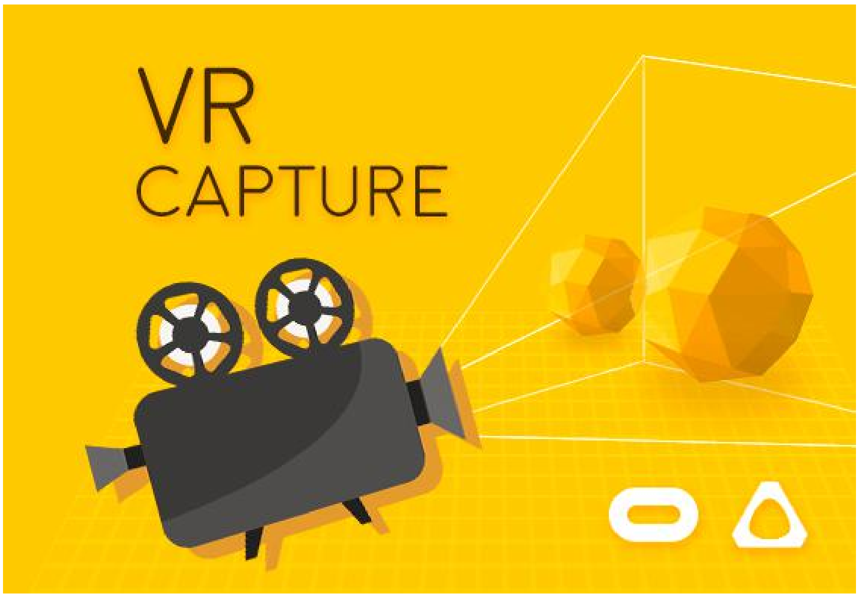


VR Capture Plugin

Vive Component

Version 1.1

Created by RockVR

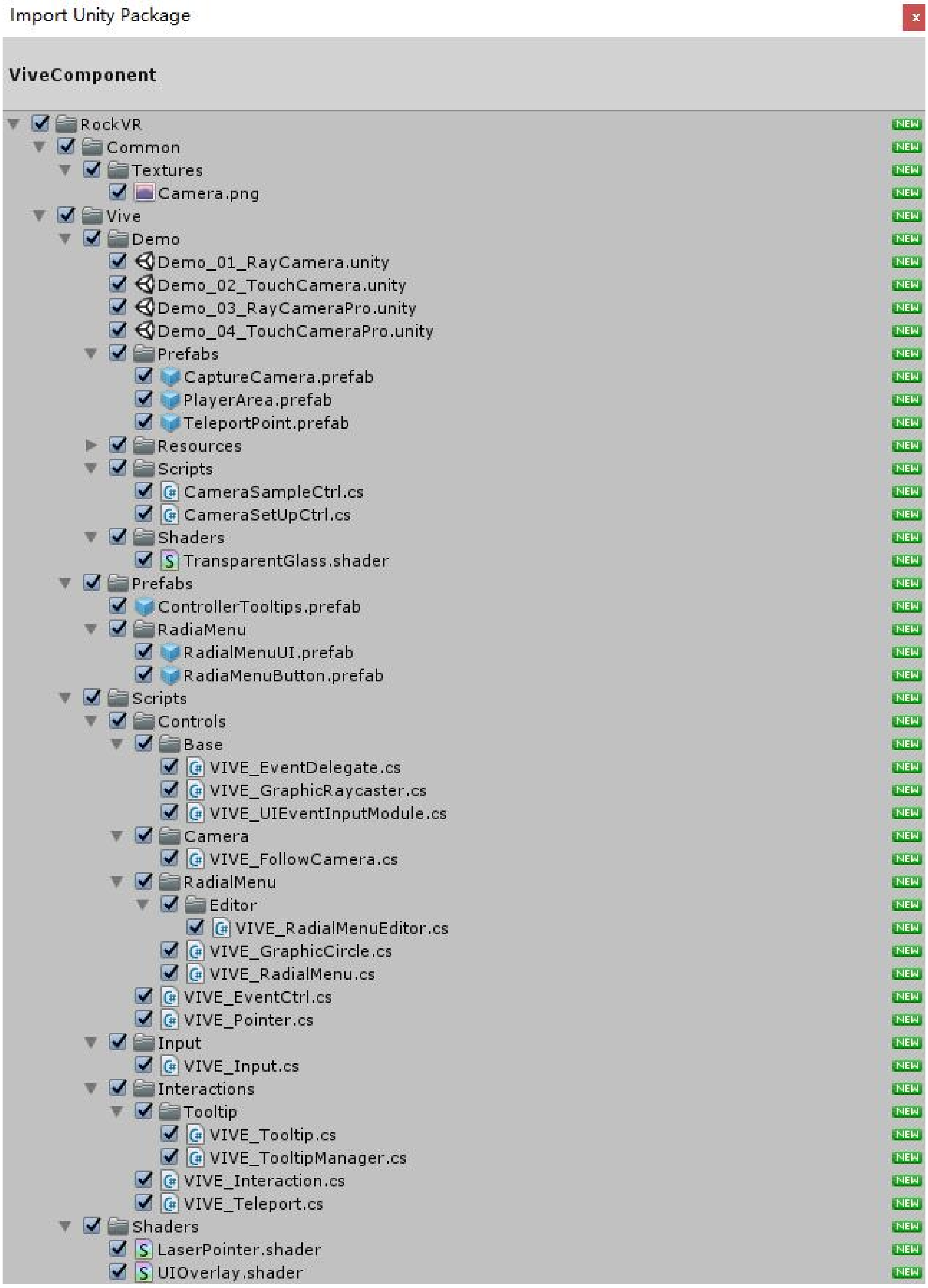
http://www.rockvr.com/vrcapture

Contact: dev@rockvr.com

1. Introduction and Overview

Vive Component is a unity package can help you to quickly understand how video capture work in VR scenes with Vive device.

When you import Vive Component into your Unity project, the following file will be added:



Import Unity Package

ViveComponent

```
▼ ☑ RockVR                                      NEW
  ▼ ☑ Common                                    NEW
    ▼ ☑ Textures                                NEW
        ☑ Camera.png                            NEW
  ▼ ☑ Vive                                      NEW
    ▼ ☑ Demo                                    NEW
        ☑ Demo_01_RayCamera.unity               NEW
        ☑ Demo_02_TouchCamera.unity             NEW
        ☑ Demo_03_RayCameraPro.unity            NEW
        ☑ Demo_04_TouchCameraPro.unity          NEW
      ▼ ☑ Prefabs                               NEW
          ☑ CaptureCamera.prefab                NEW
          ☑ PlayerArea.prefab                   NEW
          ☑ TeleportPoint.prefab                NEW
      ▶ ☑ Resources                             NEW
      ▼ ☑ Scripts                               NEW
          ☑ CameraSampleCtrl.cs                 NEW
          ☑ CameraSetUpCtrl.cs                  NEW
      ▼ ☑ Shaders                               NEW
          ☑ TransparentGlass.shader             NEW
    ▼ ☑ Prefabs                                 NEW
