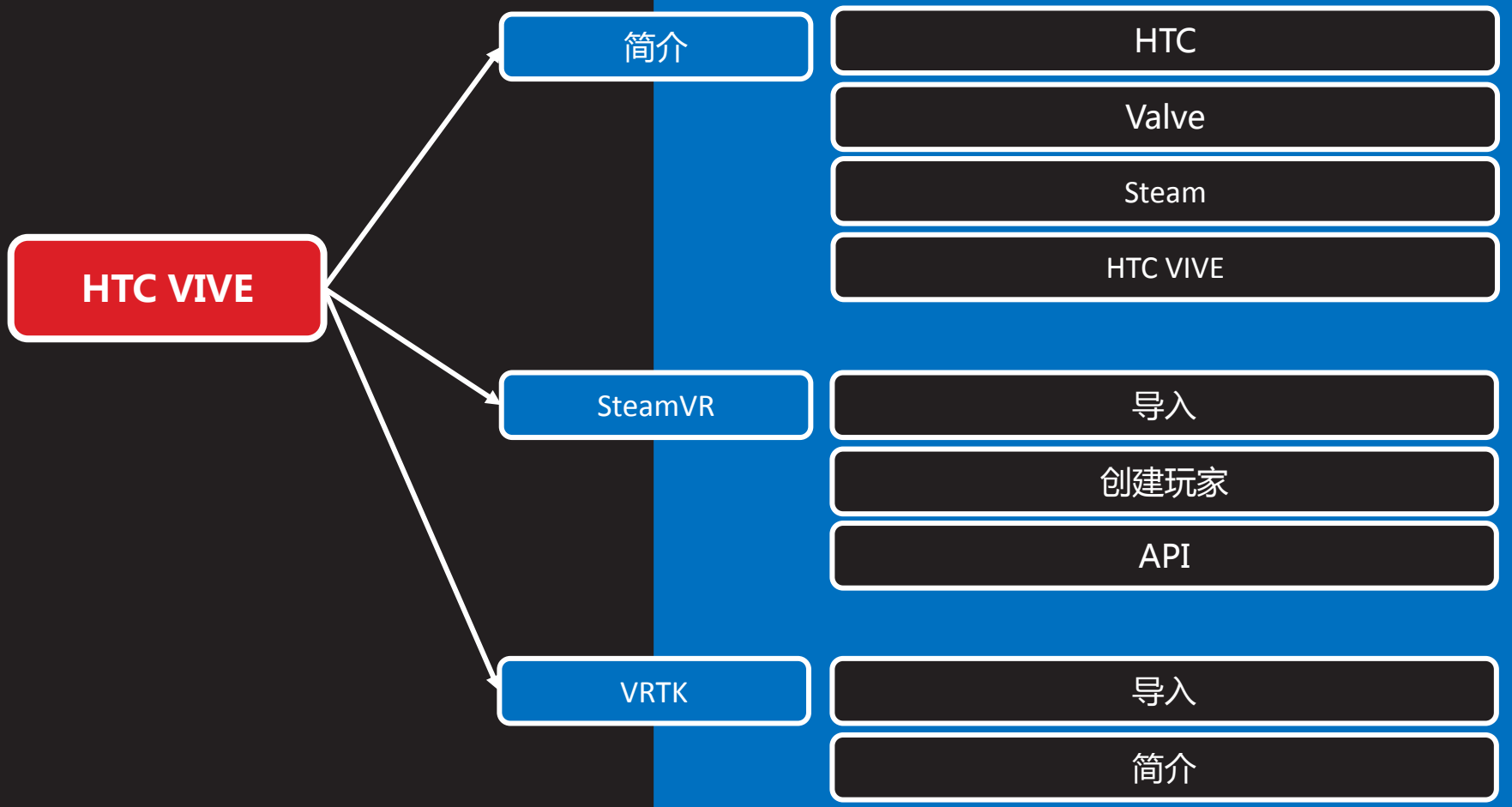
Technology Tarena 达内科技

HTC VIVE 交互指南

INTERACTIVE

HTC VIVE 开发　VRTK 开发

HTC VIVE


HTC VIVE
简介
HTC
Valve
Steam
HTC VIVE
SteamVR
导入
创建玩家
API
VRTK
导入
简介

Technology Tarena 达内科技

简介

HTC

- 宏达电子，是一家位于台湾的手机与平板电脑制造商。是全球最大的 Windows Mobile 智能手机生产厂商，全球最大的智能手机代工和生产厂商。

知识讲解

Valve

- 维尔福软件公司（Valve Software）是一家位于华盛顿州西雅图市专门开发电子游戏的公司。
- 发行的游戏有半条命系列、反恐精英系列、求生之路系列、传送门系列、军团要塞……

知识讲解

Steam

- 是 Valve 公司聘请的 BT下载 发明者亲自开发设计的一款全球最大的综合性游戏平台。玩家可以在该平台购买、下载、讨论、上传、分享游戏。

知识讲解

HTC VIVE

- 由 HTC 与 Valve 联合开发的一款VR头显，于2015年3月发布。
- 通过以下三个部分给使用者提供沉浸式体验
  --- 头戴式显示器一个
  --- 无线手持控制器两个
  --- 定位器两个

