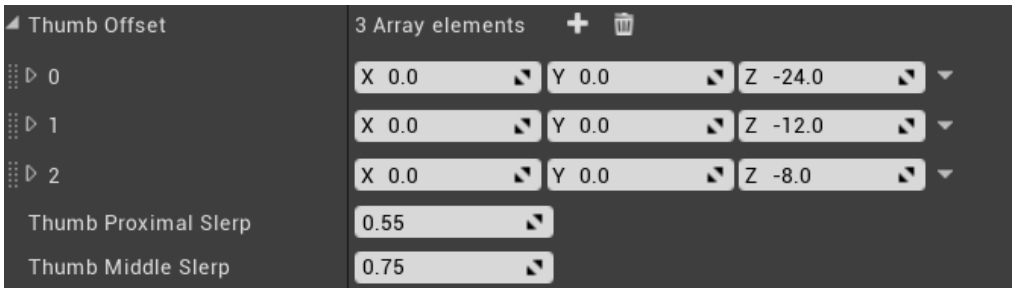
- VR 模式下，手腕初始方向默认由 Tracker 决定，故无需指定腕部骨骼初始朝向。但可以调节手部模型与 Tracker 模型之间的 offset。示例场景中使用的手部模型由于腕关节骨骼坐标系就定义在模型腕部的正中央，所以和 Tracker 模型之间的 offset 只有 x 轴上存在分量，表示了 tracker 在手腕的左侧/右侧。此外如果不使用 Tracker 进行腕部定位，使用光学 Optitrack/Vicon 等定位设备，或定位设备固定位置与 Tracker 不同，则需调整 rot offset。



至此，所有模型更换的步骤都已经完成，可以运行场景查看追踪效果了。

5. **拇指关节偏差微调：**

   由于不同手部模型大拇指处的蒙皮很难完全相同，且不同手套硬件传感器位置以及用户手指大小长度都不尽相同。所以为了达到最佳的追踪模拟效果，可以通过调整拇指关节偏差和拇指关节插值系数来调节大拇指的形态。



