## 1. 概述

随着越来越多的VR设备推出，控制器类型逐渐趋向于碎片化。每当有新的控制器发布，都会给开发者带来一些额外的工作量——游戏项目需要修改交互代码以适配新的设备。从开发层面上来看，不同的控制器具有不同的键值映射，所以，当现有 VR 应用程序移植到另外一个VR平台的时候，需要针对目标平台进行交互适配。鉴于此， Valve为Unity开发者推出了 SteamVR Unity Plugin 2.0，能够使开发者在编程中专注于用户的动作，而不是具体的控制器按键。

## 2. 重要更新：Input System

SteamVR 2.0 的重要更新是加入了 Input System。推出Input System的目的，是为了更加符合OpenXR标准，

Input System与之前处理用户输入有显著的不同，使用SteamVR Input System，开发人员可以在应用程序之外定义默认的动作并与按键进行绑定，而不需要将输入视为某一特定设备的特定按键。这样新的设备可以快速适配应用程序，无需更改代码。比如，当开发者检测玩家是否抓取某个物体的时候，不是检测Vive控制器的Trigger键或Oculus Touch控制器的Grip键是否被按下，而是检测预定义的"Grab"动作是否为True即可。作为开发者，可以在SteamVR中为Grab动作设置默认按键和阈值，当程序运行时，也可修改这些数值以满足玩家的个人偏好。基于这种机制，不光能够解决控制器碎片化的问题，也可以快速适配未来发布的设备。

## 3. 动作（Actions）

Input System 的核心概念是动作（Action），基于动作的输入系统对于游戏引擎来说更有意义， Unreal一直在沿用这种方案，而Unity目前在开发中的输入系统也将遵循这一原则。开发者需要放弃之前关于“按下某个按键发生什么事情”的思想，取而代之的是使用“做出某个动作发生什么事情”的思想。

SteamVR 2.0将动作抽象为以下6种类型，简介如下：

Boolean类型的动作代表只有两种状态的动作——True或False，比如抓取（Grab）动作，只有抓取或未抓取两种状态，不存在中间状态。在Unity中对应类为SteamVR\_Action\_Boolean。

Single类型的动作能够返回0~1之间的数值，比如 Trigger 键按下到松开的过程。在Unity中对应类为SteamVR\_Action\_Single。

Vector2类型动作能够返回二维数，比如Touchpad上的触摸或手柄摇杆。使用这样的数值能够控制物体在四个方向的运动，典型的应用时使用Touchpad控制无人机或小车的运动。在Unity中对应类为SteamVR\_Action\_Vector2。

Vector3类型的动作能够返回三维数值，在Unity中对应类为SteamVR\_Action\_Vector3。

Pose类型的动作表示三维空间中的位置和旋转，一般用于跟踪VR控制器。在Unity中对应类为SteamVR\_Action\_Pose。

Skeleton类型的动作能够获取用户在持握手柄控制器时的手指关节数据，通过返回数据，结合手部渲染模型，能够更加真实的呈现手部在虚拟世界的姿态，虽然不及像LeapMotion等设备获取手指输入那样精确，但是足以获得良好的沉浸感。在Unity中对应类为SteamVR\_Action\_Skeleton。

## 3.2 振动输出

以上是介绍的都是输入动作，另外，目前还有一种输出动作——振动，用于触发VR控制器上的触觉反馈，调用方法如下代码所示：

SteamVR\_Actions.default\_Haptic.Execute（float secondsFromNow, float durationSeconds, float frequency, float amplitude, SteamVR\_Input\_Sources inputSource）；

————————————————

## 4. 动作集（Action Sets）

动作通过动作集进行逻辑上的分组，以方便进行组织和管理。在Unity中对应的类为SteamVR\_ActionSet。在不同的场景或应用程序之间可以切换使用不同的动作集，比如，应用程序中有一个场景是在地球上拾取并投掷物体，而另一个场景则是在太空中飞行，那么这两个场景可以使用不同的动作集。同时，当针对新设备进行交互适配时，开发者只需对动作进行配置，而不必修改项目代码。比如，使用 Vive 控制器时，定义了一个Fire动作，当需要支持 Rift Touch 时，只需通过配置Touch控制器上符合 Fire 动作的键值即可。

SteamVR插件默认包含了三套动作集default、platformer、buggy，开发者也可以在SteamVR Input窗口中自行添加或删除动作集。

使用组件SteamVR\_ActivateActionSetOnLoad可以在场景中自动激活和停用指定的动作集。对应激活和停用的方法是在Start（）和OnDestroy（）中实现。

## 5. SteamVR Input 窗口

在Actions栏的右下角，可以点击加减号按钮添加或删除动作。每个动作具有名称（Name）、类型（Type）、本地化字符串（localized String）等字段。其中，类型对应上节介绍的动作类型；本地化字符串是面向用户进行绑定的动作名称。

当点击Save and Generate按钮后，插件将为动作以及动作集生成可编程访问的对象类，将它们放置在项目的SteamVR\_input目录下，

## 6. 测试动作

选择 Window > SteamVR Input Live View 命令，即可打开一个测试输入窗口。运行程序，此时该窗口将实时展示所有动作集合的状态。

7．代码

直接通过SteamVR\_Actions.xxx静态访问

activeDevice可以获取到发送动作的Source