知识讲解

HTC VIVE

- 头戴式显示器（Head Mount Display）

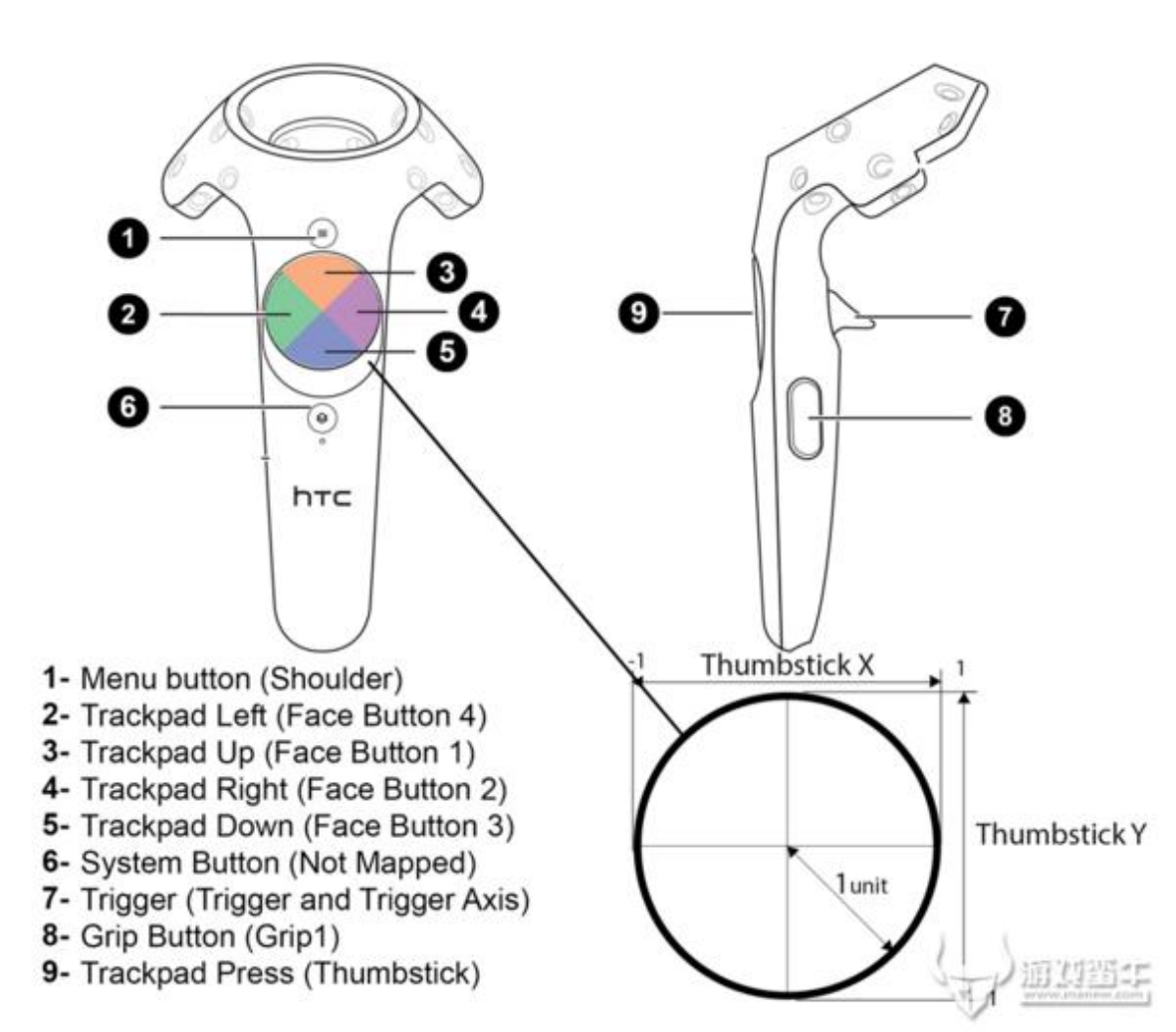
单眼有效分辨率为 1200 x 1080 ，双眼合并分辨率为 2160 x 1200，大大降低了画面的颗粒感。