        ☑ ControllerTooltips.prefab             NEW
      ▼ ☑ RadiaMenu                             NEW
          ☑ RadialMenuUI.prefab                 NEW
          ☑ RadiaMenuButton.prefab              NEW
    ▼ ☑ Scripts                                 NEW
      ▼ ☑ Controls                              NEW
        ▼ ☑ Base                                NEW
            ☑ VIVE_EventDelegate.cs             NEW
            ☑ VIVE_GraphicRaycaster.cs          NEW
            ☑ VIVE_UIEventInputModule.cs        NEW
        ▼ ☑ Camera                              NEW
            ☑ VIVE_FollowCamera.cs              NEW
        ▼ ☑ RadialMenu                          NEW
          ▼ ☑ Editor                            NEW
              ☑ VIVE_RadialMenuEditor.cs        NEW
            ☑ VIVE_GraphicCircle.cs             NEW
            ☑ VIVE_RadialMenu.cs                NEW
          ☑ VIVE_EventCtrl.cs                   NEW
          ☑ VIVE_Pointer.cs                     NEW
      ▼ ☑ Input                                 NEW
          ☑ VIVE_Input.cs                       NEW
      ▼ ☑ Interactions                          NEW
        ▼ ☑ Tooltip                             NEW
            ☑ VIVE_Tooltip.cs                   NEW
            ☑ VIVE_TooltipManager.cs            NEW
          ☑ VIVE_Interaction.cs                 NEW
          ☑ VIVE_Teleport.cs                    NEW
    ▼ ☑ Shaders                                 NEW
        ☑ LaserPointer.shader                   NEW
        ☑ UIOverlay.shader                      NEW
```

