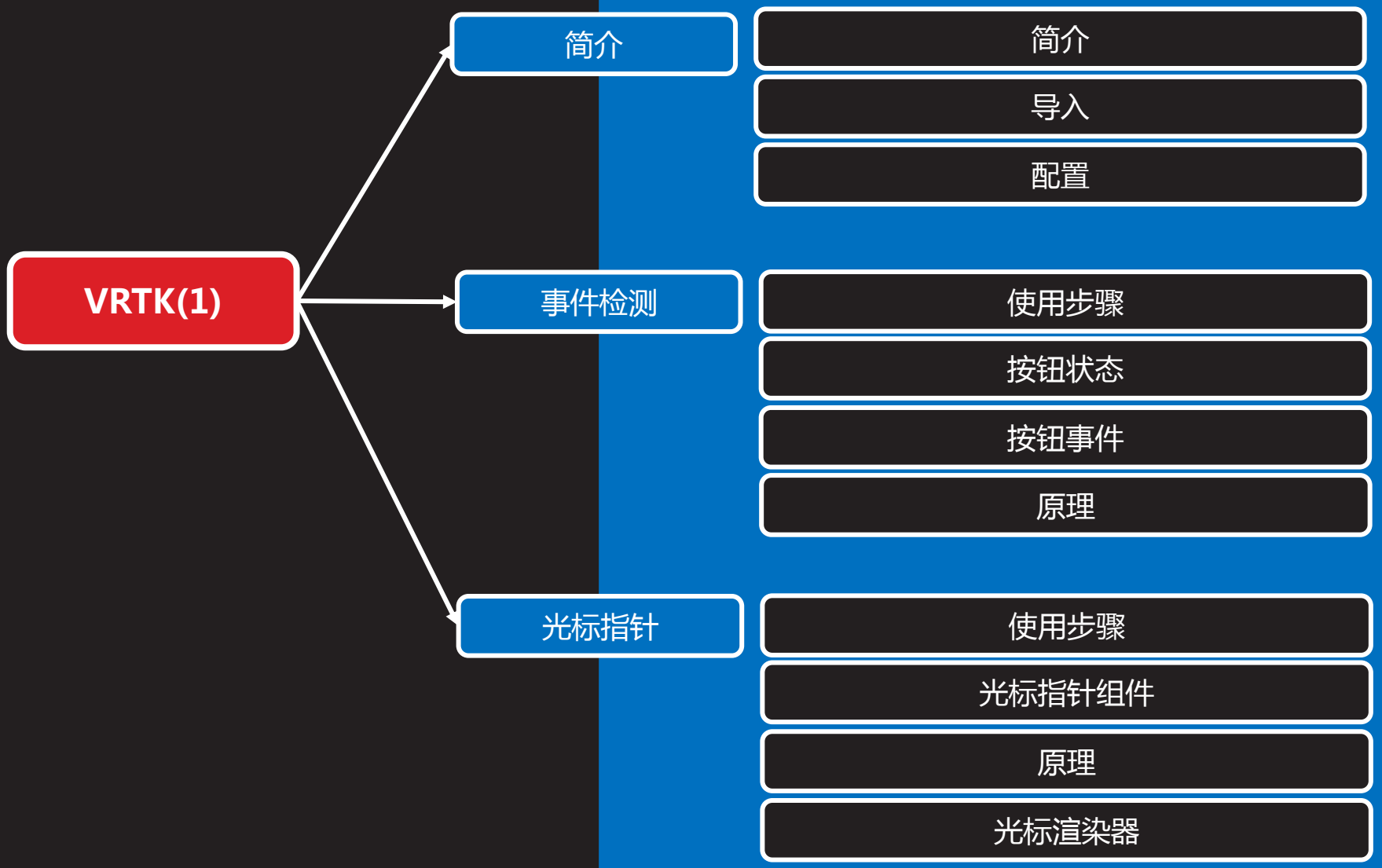
VRTK(1)