画面刷新率为 90Hz，延迟 22ms，实际体验几乎零延迟。

知识讲解

HTC VIVE

- 无线手持控制器（Controller）



知识讲解

HTC VIVE

- 定位器

采用 Valve 专利 Lighthouse ，不需要借助摄像头，而是靠激光和光敏传感器来确定移动的物体位置，允许用户在一定范围内运动。

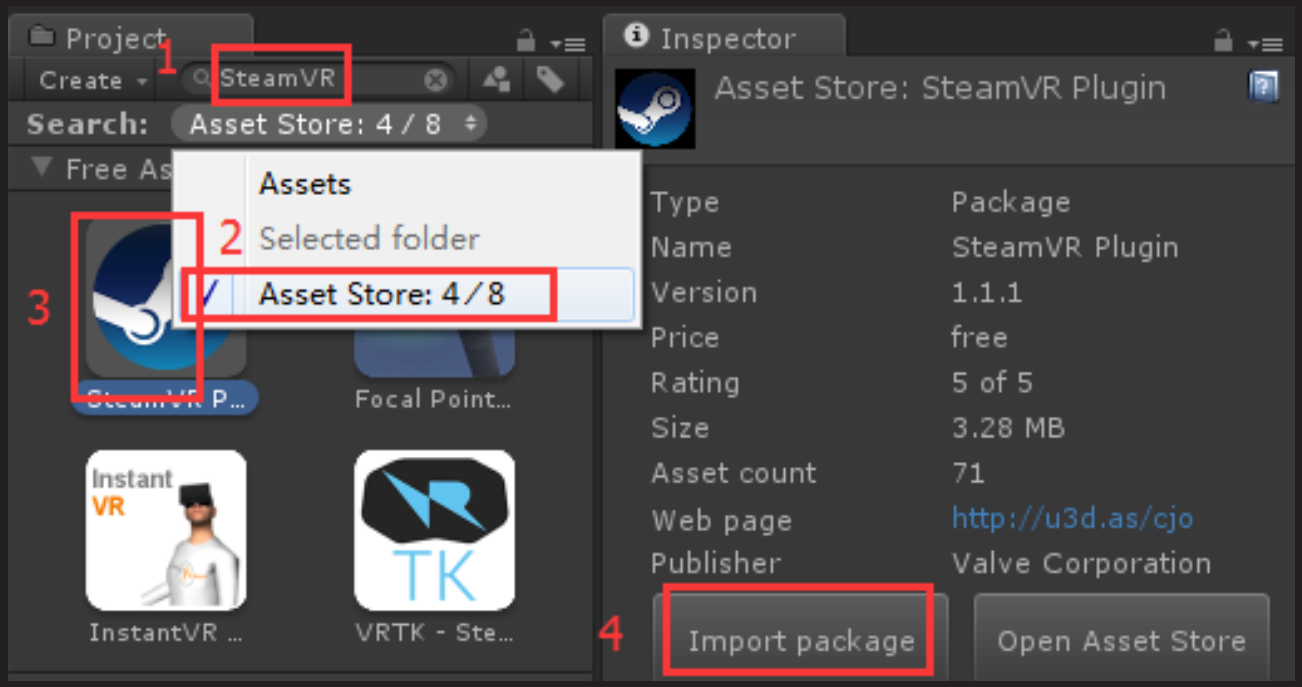
知识讲解

Technology Tarena 达内科技

SteamVR

导入

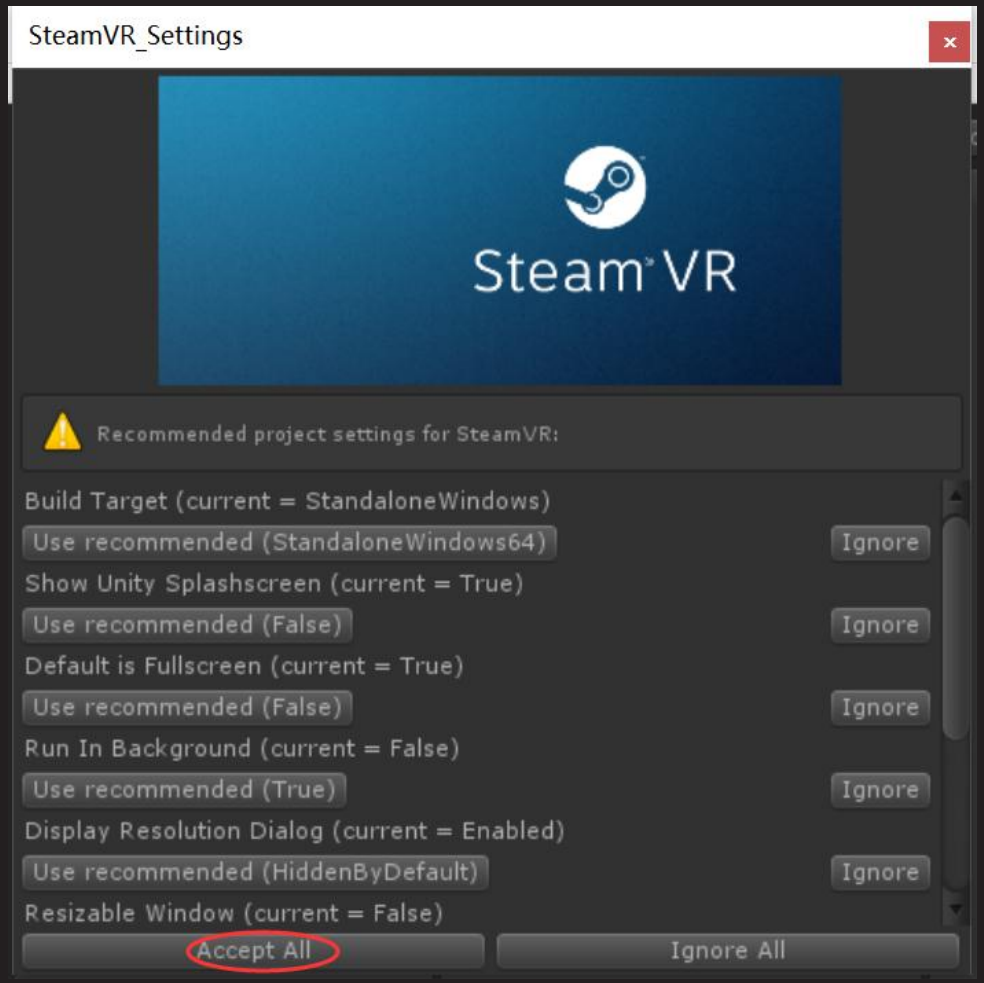
- 1. 在 Project 中输入“SteamVR”。
- 2. 选择从 Asset Store 中搜索。
- 3. 选择 SteamVR Plugin。
- 4. 导入包 Import package。



知识讲解

导入(续1)

