广州新维感 3i VR 手柄 SDK API 使用说明

Document Number:	URUS-003
Document Status:	Approved
Document Issue Number:	3.0.0
Issue Date:	2017-06-16
Security Status:	Xinweigan Confidential
Author:	Michael Yang / Lucas Wu

Intended left spaces

Table of Contents

IDENTIFICATION OF DOCUMENT **4**

STORAGE LOCATION 4
LIST OF CONTRIBUTORS 4
APPROVER(S) 4
PUBLICATION HISTORY 5

1 概述 **7**

2 UNITY3D 入门 **8**

2.1 先决条件 8
2.2 导入 SDK 8
2.3 启用 I3VR 控制器和 UI 的支持 8
2.4 示例场景 8
2.5 发布设置 9
2.6 3IVR 控制器支持 9
2.6.1 Arm Model *9*
2.6.2 Input System *10*
2.6.3 Laser and reticle visualization *10*
2.6.4 Controller Visualization *10*

3 API 参考 **11**

3.1 I3VRCONTROLLER 11
3.2 I3VRARMMODEL 14
3.3 I3VRARMMODELOFFSETS 15
3.4 I3VRLASERPOINTER 16
3.5 I3VRSETTINGS 16

4 APPENDICES **17**

IDENTIFICATION OF DOCUMENT

Storage Location

https://github.com/3ivr/3ivr-unity-sdk

List of Contributors

Identify all of the key contributors to this document.

Name	Role
Lucas Wu	Unity Software Designer
Michael Yang	Platform SDA

Approver(s)

Identify the person or people who must approve this document.

Name	Role	Area of Responsibility
Michael Yang	System Design Authority	Converged Platform

Publication History

Identify the changes that are made to the document. Please update the Issue Number appropriately. The Issue Number must conform to the Document and Record Control Process. A generic Publication History table with example text is provided below.

Issue	Change Summary	Author(s)	Date
Preliminary	Initial document creation	Michael Yang	2016-10-08
Draft 0.50	Add API description	Lucas Wu	2016-10-15
1.0.0	Changes from formal review	Michael Yang	2016-12-01
2.0.0	Add ArmModel	Lucas Wu	2017-03-01
	URUS-002 API LEVEL approved	Michael Yang	
3.0.0	URUS-003 API LEVEL approved	Michael Yang	2017-06-16

Intended left spaces

1 概述

新维感 3iVR 手柄 SDK 支持 Android 平台，包括 Daydream 和 Cardboard 平台。通过手柄可以获取手柄位置，选取，简化手柄位置、姿态和状态的控制，以及头显的控制。3iVR Unity SDK 包括手柄相关的支持、应用和示例程序。通过该使用说明可以方便开发人员创建基于 3iVR 手柄的应用程序。简化他们的 VR 开发，比如：

1) 头显回中
2) 手柄追踪
3) 手柄姿态
4) 用户输入事件处理

我们将 3iVR SDK 开源，方便和鼓励开发人员开发基于该手柄的应用。

2 UNITY3D 入门

Unity3D 使得开发人员开发 VR 应用变得简单。该指南将指导设置 Unity 开发 3iVR 控制器应用，以及创建一个 demo 场景。

Unity 对 VR 开发的支持可以方便我们：

1) 从新开始一个 Unity VR 项目
2) 移植现有 Unity 3D 应用到 VR
3) 方便在 VR 模式和非 VR 模式之间的切换

3iVR SDK 方便我们：

1) 用户头部追踪
2) 检测用户与系统之间的交互（通过触发器或者控制器）
3) 自动修正手柄位置误差

2.1 先决条件

需要 Unity 5.4.5 及以上版本。在安装 Unity 时确认已经选取"Android Build Support"组件（默认是已经选取）。如果这是首次在 Unity 下开发 Android 应用，请按照官方指南设置 Unity 的 Android SDK（https://docs.unity3d.com/Manual/android-sdksetup.html）。