VRTK(1)
简介
事件检测
光标指针
简介
导入
配置
使用步骤
按钮状态
按钮事件
原理
使用步骤
光标指针组件
原理
光标渲染器

Technology Tarena 达内科技

简介

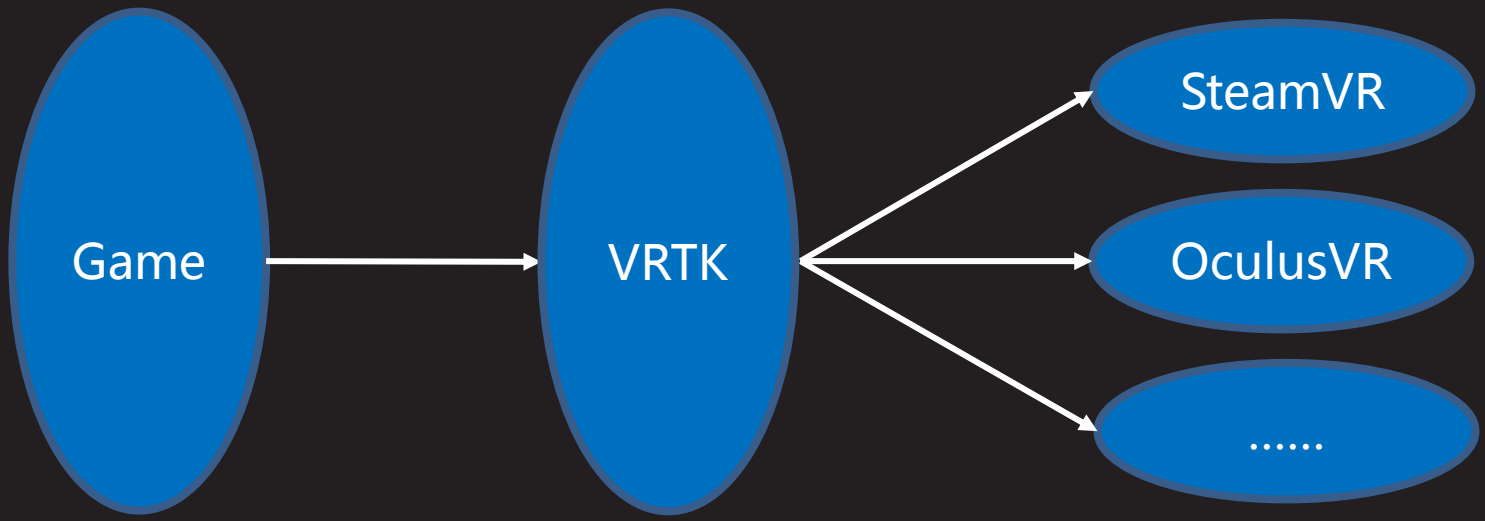
简介

- VR Toolkit 虚拟现实工具包

  统一管理各种 VR 设备SDK，提供一种方式开发 VR 程序的框架。

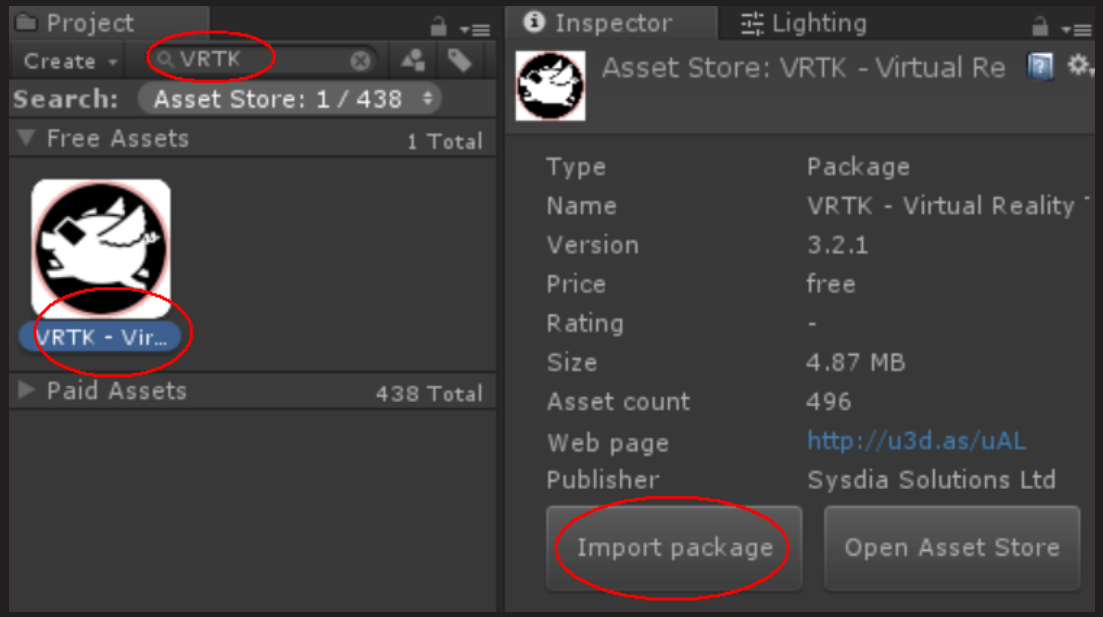
  其中包含了许多 VR 常用功能，如：控制器激光(贝塞尔)指针、虚拟空间运动、UI、虚拟物体交互等。

知识讲解



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

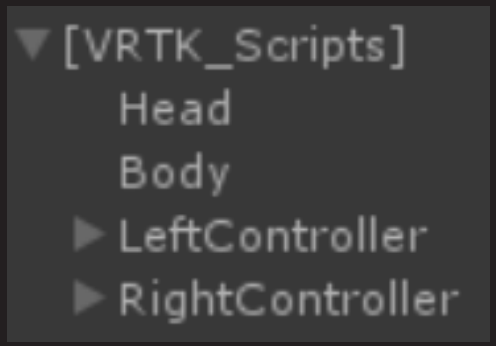
配置

- 1. 创建空物体，命名为 [VRTK_SDKManager] 。
- 2. 创建子物体，命名为 SteamVR。
- 3. 拖拽 [CameraRig] 至 SteamVR 中。
- 4. 为 SteamVR 附加 VRTK_SDK Setup 组件，设置 Quick Select 属性为 SteamVR。
- 5. 为 SDKManager 附加 VRTK_SKDManager 组件，单击 Auto Populate 按钮，自动填充安装的 SDK。