- 推荐的项目设置，建议点击接受全部 Accept All 按钮。

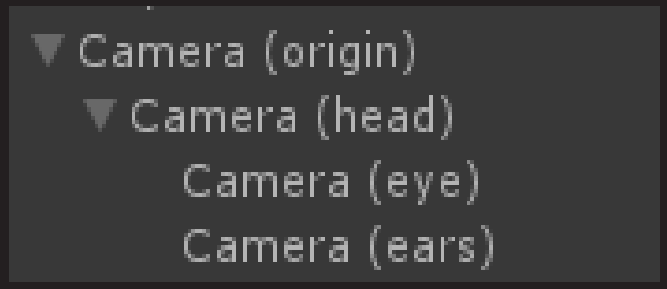


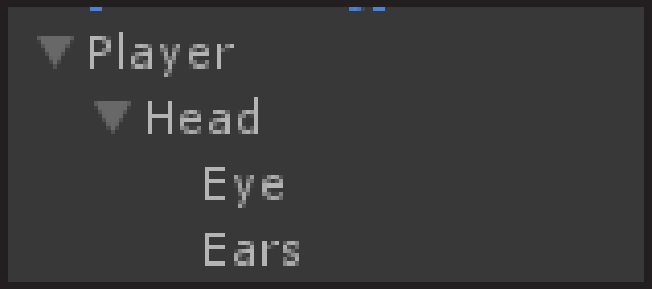
知识讲解

创建玩家

- 创建摄像机，附加 SteamVR_Camera 脚本，点击Expand按钮，将Unity Camera 转换为 VR Camera。

  修改物体名称如下：





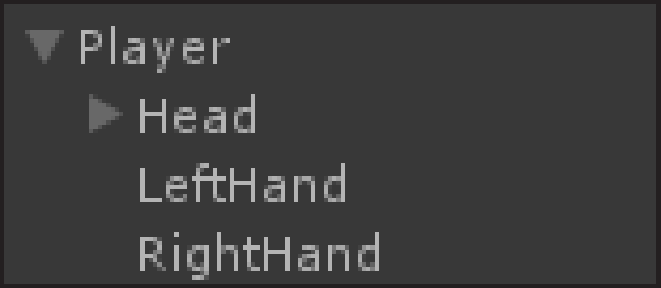
- 为 Player 添加两个子物体 LeftHandl、RightHand ，并附加脚本 Steam VR_Tracked Object，index 设置为None，作为玩家的手柄。

知识讲解

创建玩家(续1)

- 为 Player 添加脚本 Steam VR_Controller Manager ，并将 Left、Right 属性指定为 LeftHand 与 RightHand，用于管理手柄控制器。
- 为 LeftHandl 与 RightHand 分别添加子物体 Model，并附加Steam VR_Render Model 脚本，用于渲染手柄模型。

知识讲解



添加手柄模型(枪)

- 1. 使用 SteamVR/Prefabs/[CameraRig] 作为玩家。
- 2. 将模型(枪)添加至 Controller(left/right)中。
- 3. 运行场景后，调整模型(枪)的位置与角度，使枪的扳机与手柄的Trigger 键重合。
- 4. 点击模型(枪)的 Copy Component 按钮记录变换组件数值，停止运行后再 Paste Component Values。
- 5. 禁用 Controller/Model 中的 SteamVR_RenderModel 脚本。
- 6. 将 [CameraRig] 更名为 Player 后制为预制件。

知识讲解

API

- SteamVR_Controller 手柄控制器类：提供了对手柄的操作。
  -- public static Device Input(int deviceIndex)
  根据设备索引号，获取设备对象。
  -- 内部类
  Device 设备类：提供了手柄具体功能，如按下、触摸。
  ButtonMask 按键名称类：为按键ID提供名称。

知识讲解

API(续1)

- Device 设备类
  - -- GetPress：按住指定按键时返回true
  - -- GetPressDown：按下指定按键的第一帧返回true
  - -- GetPressUp：释放指定按键的第一帧返回true
  - -- GetTouch：触碰指定按键时返回true
  - -- GetTouchDown：触碰指定按键第一帧返回true
  - -- GetTouchUp：触碰指定按键最后一针返回true
  - -- GetAxis：获取 Touchpad 坐标或 Trigger 行程值（0-1）
  - -- TriggerHapticPulse：震动，参数为力度 1--4000。
  - -- GetHairTrigger：按下扳机键且行程值达到0.1时为true。

API(续2)

Tarena Technology 达内科技

代码实践

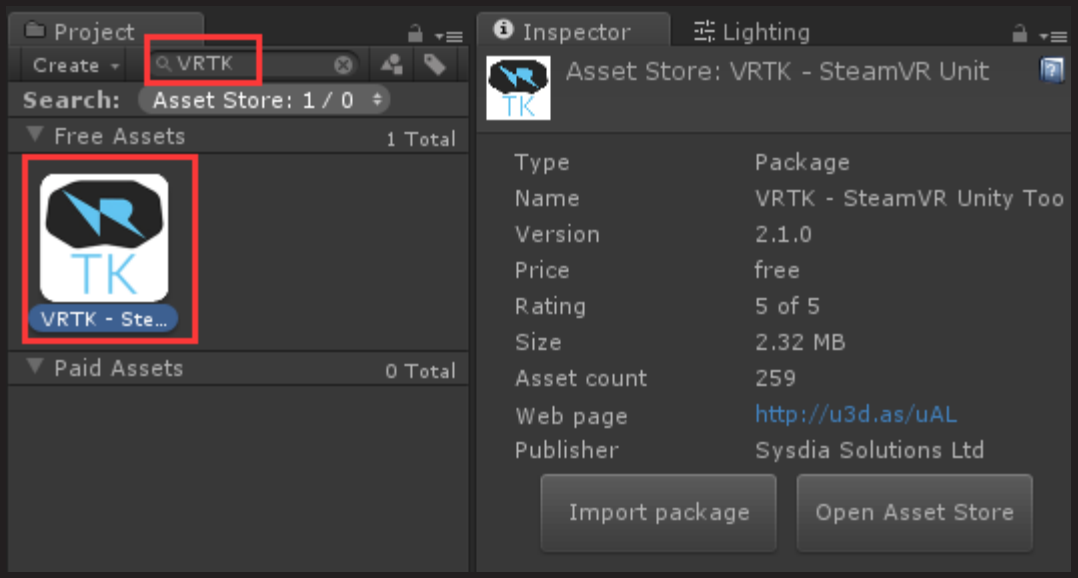
```
private SteamVR_TrackedObject trackdeObjec;
private SteamVR_Controller.Device device;
private void Awake()
{
    //1. 获取手柄上的追踪对象引用
    trackdeObjec = GetComponent<SteamVR_TrackedObject>();
}
private void Update()
{
    //2. 通过当前手柄索引 获取 手柄设备Device对象引用
    if (device == null)
        device = SteamVR_Controller.Input((int)trackdeObjec.index);

    //3. 调用手柄设备对象中的各个方法，获取用户的操作。
    //Trigger 按下扳机键时
    if (device.GetPressDown(SteamVR_Controller.ButtonMask.Trigger))
    {
        Debug.Log("GetPressDown -- Trigger 按下扳机键");
    }
}
```

Technology Tarena 达内科技

VRTK

导入

- 1. 在 Project 中输入“VRTK”。
- 2. 选择从 Asset Store 中搜索。
- 3. 选择 VRTK – SteamVR Unity Toolkit。
- 4. 导入包 Import package。