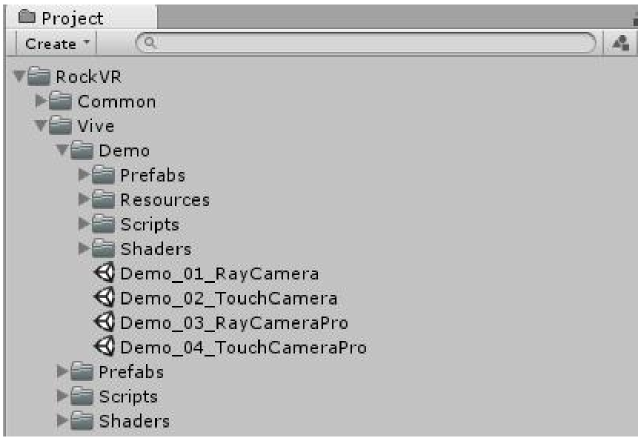
RockVR/Vive/Demo	Contains the scene file and all other assets for a fully functional demonstration of Video Capture with vive interaction.
RockVR/Vive/ Prefabs	Contains useful prefabs can be dragged and dropped in to your scene.
RockVR/Vive/ Scripts	Contains the core vive interaction logic scripts.
RockVR/Vive/ Shaders	Shaders work with materials for vive interaction.

This guide covers integrating VR Capture to your own Unity project with HTC Vive device, and provides a detailed explanation on how the package works under the hood.

If you have any questions, feedback or having issues, please contact us directly at *dev@rockvr.com*. We will respond to you as quickly as possible.

2. Demo Quick Start

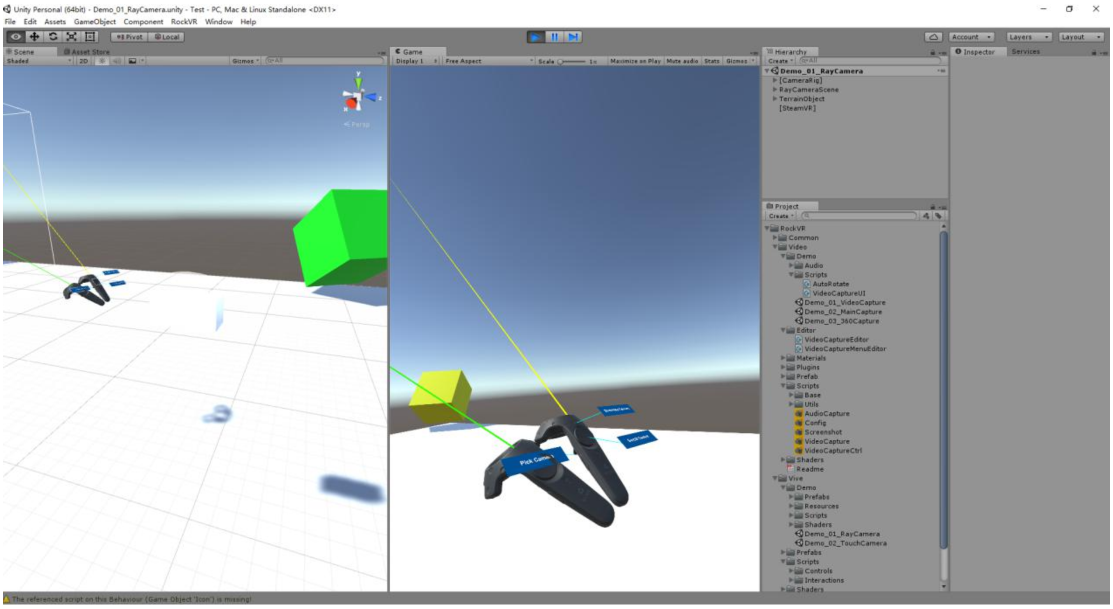
VR Capture *ViveComponent* come with several demos to help you understanding functionality quickly. Start by importing [*SteamVR*]() plugin then importing all *ViveComponent* package assets included demo scenes files.



```
▼ RockVR
    ▶ Common
    ▼ Vive
        ▼ Demo
            ▶ Prefabs
            ▶ Resources
            ▶ Scripts
            ▶ Shaders
              Demo_01_RayCamera
              Demo_02_TouchCamera
              Demo_03_RayCameraPro
              Demo_04_TouchCameraPro
        ▶ Prefabs
        ▶ Scripts
        ▶ Shaders
```