知识讲解

配置

- 6. 创建空物体，命名为 [VRTK_Scripts] ，在编译器环境下作为玩家。
- 7. 创建子物体，命名为 Head，附加 VRTK_SDKObjectAlias 组件，设置 Sdk Object 属性为 Headset 。
- 8. 创建子物体，命名为 Body，附加 VRTK_SDKObjectAlias 组件，设置 Sdk Object 属性为 Boundary。
- 9. 创建子物体，命名为 LeftController，附加 VRTK_Controller Event 组件。
- 10. 创建子物体，命名为 RightController，附加 VRTK_Controller Event 组件。
- 11. 分别将 LeftController、 LeftController 物体拖拽至 VRTK_SKDManager 组件的相应属性中。

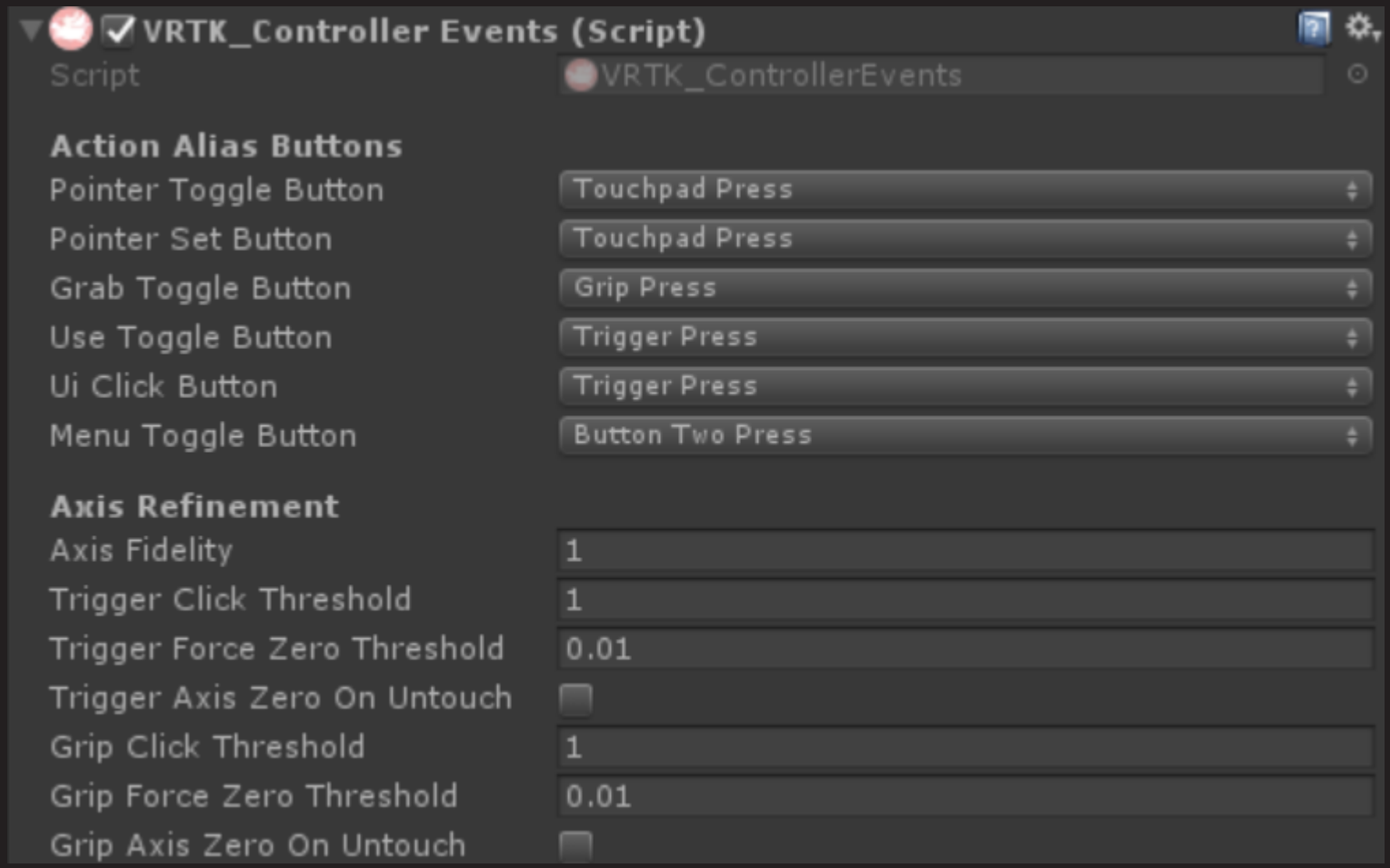


Technology Tarena 达内科技

事件检测

使用步骤

- 在左右手柄控制器中，附加 VRTK_ControllerEvents 组件。



知识讲解

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。
- 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

知识讲解

```
//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;
//.....
```

原理


```
protected virtual bool IsButtonPressed(uint index, ButtonPressTypes type, ulong button)
{
    if (index >= OpenVR.k_unTrackedDeviceIndexInvalid)
    {
        return false;
    }
    SteamVR_Controller.Device device = SteamVR_Controller.Input((int)index);
    switch (type)
    {
        case ButtonPressTypes.Press:
            return device.GetPress(button);
        case ButtonPressTypes.PressDown:
            return device.GetPressDown(button);
        case ButtonPressTypes.PressUp:
            return device.GetPressUp(button);
        case ButtonPressTypes.Touch:
            return device.GetTouch(button);
        case ButtonPressTypes.TouchDown:
            return device.GetTouchDown(button);
        case ButtonPressTypes.TouchUp:
            return device.GetTouchUp(button);
    }
    return false;
}
```