知识讲解

简介

- SteamVR Unity Toolkit 工具包

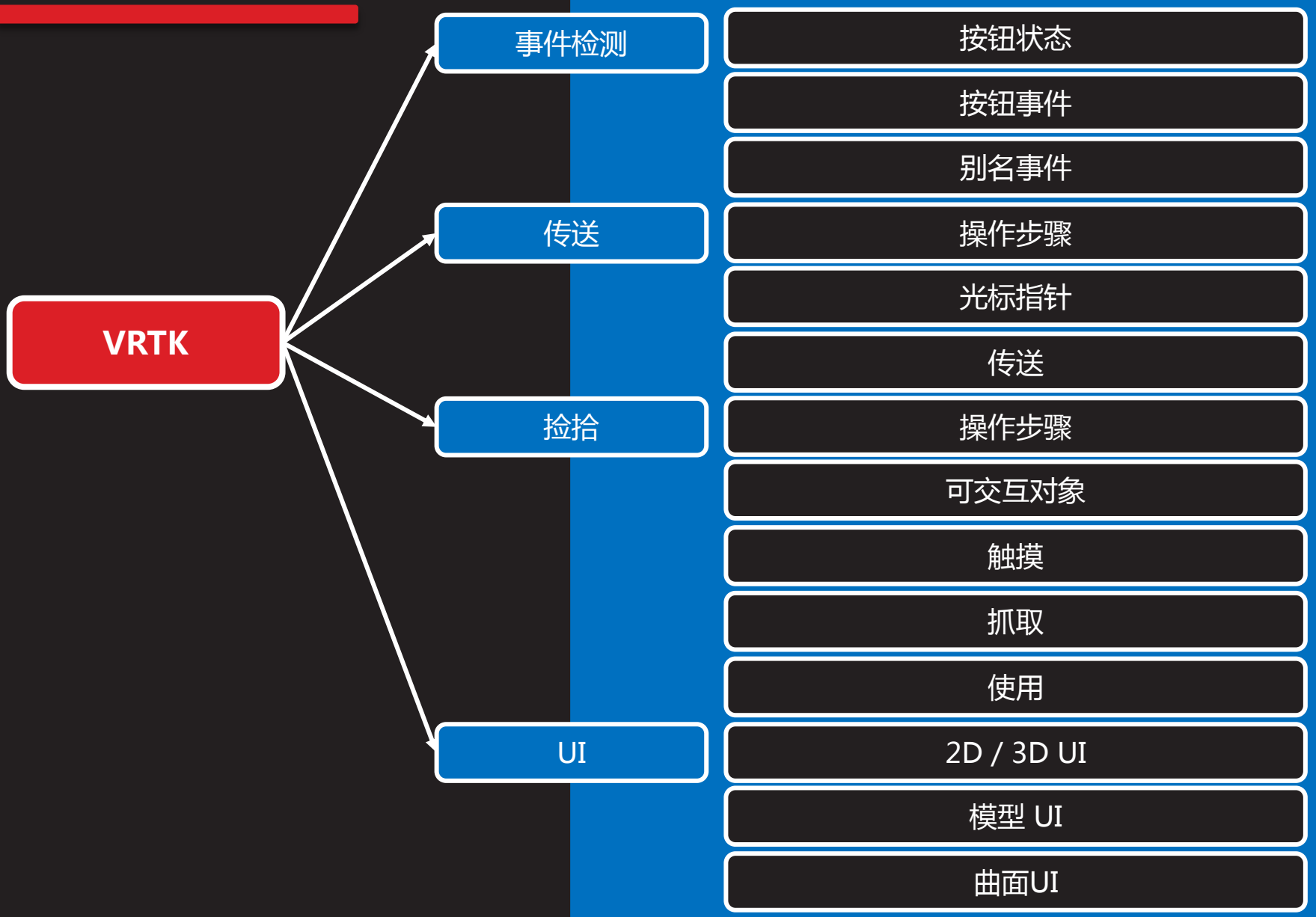
  提供了很多 VR 常用功能，如：控制器激光(贝塞尔曲线)指针、玩家传送、控制器抓取对象、UI……

- 功能介绍：

  http://v.youku.com/v_show/id_XMTYzODA1NzY0NA

知识讲解

VRTK


VRTK
事件检测
传送
捡拾
UI
按钮状态
按钮事件
别名事件
操作步骤
光标指针
传送
操作步骤
可交互对象
触摸
抓取
使用
2D / 3D UI
模型 UI
曲面UI

Technology Tarena 达内科技

事件检测

按钮状态

- 脚本 VRTK_ControllerEvents ：检测用户输入，提供手柄控制器中按钮事件，移动速度等信息。
- 属性：

```
//扳机键是否正在按下
public bool triggerPressed = false;
//扳机键是否正在单击(按到底)
public bool triggerClicked = false;
//扳机键按下程度是否发生改变
public bool triggerAxisChanged = false;
//触摸板是否正在按下
public bool touchpadPressed = false;
//触摸板是否正在触摸
public bool touchpadTouched = false;
//握柄键是否正在按下
public bool gripPressed = false;
//.....
```

知识讲解

按钮事件


```
//扳机键按下事件
public event ControllerInteractionEventHandler TriggerPressed;
//扳机键释放事件
public event ControllerInteractionEventHandler TriggerReleased;

//扳机键开始触摸事件
public event ControllerInteractionEventHandler TriggerTouchStart;
//扳机键结束触摸事件
public event ControllerInteractionEventHandler TriggerTouchEnd;

//扳机键单击事件(按到底)
public event ControllerInteractionEventHandler TriggerClicked;
//扳机键结束单击事件
public event ControllerInteractionEventHandler TriggerUnclicked;
//.....
```