下载 3iVR SDK for Unity（https://github.com/3ivr/3ivr-unity-sdk）。包括 SDK 和 demo 程序。

当然你还需要 Android 手机和 3i 手柄。

2.2 导入 SDK

1) 打开 Unity 5.4.5 或以上版本,创建或打开一个工程项目.
2) 找到选项 Assets > Import Package > Custom Package。打开并 选中 3ivrSdkForUnity 文件，检查导入资源包内所有资源是否选中，然后单击 Import。如果出现提示，请接受升级。

2.3 启用 I3vr 控制器和 UI 的支持

1) 创建一个空物体，并命名为 Player。
2) 设置 Player Position 为 (0,1.6,0)。
3) 将 Main Camera 设置为 Player 子物体，并位于(0,0,0)。
4) 添加 Assets >I3VRSDK>Prefabs 下 I3vrControllerPointer 到 Player 子物体.并位于(0,0,0)。
5) 添加 Assets >I3VRSDK>Prefabs 下 I3vrCanvas 到场景内。
6) 添加 Assets >I3VRSDK>Prefabs 下 I3vrEventSystem 到场景内。
7) 添加 Assets >I3VRSDK>Prefabs 下 I3vrControllerMain 到场景内。
8) 保存场景。

2.4 示例场景

1) Assets>I3VRSDK>Scene>Main.unity
   a) 按钮状态显示，触摸点触点位置，触摸板手势触发。

b) 按钮检测到 ButtonDown 切换绿色，检测到 ButtonUp 切换红色。
c) 触摸点检测到 TouchDown，显示绿点，并显示对应的触点位置。检测到 TouchUp 时隐藏。
d) 触摸手势触发会显示对应手势箭头(左划，右划，上划，下划，在显示 0.1S 后隐藏)。
e) 手柄实时姿态展示。
f) Rotation，Gyro，Accele，Touch Pos 对应 API 读数显示。

2) Assets>I3VRSDK>Scene>TestButton.unity
a) UGUI-Button 触发展示。
b) Arm Model 手臂模型展示。
c) 激光光标与按钮提醒展示。

2.5 发布设置

1) 找到选项 File>Build Settings，将场景添加到场景列表，并确认为首个启动的场景，将发布平台选择为 Android 平台，并点击 Switch PImport，确认 Unity 图标在对应 Android 平台。
2) 选中 Player settings，找到 Other Settings 并执行以下操作。
3) Bundle Identifier 输入包名(例如 com.i3vr.UnityDemo)。
4) Minimum API Level 设置为 Android 5.0“Lollipop”(API LEVEL 21)或以上。

2.6 3iVR 控制器支持

1) Arm Model：使得 VR 中的控制器模型接近于 I3vr 控制器物理位置的数学模型。
2) Input System：控制器模型射线交互模块,包含输入模块和接收模块,使得更好的通过手柄与 UI 或其他对象进行交互。
3) Controller visualization：I3vr 控制器的 3D 模型,按钮提醒,触摸板触点可视化。
4) Laser and reticle visualization：显示激光笔和光标,试用户轻松的和 VR 环境交互。

2.6.1 Arm Model

提供了一个手臂模型，一个使用物理控制器的方向和位置的数学模型，并且预测肩部，肘部，手腕和指针的位置，以确定在场景中放置控制器模型的位置。手臂模型通过 I3vrArmModel 脚本进行控制，该脚本附加到提供的预制 I3vrControllerMain。

❖ 注意：场景中必须只有一个 I3vrArmModel 实例。

该 I3vrArmModel 脚本允许您调整以下属性：

1) 添加肘部高度：增加了一个偏移量来预测肘部的高度。
2) 添加肘部深度：增加一个偏移量预测肘部深度。
3) 指针倾斜角度：相对于控制器的指针向下倾斜的角度。对于和眼睛水平的物体进行交互时，15°的默认值是比较舒适的。如果用户需要与显着高于或低于对象的对象进行交互，则可能需要调整此属性。
4) 脸部淡出距离：控制器与脸部之间的距离，此后 α 值降低。
5) 追随目光：确定肩膀是否应该跟随用户的目光。
6) 使用加速度计：关闭加速器使控制器感觉更准确和可预测。打开它减少精度，但可以使用户感觉更像是持有一个真正的对象。