Technology Tarena 达内科技

光标指针

使用步骤

- 1. 手柄控制器附加光标指针组件 VRTK_Pointer，用于发射光标指针；
- 2. 附加贝塞尔光标渲染器 VRTK_BezierPointerRenderer，用于绘制光标。

知识讲解

光标指针组件

- VRTK_Pointer：用于从一个游戏对象发出光线选择另一游戏对象。
- 继承自目标点标记类 VRTK_DestinationMarker。
- 通过激活按钮 Activation Button 与选中按钮 Selection Button改变发射光标指针的操作方式。
- 通过光标渲染器 Pointer Renderer 属性指定光标种类。

知识讲解

光标指针组件

- 属性：

  Enable Teleport 是否启用传送功能。

  Hold Button To Activate 是否持续按下按钮激活光标

  Activate On Enable 是否启动时激活

  Activate Delay 激活延迟时间

  Select On Press 是否按下选中按钮立即选择

  Select Delay 选中延迟时间

  Select After Hover Duration 自动选择的悬停持续时间(0为取消)

知识讲解

光标指针组件

- 事件：

  PointerStateValid 光标激活时引发

  PointerStateInvalid 光标禁止时引发

  ActivationButtonPressed 按下激活按钮时引发

  ActivationButtonReleased 释放激活按钮时引发

  SelectionButtonPressed 按下选择按钮时引发

  SelectionButtonReleased 释放选择按钮时引发

  DestinationMarkerEnter 光标进入目标点

  DestinationMarkerExit 光标离开目标点

  DestinationMarkerHover 光标在目标点悬停

  DestinationMarkerSet 光标选中目标点

知识讲解

原理

- 通过 VRTK_ControllerEvents 类别名事件回调。

```
protected virtual void SubscribeSelectionButton()
{
    if (subscribedSelectionButton != VRTK_ControllerEvents.ButtonAlias.Undefined)
    {
        UnsubscribeSelectionButton();
    }

    if (controller != null)
    {
        controller.SubscribeToButtonAliasEvent(selectionButton, true, DoSelectionButtonPressed);
        controller.SubscribeToButtonAliasEvent(selectionButton, false, DoSelectionButtonReleased);
        controller.SubscribeToButtonAliasEvent(selectionButton, selectOnPress, SelectionButtonAction);
        subscribedSelectionButton = selectionButton;
        currentSelectOnPress = selectOnPress;
    }
}
```