知识讲解

别名事件

知识讲解

```
public enum ButtonAlias
{
    Trigger_Hairline,
    Trigger_Touch,
    Trigger_Press,
    Trigger_Click,
    Grip,
    Touchpad_Touch,
    Touchpad_Press,
    Application_Menu,
    Undefined
}
```

别名枚举

```
public ButtonAlias pointerToggleButton
public ButtonAlias grabToggleButton
public ButtonAlias pointerSetButton
public ButtonAlias useToggleButton
public ButtonAlias uiClickButton
public ButtonAlias menuToggleButton
```

别名按钮

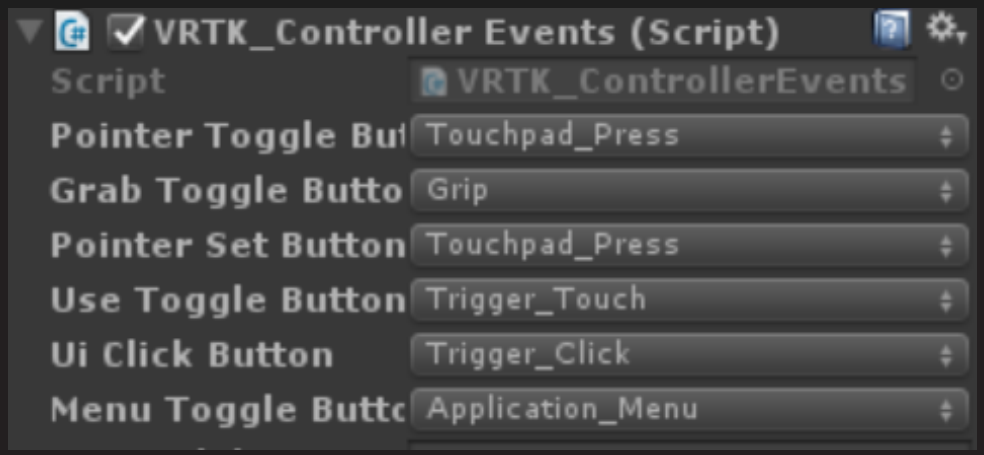
别名事件(续1)

Tarena Technology 达内科技

- Update 每帧检测用户输入，判定别名按钮指定的别名枚举从而引发别名事件。
- 别名事件与别名按钮对应，别名枚举与真实按键对应。

知识讲解

```
//指针打开 别名事件  对应 pointerToggleButton  按钮
public event ControllerInteractionEventHandler AliasPointerOn;
//指针关闭 别名事件  对应 pointerToggleButton  按钮
public event ControllerInteractionEventHandler AliasPointerOff;
//.....
```



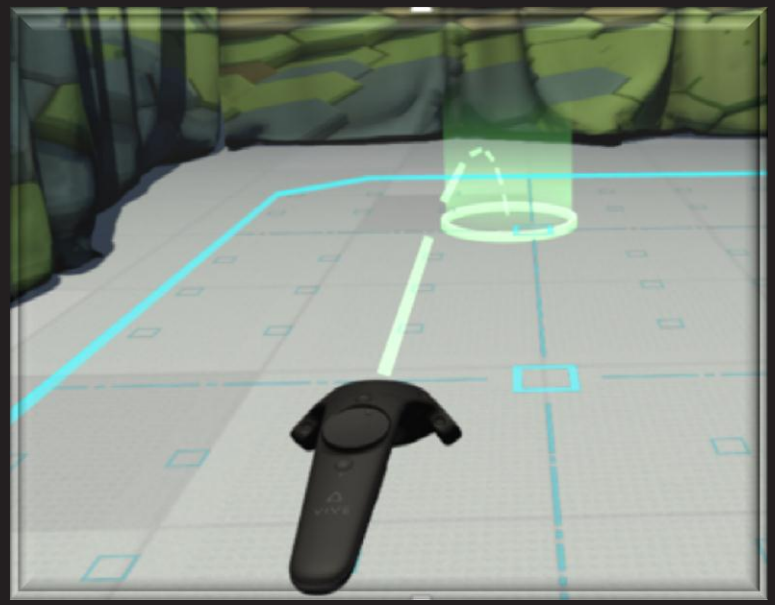
编译器面板

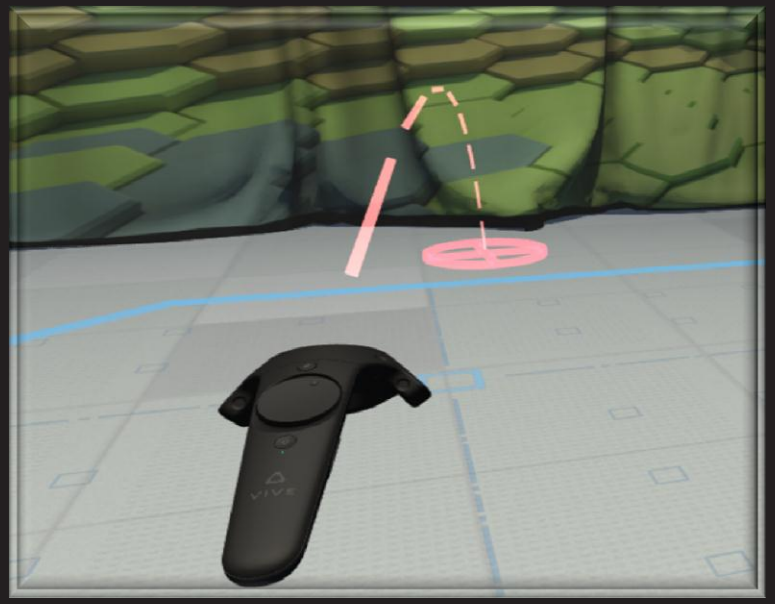
Technology Tarena 达内科技

传送

操作步骤

- 1. 玩家 [CameraRig] 游戏对象中附加传送组件 BasicTeleport，用于改变玩家位置。
- 2. 手柄控制器中附加光标指针组件 SimplePointer，用于发射光标指针 。