We provide several ways of interaction with recording camera, including touch, laser ray controller, etc.

2.1 Laser Ray Controller Demo

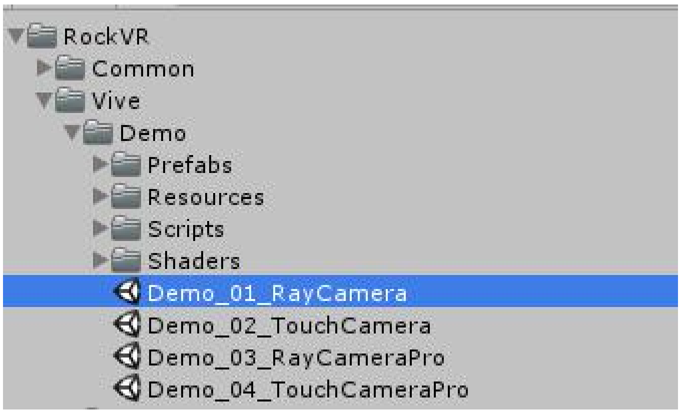


```
▼ Demo_01_RayCamera
    ▶ [CameraRig]
    ▶ RayCameraScene
    ▶ TerrainObject
      [SteamVR]
```

```
▼ RockVR
    ▶ Common
    ▼ Video
        ▼ Demo
            ▶ Audio
            ▼ Scripts
                  AutoRotate
                  VideoCaptureUI
              Demo_01_VideoCapture
              Demo_02_MainCapture
              Demo_03_360Capture
        ▼ Editor
              VideoCaptureEditor
              VideoCaptureMenuEditor
        ▶ Materials
        ▶ Plugins
        ▶ Prefab
        ▼ Scripts
            ▶ Base
            ▶ Utils
              AudioCapture
              Config
              Screenshot
              VideoCapture
              VideoCaptureCtrl
        ▶ Shaders
          Readme
    ▼ Vive
        ▼ Demo
            ▶ Prefabs
            ▶ Resources
            ▶ Scripts
            ▶ Shaders
              Demo_01_RayCamera
              Demo_02_TouchCamera
        ▶ Prefabs
        ▼ Scripts
            ▶ Controls
            ▶ Interactions
        ▶ Shaders
```