光标渲染器

Technology Tarena 达内科技

- 类型：

  直线光标渲染器 VRTK_StraightPointerRenderer

  贝塞尔曲线光标渲染器 VRTK_BezierPointerRenderer

- 属性：

  Playarea Cursor 游玩区光标，可通过 PlayAreaCursor 对象开启碰撞检测。

  Layers To Ignore 忽略射线检测

  Tracer Visibility 曲线可见性

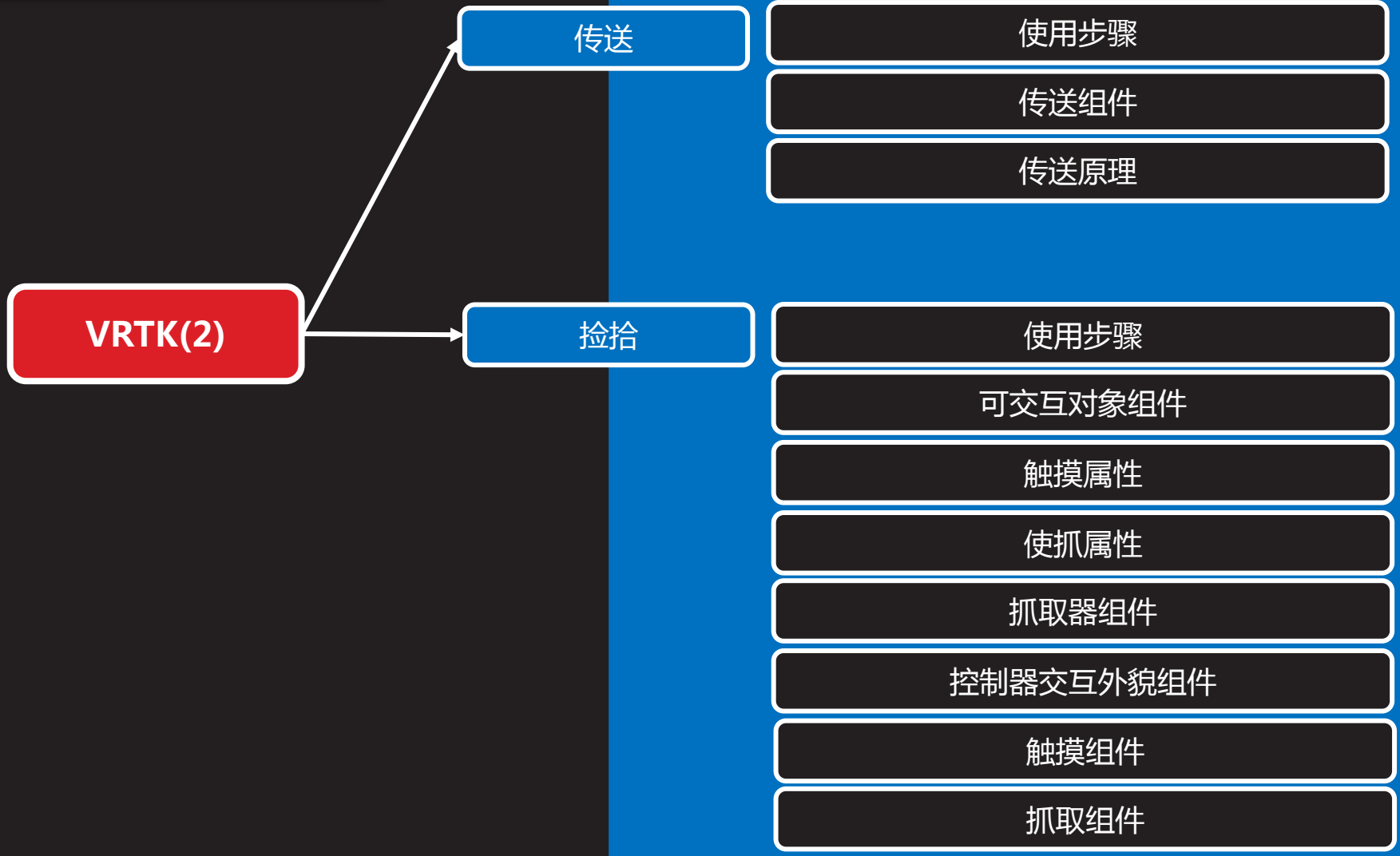
  Cursor Visibility 光标可见性

  Maximum Length 曲线最大长度，X表示前后距离。

  Tracer Density 曲线密度

知识讲解

VRTK(2)

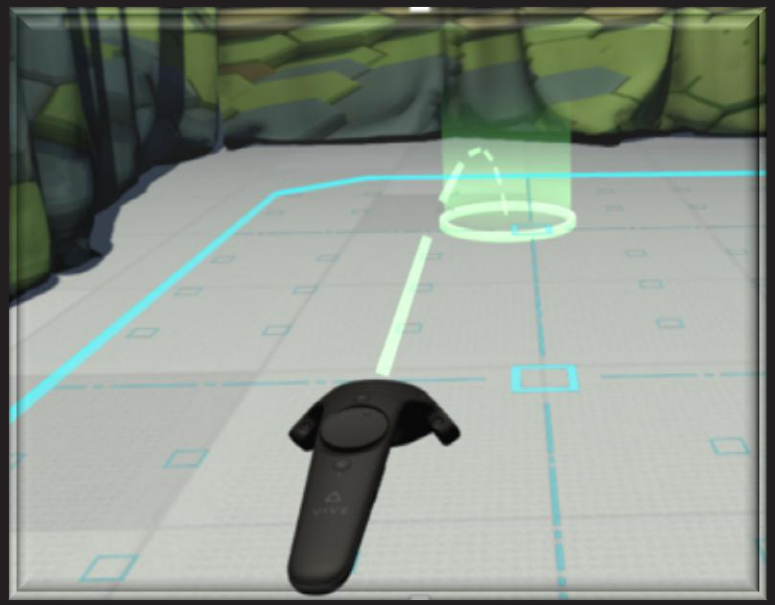

VRTK(2)
传送
使用步骤
传送组件
传送原理
捡拾
使用步骤
可交互对象组件
触摸属性
使抓属性
抓取器组件
控制器交互外貌组件
触摸组件
抓取组件

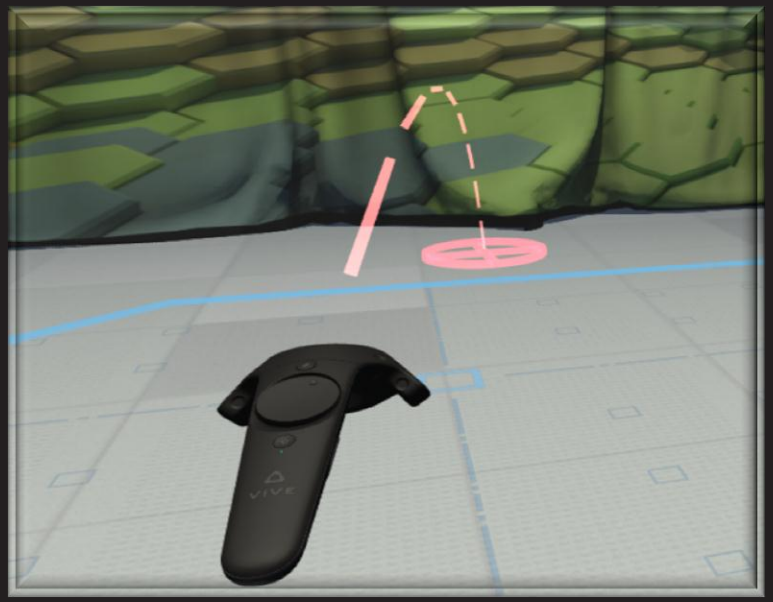
Technology Tarena 达内科技

传送

使用步骤

- 1. 创建贝塞尔光标指针。
- 2. 为 Body 附加传送组件 VRTK_HeightAdjustTeleport ，用于改变玩家位置。





传送组件

- 类型： VRTK_BasicTeleport / VRTK_HeightAdjustTeleport
- 通过注册指针基类VRTK_DestinationMarker的DestinationMarkerSet事件，实现水平/高度可调节的瞬移功能。
- 属性：