知识讲解

光标指针

- 类型：VRTK_SimplePointer / VRTK_BezierPointer
- 默认按下触碰板时从控制器发出激光模拟射线判断指向的物体。
- 通过 VRTK_ControllerEvents 类别名按钮可以改变发射激光的操作方式。

知识讲解

```
public abstract class VRTK_WorldPointer : VRTK_DestinationMarker
{
    4 个引用
    protected virtual void OnEnable()
    {
        controller.AliasPointerOn += EnablePointerBeam;
        controller.AliasPointerOff += DisablePointerBeam;
        controller.AliasPointerSet += SetPointerDestination;
```

光标指针

知识讲解

- 属性：

  Enable Teleport 是否使用传送。

  Controller 如果当前脚本没有附加到控制器中需要指定控制器。

  Show Play Area Cursor 显示游玩区。

  Play Area Cursor Dimensions 游玩区尺寸。

  Handle Play Area Cursor Collisions 对游玩区进行碰撞检测。

  Layers To Ignore 忽略射线检测。

  Ignore Target With Tag Or Class 忽略碰撞检测。

- 事件：

  DestinationMarkerEnter 进入一个目标每帧引发

  DestinationMarkerExit 离开一个目标时引发

  DestinationMarkerSet 选择一个目标时引发

传送

- 类型： VRTK_BasicTeleport / VRTK_HeightAdjustTeleport
- 通过注册指针基类VRTK_DestinationMarker的DestinationMarkerSet事件，实现水平/高度可调节的传送移动。

  属性：

  Blink Transition Speed 传送时眨眼淡入淡出速度

  Distance Blink Delay 基于传送距离的眨眼(保持黑屏)时间

  gnore Target With Tag Or Class 忽略传送目标点(指针设为丢失颜色)

  Gravity Fall Height 模拟下落的最小高度

  事件：

  Teleporting 传送开始时引发

  Teleported 传送完成时引发

Technology Tarena 达内科技

捡拾

操作步骤

- 1. 被捡拾物体附加碰撞器组件 Collider。
- 2. 被捡拾物体附加交互对象组件 VRTK_InteractableObject。
- 3. 交互对象启用抓取属性 Is Grabbable。
- 4. 手柄控制器附加抓取组件 VRTK_InteractGrab。
  （依赖的触摸组件 VRTK_InteractTouch 会自动添加）

知识讲解

可交互对象

- 可交互对象 VRTK_InteractableObject

  添加到需要交互的任何游戏对象上，提供了触摸、抓取与使用的功能。

- 事件：

知识讲解

```
public event InteractableObjectEventHandler InteractableObjectTouched;
public event InteractableObjectEventHandler InteractableObjectUntouched;
public event InteractableObjectEventHandler InteractableObjectGrabbed;
public event InteractableObjectEventHandler InteractableObjectUngrabbed;
public event InteractableObjectEventHandler InteractableObjectUsed;
public event InteractableObjectEventHandler InteractableObjectUnused;
```