❖ 注意：默认设置被调整，假设用户将像激光笔一样握住控制器。如果您的应用程序需要使用控制

器进行诸如摆动蝙蝠或捕捉球的操作，则需要适当地调整设置。

2.6.2 Input System

要想使用此系统，请添加预制体 I3vrEventSystem 到场景中，你可以通过 I3vrPointerInputModule 脚本来配置输入系统。

2.6.3 Laser and reticle visualization

预制体 I3vrControllerPointer 下 Laser 挂载 I3vrLaserPointer 脚本控制着激光可视化和显示光标。该脚本具有以下设置：

1) 激光颜色：确定激光的颜色。
2) 最大激光距离：当您目前没有指向物体时，激光显示的最大距离（以米为单位）。
3) 最大刻线距离：当您目前未指向对象时，这是激光光标显示的最大距离（以米为单位）。

2.6.4 Controller Visualization

预制体 I3vrControllerPointer 下 I3vrController 对象下挂载的 DisplayTooltips 脚本控制触点可视化和按钮提醒 UI,有如下设置：

1) 是否显示提醒 UI。
2) X 轴最小显示角度：大于该值小于最大角度时显示 UI，为顺时针旋转值。
3) X 轴最大显示角度：小于该值大于最小角度时显示 UI 该值为顺时针旋转值。

3 API 参考

3.1 I3vrController

Summary:

3iVR controller API 的主入口。

为了使用 API，增加这个对象到你场景中的 GameObject 中，或者使用 I3vrControllerMain 预制体。一个场景中只允许一个对象允许有该对象。

这是一个单例对象。

为了存取控制器的状态，简单读取该类的静态属性。比如，为了知道控制器当前方向，使用 I3vrController.Orientation。

Inheritance:

```
Inherits from: MonoBehaviour
```

Events:

OnControllerUpdate	OnControllerUpdateEvent
HeadsetRecenter	OnHeadsetRecenter

Properties:

ArmModel	static I3vrArmModel 返回手臂与控制器关联模型实例。
ConnectionState	static I3vrConnectionState 返回当前控制器连接状态。
ApiStatus	static I3vrControllerApiStatus 返回当前控制器 API 状态。
Orientation	static Quaternion 表示方向的为 Unity 坐标系，X 指向右侧，Y 指向上，Z 指向前方。因此，要使场景中的对象与控制器具有相同的方向，只需将该四元数赋值到 GameObject 的 transform.rotation。
Gyro	static Vector3 返回控制器的陀螺仪读数。

	陀螺仪表示其每个局部坐标的角度。控制器的轴是：X 指向右侧，Y 指向上，Z 指向前方。角速度单位为弧度/秒。
Accel	static Vector3 返回控制器的加速度计读数。 加速度计表示加速度和重力在每个控制器局部坐标的方向上的影响。控制器的局部坐标为：X 指向右侧，Y 指向上，Z 指向前方。加速度以米/秒为单位进行测量。
IsTouching	static bool 如果为 Ture，则用户正在触摸。
TouchDown	static bool 如果为 Ture，则用户刚触摸。 这是一个事件标志：事件发生后只有一帧是 true。
TouchUp	static bool 如果为 Ture，则用户刚停止触摸。 这是一个事件标志：事件发生后只有一帧是 true。
TouchPos	static Vector2 返回触摸板触点读数。原点位于矩形左上角(0，0)，右下角为(1，1)。
Recentered	static bool 如果是 true，则用户刚刚完成了手柄回中。
TriggerButton	static bool 如果是 true，则用户按下了 Trigger 键。
TriggerButtonDown	static bool 如果是 true，则用户刚按下了 Trigger 键。 这是一个事件标志：事件发生后只有一帧是 true。
TriggerButtonUp	static bool 如果是 true，则用户弹起了 Trigger 键。 这是一个事件标志：事件发生后只有一帧是 true。
AppButton	static bool