  Blink To Color 传送时眨眼颜色

  Blink Transition Speed 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

  Target List Policy 目标策略列表，可设置禁止传送的层。
- 事件：

  Teleporting 传送开始时引发

  Teleported 传送完成时引发

知识讲解

传送原理

- 传送组件启用时，从控制器中获取 VRTK_DestinationMarker 组件(VRTK_Pointer 的父类)，注册 DestinationMarkerSet 事件。
- 通过 VRTK_DeviceFinder 类查找玩家变换组件，使用光标选中的终点修改位置。

知识讲解

传送原理

Tarena Technology 达内科技

```
public virtual void InitDestinationSetListener(GameObject markerMaker, bool register)
{
    if (markerMaker != null)
    {
        VRTK_DestinationMarker[] worldMarkers = markerMaker.GetComponentsInChildren<VRTK_Destinati
        for (int i = 0; i < worldMarkers.Length; i++)
        {
            VRTK_DestinationMarker worldMarker = worldMarkers[i];
            if (register)
            {
                worldMarker.DestinationMarkerSet += new DestinationMarkerEventHandler(DoTeleport);
进
protected virtual void DoTeleport(object sender, DestinationMarkerEventArgs e)
{
    if (enableTeleport && ValidLocation(e.target, e.destinationPosition) && e.enableTeleport)
    {
        StartTeleport(sender, e);
        Vector3 newPosition = GetNewPosition(e.destinationPosition, e.target, e.forceDestinationPosition);
        CalculateBlinkDelay(blinkTransitionSpeed, newPosition);
        Blink(blinkTransitionSpeed);
        Vector3 updatedPosition = SetNewPosition(newPosition, e.target, e.forceDestinationPosition);
        Quaternion updatedRotation = SetNewRotation(e.destinationRotation);
        ProcessOrientation(sender, e, updatedPosition, updatedRotation);
        EndTeleport(sender, e);
    }
}
```

传送原理


```
protected virtual void OnEnable()
{
    VRTK_PlayerObject.SetPlayerObject(gameObject, VRTK_
    headset = VRTK_SharedMethods.AddCameraFade();
    playArea = VRTK_DeviceFinder.PlayAreaTransform();
```

知识

```
protected virtual Vector3 SetNewPosition(Vector3 position, Transform target, bool forceDe
{
    if (ValidRigObjects())
    {
        playArea.position = CheckTerrainCollision(position, target, forceDestinationPosition);
        return playArea.position;
    }
    return Vector3.zero;
}
```

练习：玩家通过贝塞尔曲线传送

- 要求：
  1. 禁止传送到指定区域。
  2. 传送时与墙壁保持间距。
  3. 允许进入触发器内。

课堂练习

练习：玩家通过贝塞尔曲线传送

- 提示：禁止传送到指定区域

1. 为 Body 物体附加 VRTK_Policy List 组件，并设置忽略的层。

2. 将 VRTK_Policy List 组件设置到传送组件的 Target List Policy 属性中。

课堂练习

练习：玩家通过贝塞尔曲线传送

- 提示：传送时与墙壁保持间距

1. 为手柄控制器附加 VRTK_Play Area Cursor 组件，并勾选 Handle Play Area Cursor Collisions 属性。

2. 将 VRTK_Play Area Cursor 组件设置到光标渲染器组件的 Playarea Cursor 属性中。

课堂练习

练习：玩家通过贝塞尔曲线传送

- 提示：允许进入触发器内

-- 射线忽略触发器

1. 为手柄控制器附加 VRTK_Custom Raycast 组件，并设置忽略的层。

2. 将 VRTK_Custom Raycast 组件设置到光标渲染器组件的 Custom Raycast 属性中。

-- 碰撞检测忽略触发器

1. 为手柄控制器附加 VRTK_Policy List 组件，并设置忽略的层。

2. 将 VRTK_Policy List 组件设置到 VRTK_Play Area Cursor 组件的 Target List Policy 属性中。

Technology Tarena 达内科技

捡拾

使用步骤

- 被捡拾物体
  1. 附加碰撞器组件 Collider。
  2. 附加可交互对象组件 VRTK_InteractableObject。
  3. 交互对象启用抓取属性 Is Grabbable。
- 手柄控制器
  1. 附加触摸组件 VRTK_InteractTouch 。
  2. 附加抓取组件 VRTK_InteractGrab。