- 依赖于手柄控制器中 VRTK_InteractTouch、 VRTK_InteractGrab 、VRTK_InteractUse 组件检测用户操作。

  -- VRTK_InteractTouch 通过 OnTriggerStay 进行检测

  -- VRTK_InteractGrab 注册 AliasGrabOn / AliasGrabOff 事件

  -- VRTK_InteractUse 注册 AliasUseOn / AliasUseOff 事件

触摸

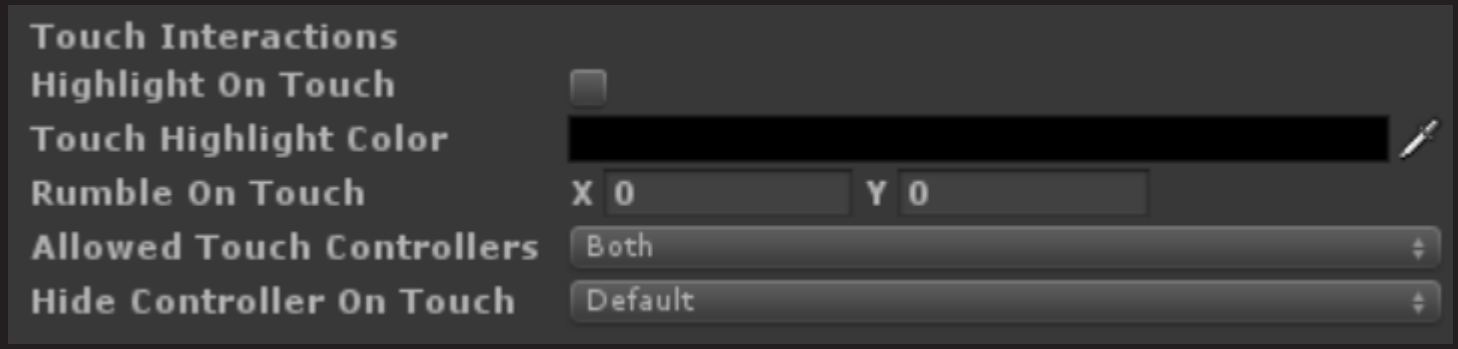
- 触摸 Touch Interactions

  Highlight On Touch 触摸时是否高亮

  Rumble On Touch 触摸反馈 x为时间 y为强度

  Allowed Touch Controllers 允许触摸的控制器

  Hide Controller On Touch 触摸时隐藏控制器方式



知识讲解

抓取

- 抓取 Grab Interactions

  Is Grabbable 是否启用抓取

  Is Droppable 是否启用放下

  Is Swappable 是否启用交换

  Precision Snap 精确捕捉（物体随手柄接触点移动）

  Right / Left Snap Handle 左右手柄对齐位置

知识讲解

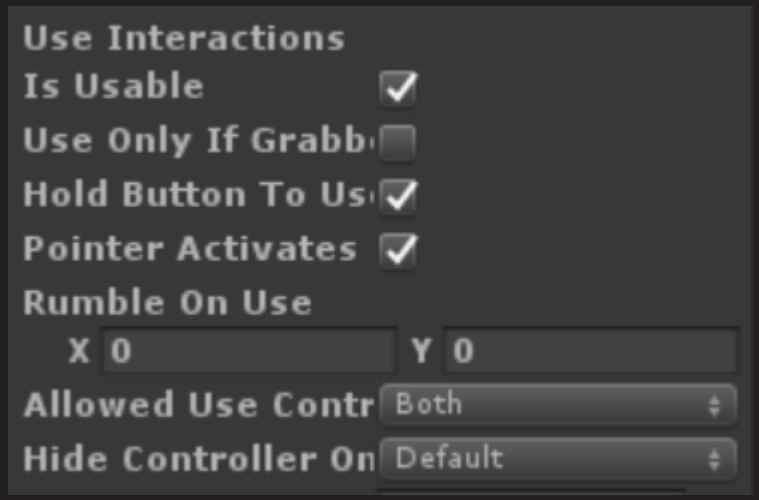
使用

- 使用 Use Interactions

  Is Usable 是否可以使用(Use Toggle Button 控制)

  Use Only If Grabbed 仅仅握住物体时可以使用

  Pointer Activates Use Action 光标激活使用



知识讲解

Technology Tarena 达内科技

UI

2D UI

- 画布 Canvas 三种渲染模式 Render Mode
  - -- Screen Space – Overlay （头盔不支持）
  - -- Screen Space – Camera
  - -- World Space

知识讲解

- 操作步骤：

1. 设置画布渲染模式为 Screen Space – Camera 。

2. 创建摄像机，只负责渲染 UI，并设置到画布的 Render Camera 属性中。

3. 设置摄像机属性：

Clera Flags 为 Depth Only

Culling Mask 为 UI

Depth 大于主摄像机 Depth

3D UI

Tarena Technology 达内科技

知识讲解

3D UI(续1)

- VRTK UI 操作步骤：

1. 手柄控制器添加 VRTK_ControllerEvents 组件，用于事件检测。
2. 手柄控制器添加 VRTK_SimplePointer 组件，用于显示光标指针（可选）。
3. 手柄控制器添加 VRTK_UIPointer 组件，用于实现 VR UI。

   VRTK_UIPointer 在 Start 方法中主要完成以下工作：

   (1)将 StandaloneInputModule 替换为 VRTK_EventSystemVRInput。

   (2)将 GraphicRaycaster 替换为 VRTK_UIGraphicRaycaster。

知识讲解

3D UI(续2)

- 回顾 UGUI 事件处理流程：

1. EventStstem 每帧调用 BaseInputModule 中 Process 方法。
（实现类：StandaloneInputModule / TouchInputModule）

2. 计算光标接触的物体（Graphic ）。
-- Process 方法调用 BaseRaycaster 的 Raycast 方法获取所有 Graphic。
（实现类：GraphicRaycaster / PhysicsRaycaster / Physics2DRaycaster）
-- 通过 Graphic 的 IsRaycastLocationValid 方法确定光标选中的 Graphic。

3. 通过 ExecuteEvents 引发物体的相关事件。
-- 调用 Execute 方法获取相关接口类型对象，再调用其接口方法。

知识讲解

3D UI(续3)

- VRTK UI 事件处理流程：

1. EventStstem 每帧调用 BaseInputModule 中 Process 方法。
（实现类： VRTK_EventSystemVRInput ）

2. 计算光标接触的物体（ Graphic ）。

-- Process 方法调用 BaseRaycaster 的 Raycast 方法获取所有 Graphic。
（实现类： VRTK_UIGraphicRaycaster ）

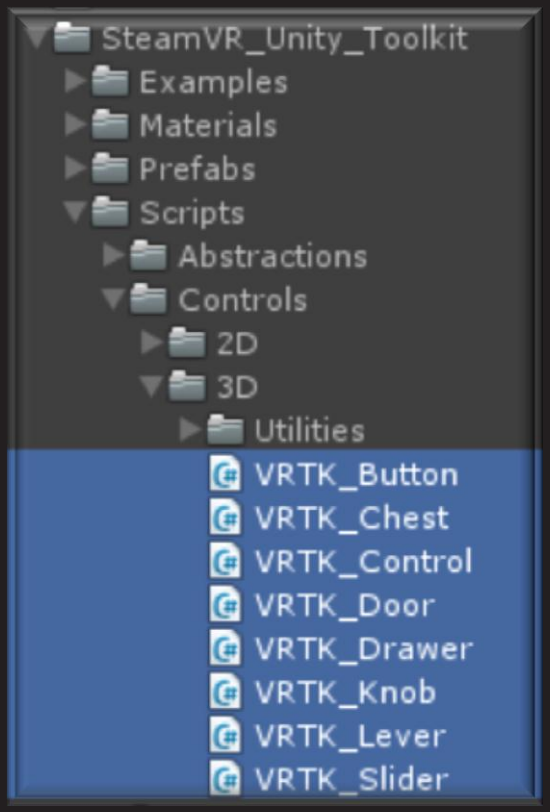
-- 通过 Graphic 的 IsRaycastLocationValid 方法确定手柄选中的 Graphic。
（从 VRTK_UIPointer 所在物体位置，向其 forward 方向发出射线）

3. 通过 ExecuteEvents 引发物体的相关事件。

-- 调用 Execute 方法获取相关接口类型对象，再调用其接口方法。

模型 UI

- 演示场景： Examples / 025_Controls_Overview



知识讲解

曲面 UI

- 创建类继承自 Graphic 类，使用 ExecuteInEditMode 特性修饰。
- 重写 OnPopulateMesh 方法，用于在生成图形顶点时修改图形网格。
- 重写 mainTexture 属性，用于指定贴图。

知识讲解