	如果是 true，则用户按下了 App 键。
AppButtonDown	static bool 如果是 true，则用户刚按下了 App 键。 这是一个事件标志：事件发生后只有一帧是 true。
AppButtonUp	static bool 如果是 true，则用户弹起了 App 键。 这是一个事件标志：事件发生后只有一帧是 true。
HomeButton	static bool 如果是 true，则用户按下了 Home 键。
HomeButtonDown	static bool 如果是 true，则用户刚按下了 Home 键。 这是一个事件标志：事件发生后只有一帧是 true。
HomeButtonUp	static bool 如果是 true，则用户弹起了 Home 键。 这是一个事件标志：事件发生后只有一帧是 true。
SwitchButton	static bool 如果是 true，则用户按下了 Switch 键。
SwitchButtonDown	static bool 如果是 true，则用户刚按下了 Switch 键。 这是一个事件标志：事件发生后只有一帧是 true。
SwitchButtonUp	static bool 如果是 true，则用户弹起了 Switch 键。 这是一个事件标志：事件发生后只有一帧是 true。
TouchGestureLeft	static bool 如果是 true，则用户触发了左划手势。
TouchGestureRight	static bool 如果是 true，则用户触发了右划手势。
TouchGestureUp	static bool

	如果是 true，则用户触发了上划手势。
TouchGestureDown	static bool 如果是 true，则用户触发了下划手势。
ErrorDetails	static string 如果 State == I3vrConnectionState.Error，则包含有关错误的详细信息。

3.2 I3vrArmModel

Summary:

I3vrArmModel 是控制器和场景之间的标准接口，其主要负责：

- 检测手柄的方向和位置
- 预测肩膀、肘部、手腕和指针的位置

在场景中应该只有一个实例，而且需依附于 I3vrController 对象。

Inheritance:

Inherits from: MonoBehaviour

Public types:

GazeBehavior	Enmu 设置追随模式

Properties:

Instance	static I3vrArmModel 使用 I3vrController 获取此类的单例。
pointerPosition	Vector3 表示指针位置的向量。
pointerRotation	Quaternion 表示指针旋转的四元数。
wristPosition	Vector3 表示手腕位置的向量。
wristRotation	Quaternion 表示手腕旋转的四元数
elbowPosition	Vector3

	表示肘部位置的向量。
elbowRotation	Quaternion 表示肘部旋转的四元数
shoulderPosition	Vector3 表示肩膀位置的向量。
shoulderRotation	Quaternion 表示肩膀旋转的四元数。
alphaValue	Float 控制器射线的 alpha 值。

Public attributes:

addedElbowHeight = 0.0f	Float 肘部高度（m）。
addedElbowDepth = 0.0f	Float 肘部深度（m）。
pointerTiltAngle = 15.0f	Float 激光指示器相对于控制器的向下倾斜的角度。
fadeDistanceFromFace = 0.32f	Float 控制器距离摄像机消失距离(m)。
followGaze = GazeBehavior.DuringMotion	GazeBehavior 确定肩膀是否应该追随目光。
useAccelerometer = false	Bool 是否开启加速度计。

3.3 I3vrArmModelOffsets

Summary

该脚本根据手臂模型的关节来确定位置和旋转变换。

Inheritance

Inherits from: MonoBehaviour

Public types

Joint	Enum

	关节设置

Public attributes

joint	Joint 确定哪个关节设置位置和旋转。

3.4 I3vrLaserPointer

Summary

激光射线点附属于 Controller 对象。激光射线可以帮助用户定位光标。

Inheritance

Inherits from: MonoBehaviour

Public attributes

laserColor = new Color(1.0f, 1.0f, 1.0f, 0.25f)	Color 激光指示器的颜色包括 alpha 透明度。
maxLaserDistance = 0.75f	Float 指针的最大距离（m）。
maxReticleDistance = 2.5f	Float 光标的最大距离（m）。
reticle	GameObject 光标对象。

3.5 I3vrSettings

Public types

UserPrefsHandedness	enum 设置惯用手。

Public Properties

VrHeadTransform	Transform 设置 VR 头显 Transform。

4 APPENDICES