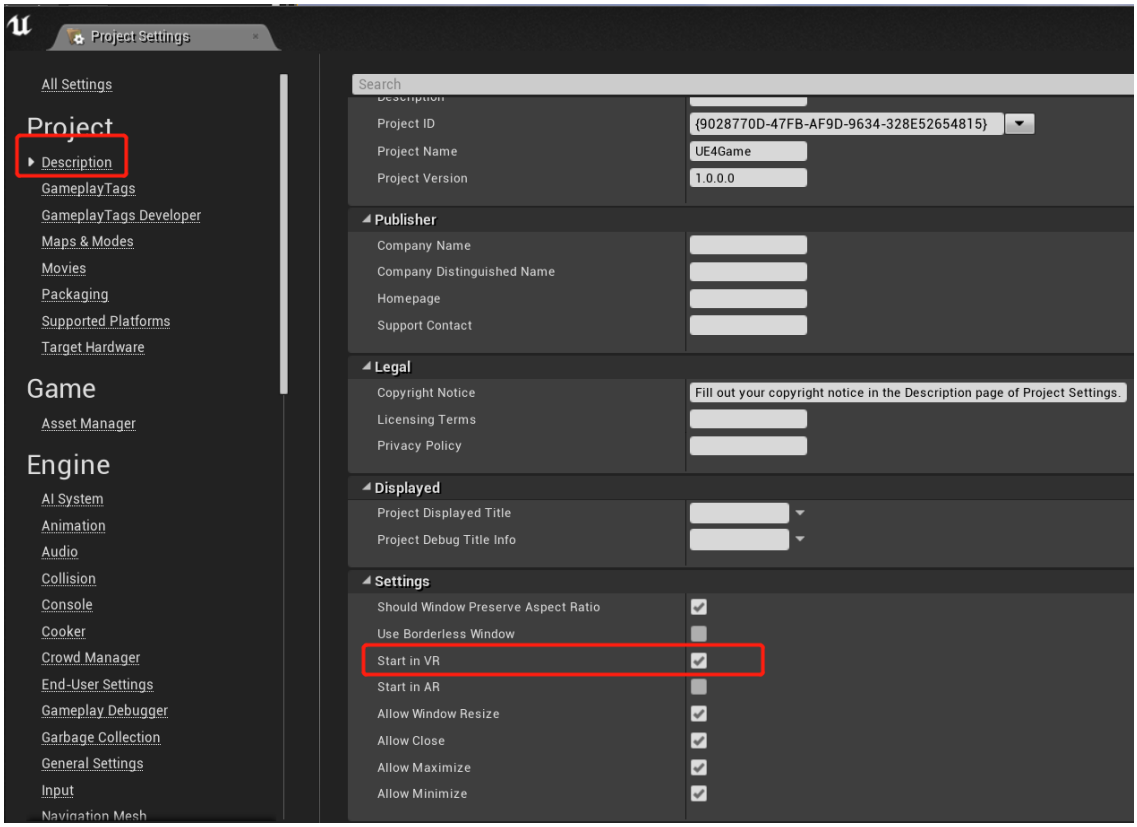
6. **四指间距和弯曲后间距调整：**

   可以通过调整以下参数来调整四指间距，以及完全弯曲后四指的间距。以下参数代表伸直状态下四指间距 8 度，完全弯曲后四指间距往里收缩 2 度。

工程打包

Project Packaging

1. 如果需要对包含手套插件的 VR 工程进行打包，需要在 Project Settings-Description-Settings 中勾选 Start in VR，同样地，如果对非 VR 工程进行打包，需要去掉 Start in VR 的勾选。



2. 目前 UE4 SDK 仅支持 Windows 平台下 Win64 版本的打包和发布，如需其他版本的 SDK，请联系技术支持 info@vrtrix.com

常见问题

FAQ

1. 为何运行 UE4 Demo 场景手部动作卡顿或经常断开？

首先确保在运行 UE4 SDK 前先安装并运行过客户端配置软件 VRTRIXGloveConfigTool，如果未安装请参考文档“北京无远弗届数据手套操作手册”先安装客户端软件对硬件驱动进行正确的安装和初始化配置。

如果已经安装和运行过客户端软件，依然有卡顿或者断开的情况，将手套 USB 接收器通过 USB Hub 接出，避免 PC 相邻 USB 口上有大流量的硬件设备例如无线网卡，移动硬盘等，同时避免手套 USB 接收器和手套之间没有过多的遮挡。在正常电磁环境下，手套的正常通信距离应在 10m 以上。

2. **为何运行 UE4 Demo 场景开始运行后手套无反应，或者模型不动？**

   首先确保两个 USB 接收器正常插在 PC 的 USB 接口上。另外保证 VRTRIXGloveConfigTool 该客户端软件在 UE4 Demo 运行时关闭。客户端配置软件只用于配置硬件参数，进行配对，以及运行 Demo，不是作为 Runtime 存在，开发时无需打开该软件。

3. **为何运行 UE4 Demo 3D 场景正常但是 VR 场景无法连接或者场景中手套不出现？**

   首先确保在运行 VR Demo 前运行过客户端配置软件 VRTRIXGloveConfigTool，对 HTC Tracker 进行过配置。从本公司直接购买的 HTC Tracker 出厂时已经提前做好配置，可以直接适配手套使用。未经过配置的 HTC Tracker 无法正确识别手套故而 VR 场景无法连接。未配置请参考文档“北京无远弗届数据手套操作手册”。

   如果已经配置过 HTC Tracker，那么请确保运行 VR 场景时 HTC Tracker 正常打开且已经配对好。

4. **为何运行 UE4 Demo VR 场景时，手部俯仰角度不对？**

   将左手朝前平举伸直，点击键盘“L”按键，可以将左手朝前对齐。同样的，将右手朝前平举伸直，点击键盘“R”按键，可以将右手朝前对齐。

   在进入场景时，插件会自动进行一次初始方向校准，所以如果双手朝前平举伸直状态下开始运行场景，则无需校准。

5. **为何运行 UE4 Demo 场景时，手部形态有时出现不正常的跳动或者有所扭曲？**

   请远离磁性物体，包括电脑系统，铁架桌椅，耳机，手机等磁场辐射源，远离 1m 以上然后戴着手套划八字，八字校准 1 分钟后或看手部形态稳定下来后，点击按键“C”对当前磁环境参数进行储存。该操作只需要进行一次，以后在相同环境下无需再进行校准。

6. **为何替换手部模型后，手部形态不正常？**

   请确认手部骨骼名字是否有对应错误的地方，以及手部骨骼轴向是否有计算错误。如果都没有，请检查模型，是否手部每个骨骼的轴向一致。请认真阅读本文档手部模型替换部分关于模型替换的说明再试。