知识讲解

可交互对象组件

Technology Tarena 达内科技

- 可交互对象 VRTK_InteractableObject

  添加到需要交互的游戏对象上，用于标识可以被触摸、被抓取与使用。

- 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 组件检测用户操作。

  -- VRTK_InteractTouch 通过 OnTriggerXX 进行碰撞检测

  -- VRTK_InteractGrab 通过 Grab Button 属性指定的按钮获取用户输入。

触摸属性

- 触摸选项 Touch Options

  Touch Highlight Color 触摸时的高亮颜色

  Allowed Touch Controllers 允许触摸的控制器

  Ignored Colliders 忽略碰撞检测的标签

知识讲解

抓取属性

- 抓取选项 Grab Options

  Is Grabbable 是否启用抓取

  Hold Button To Grab 是否持续抓取

  Stay Grabbed On Teleport 如果不勾选，传送时释放抓取的物体

  Valid Drop 释放物体方式

  -- No Drop 抓取后不释放

  -- Drop Anywhere 随时释放

  Grab Override Button 覆盖抓取按钮

  Grab Attach Mechanic Script 抓取器对象

  Secondary Grab Action Script 抓取行为对象

抓取器组件 Grab Attach

- 控制器子物体抓取器 VRTK_ChildOfControllerGrabAttach
- 固定关节抓取器 VRTK_FixedJointGrabAttach
- 弹簧关节抓取器 VRTK_SpringJointGrabAttach
- 追踪对象抓取器 VRTK_TrackObjectGrabAttach
- 攀岩抓取器 VRTK_ClimbableGrabAttach
- 属性

  Precision Snap 精确捕捉（物体随手柄接触点移动）

  Right / Left Snap Handle 左右手柄定位点

知识讲解

控制器交互外貌组件

- VRTK_InteractControllerAppearance
- 附加在交互对象物体中。
- 属性

  Hide Controller On Touch 触摸时是否隐藏控制器

  Hid Delay On Touch 触摸隐藏延迟时间

  Hide Controller On Grab 抓取时是否隐藏控制器

  Hid Delay On Grab 抓取隐藏延迟时间

知识讲解

触摸组件

- VRTK_InteractTouch
- 附加在手柄控制器中，通过 OnTriggerXXX 检测与交互对象的接触。
- 属性

  Custom Collider Container 自定义碰撞器

知识讲解

抓取组件

- VRTK_InteractGrab
- 附加在手柄控制器中。
- 属性

Grab Button 抓取按钮

Throw Multiplier 抛出力度增倍

Create Rigid Body When Not Touching 在接触交互对象前握住抓取按钮，可以推动交互对象。

Controller Attach Point 将被抓取物体附加到控制器的点。(如果不使用自带控制器，必须自行指定)

知识讲解

练习：捡拾武器

- 基础步骤：

  1. 为武器附加 Box Collider 组件，设置大小。

  2. 为武器附加 VRTK_InteractableObject 组件，勾选 Is Grabbable 属性、取消 Hold Button To Grab 属性。

  3. 为手柄控制器附加 VRTK_InteractTouch 组件。

  4. 为手柄控制器附加 VRTK_InteractGrab 组件。

练习：捡拾武器

- 武器设置为手柄控制器的子物体：

1. 为武器附加 VRTK_Child Of Controller Grab Attach 组件，并设置到 VRTK_InteractableObject 的 Grab Attach Mechanic Script 属性中。

2. 为手柄控制器添加子物体并附加 Rigidbody 组件，设置到 VRTK_InteractGrab 组件的 Controller Attach Point 属性中。

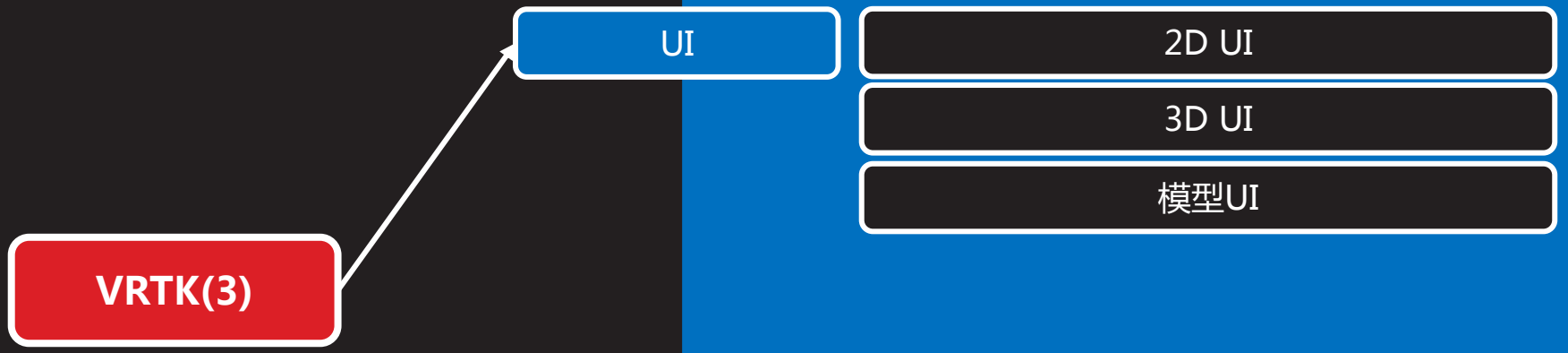
练习：捡拾武器

- 设置抓取的定位点：

  1. 为武器添加子物体作为定位点，并设置到 VRTK_ChildOfControllerGrabAttach 组件的 Controller Attach Point 属性中。

  2. 反复抓取，以调整定位点的方位。

VRTK(3)



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
- 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

知识讲解

3D UI

- VRTK UI 操作步骤：

1. 手柄控制器附加 VRTK_Pointer 组件，用于发射光标指针。

2. 附加直线渲染器 VRTK_StraightPointerRenderer，用于绘制光标。

3. 手柄控制器附加 VRTK_UIPointer，用于将 EventSystem 替换为 VRTK_EventSystem 组件。

4. 画布添加 VRTK_UICanvas 组件，用于初始化 UGUI 画布。

在 OnEnable 方法中完成以下工作：

-- 将 GraphicRaycaster 替换为 VRTK_UIGraphicRaycaster。

-- 添加 Box Collider 组件，并设置为画布大小。

-- 添加 Rigidbody组件，并勾选 Is Kinematic 属性。

3D UI(续1)

- 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2)

- VRTK UI 事件处理流程：

1. VRTK_EventSystem 创建 VRTK_VRInputModule 对象并每帧调用其 Process 方法。

知识讲解

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 ）

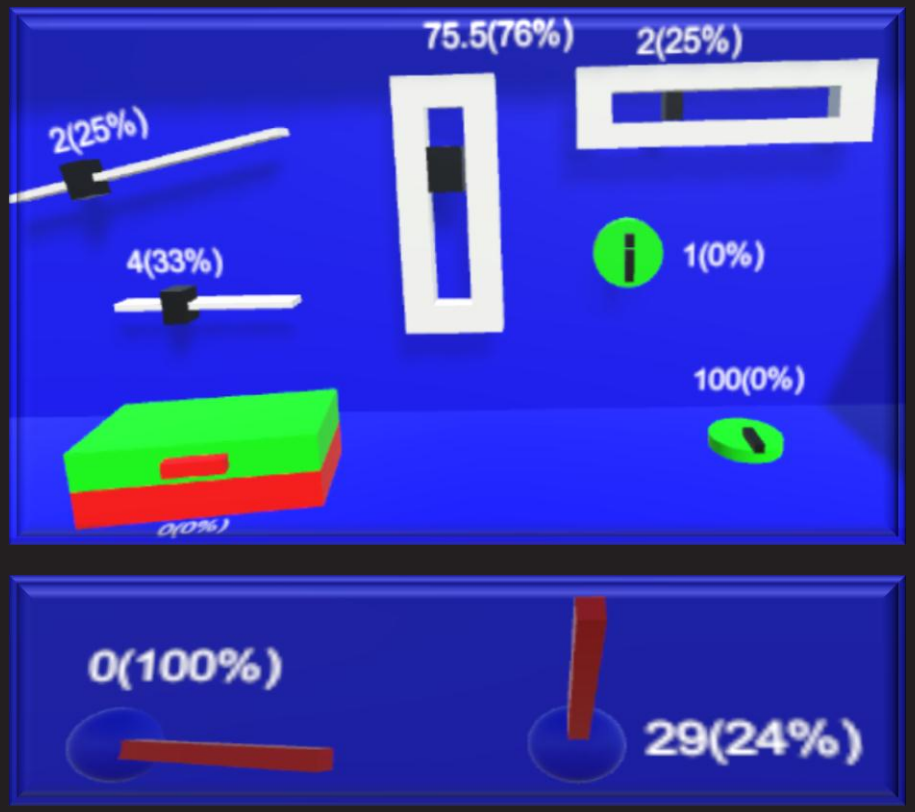
-- 通过 Graphic 的 IsRaycastLocationValid 方法确定手柄选中的 Graphic。（从 VRTK_UIPointer 所在物体位置，向其 forward 方向发出射线）

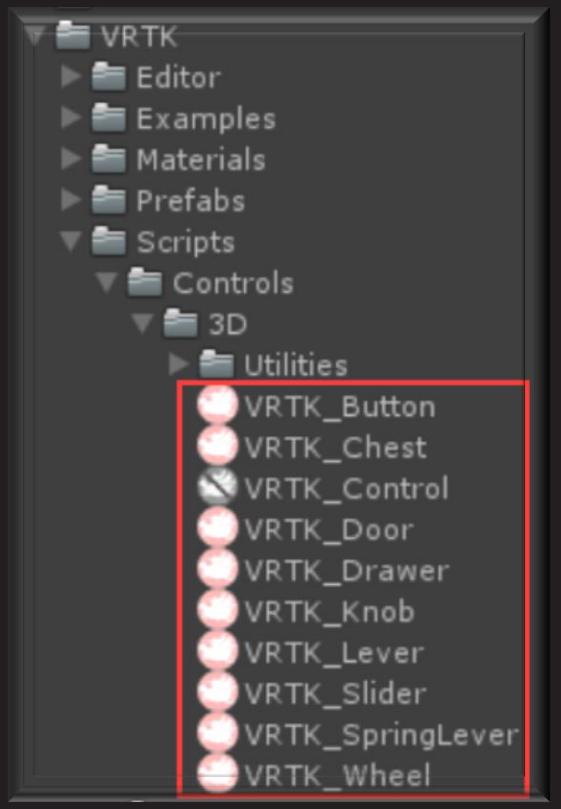
3. 通过 ExecuteEvents 引发物体的相关事件。

-- 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