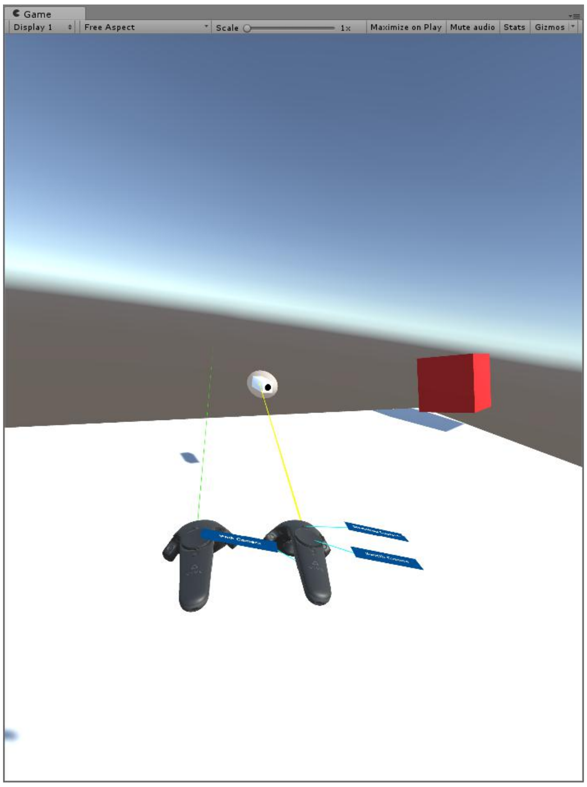
The referenced script on this Behaviour (Game Object 'Icon') is missing!

Step 1: Open the demo scene located in */RockVR/Vive/Demo/Demo_01_RayCamera* :

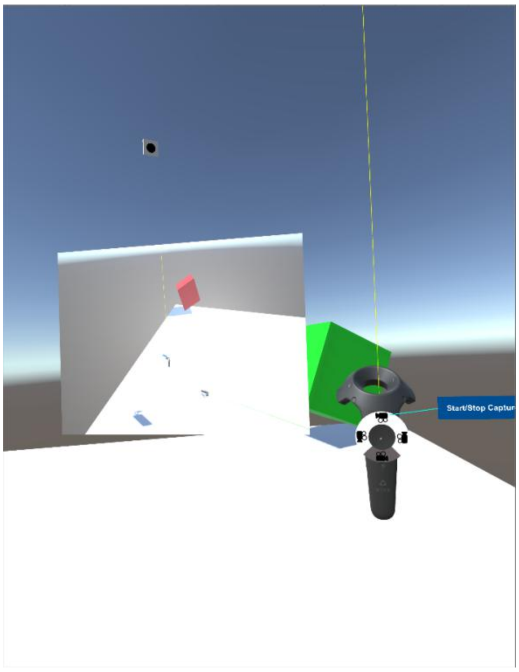


- RockVR
  - Common
  - Vive
    - Demo
      - Prefabs
      - Resources
      - Scripts
      - Shaders
      - Demo_01_RayCamera
      - Demo_02_TouchCamera
      - Demo_03_RayCameraPro
      - Demo_04_TouchCameraPro

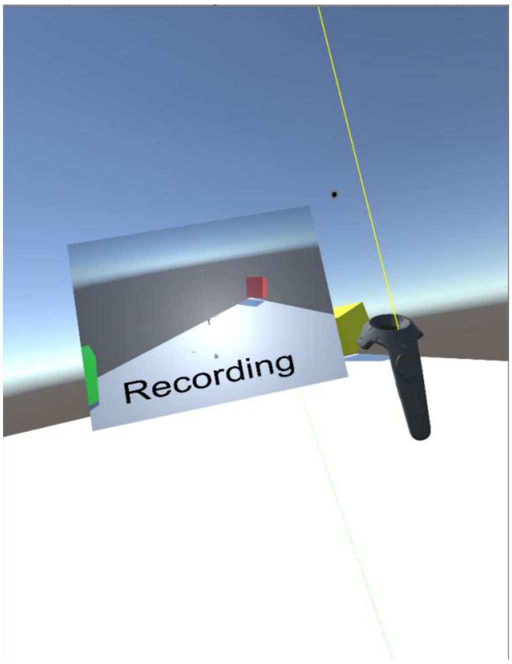
Step 2: Interact with recording camera. You need using laser ray to select the camera:



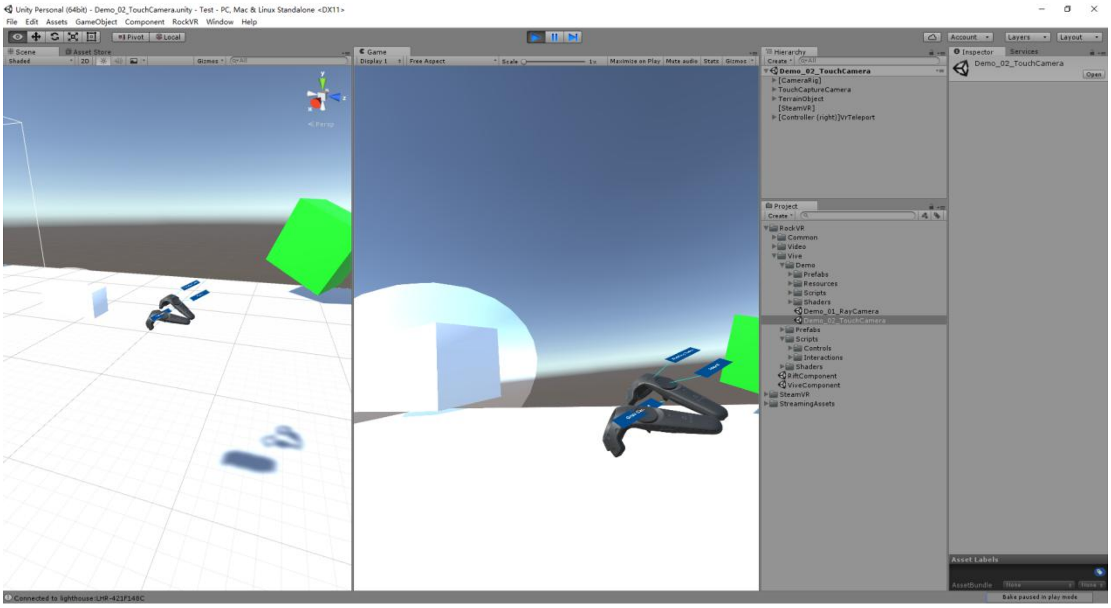
Step 3: Once you grab the camera, you can use radial menu to choose the shooting position with pre-set value:



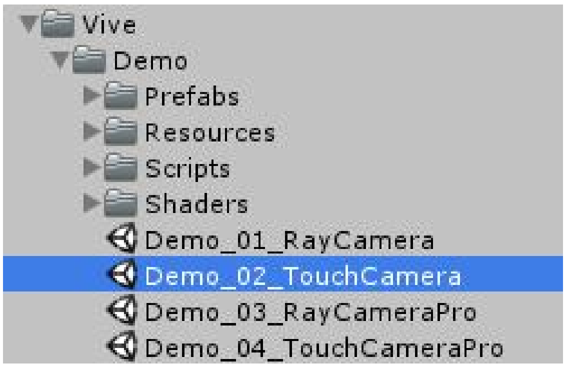
Step 4: After your perfect camera position set, press trigger to start video recording session:



2.1 Touch Controller Demo

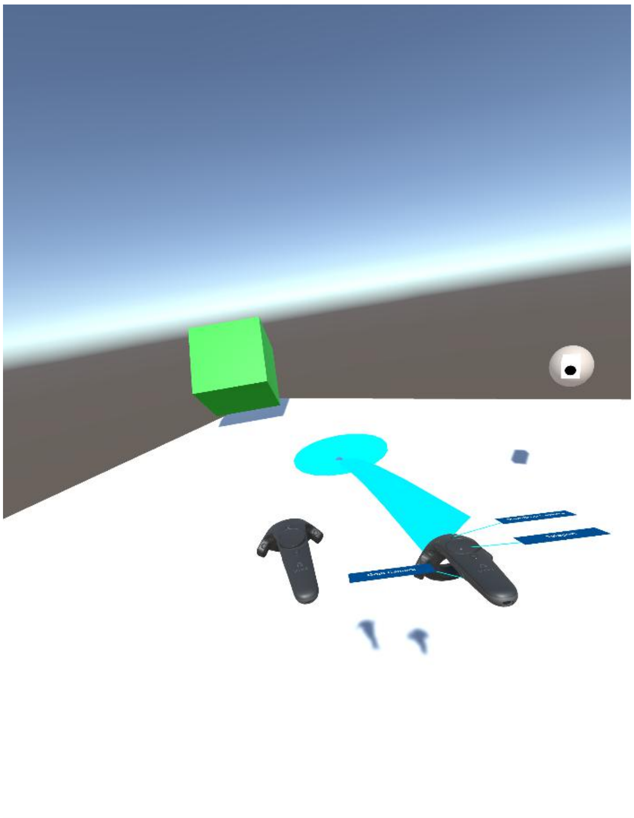

Unity Personal (64bit) - Demo_02_TouchCamera.unity - Test - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component RockVR Window Help
Scene
Game
Hierarchy
Demo_02_TouchCamera
[CameraRig]
TouchCaptureCamera
TerrainObject
[SteamVR]
[Controller (right)]VrTeleport
Project
Inspector
Demo_02_TouchCamera
Connected to lighthouse:LHR-421F148C

Step 1: Open the demo scene located in *[/RockVR/Vive/Demo/Demo_02_TouchCamera :](/RockVR/Vive/Demo/Demo_02_TouchCamera)*

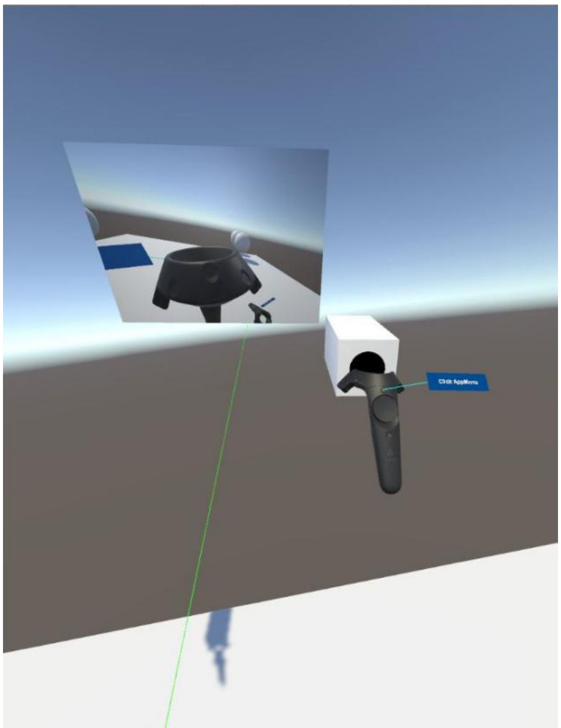


Vive
Demo
Prefabs
Resources
Scripts
Shaders
Demo_01_RayCamera
Demo_02_TouchCamera
Demo_03_RayCameraPro
Demo_04_TouchCameraPro

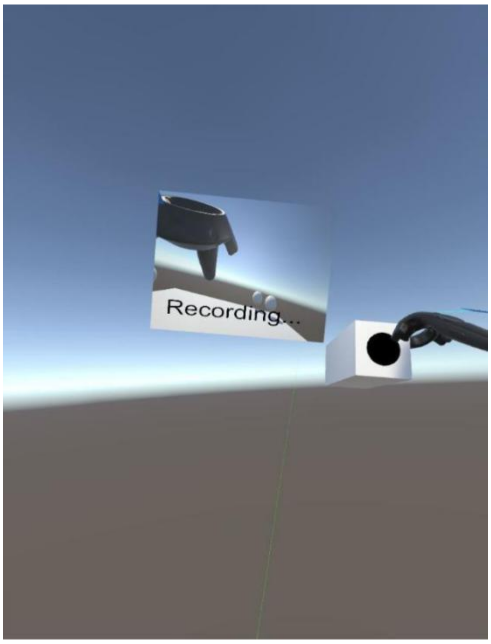
Step 2: Teleport to camera object by interaction with touchpad:



Step 3: Grab the camera and adjust to desired position and angle:



Step 4: Press the trigger to start video recording session, also you can grab camera and move it around while recording:

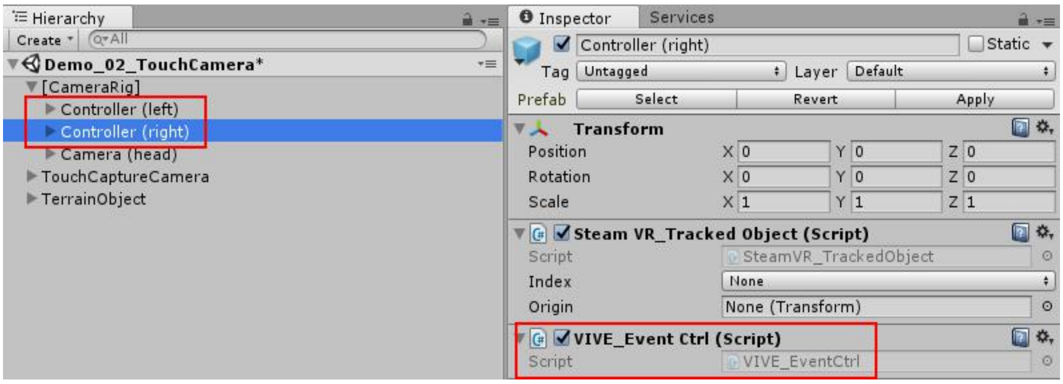


There are a few more demo you can try, *Demo_03_RayCameraPro* and *Demo_04_TouchCameraPro* both of them is recording by VideoCapturePro. The setup process should be same.

3. Integration Guide

VIVE_Interation module implemented the core function of VR interaction control.

Step 1: Attach *VIVE_EventCtrl* script to the Steam VR devices controller you want to control.



Demo_02_TouchCamera*
[CameraRig]
Controller (left)
Controller (right)
Camera (head)
TouchCaptureCamera
TerrainObject

☑ Controller (right) □ Static
Tag Untagged Layer Default
Prefab

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

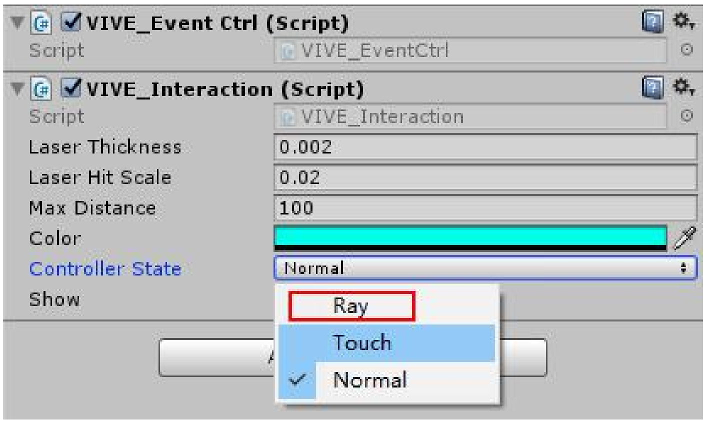
☑ Steam VR_Tracked Object (Script)
Script SteamVR_TrackedObject
Index None
Origin None (Transform)

☑ VIVE_Event Ctrl (Script)
Script VIVE_EventCtrl

Note: *VIVE_EventCtrl* is based on the *VIVE_EventDelegate* and SteamVR plugin, it is an event script to control the Vive handle devices.

Step 2: Adding different interactive features of the script to the Steam VR devices after the first step.

VIVE_Interaction- Set the interaction patterns, choose Ray or touch to interacting.



VIVE_Event Ctrl (Script)

Script: VIVE_EventCtrl

VIVE_Interaction (Script)

Script: VIVE_Interaction

Laser Thickness: 0.002

Laser Hit Scale: 0.02

Max Distance: 100

Color

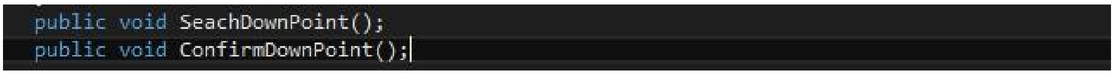
Controller State: Normal

Show

VIVE_Pointer - This script encapsulates system event to take response for user interaction.

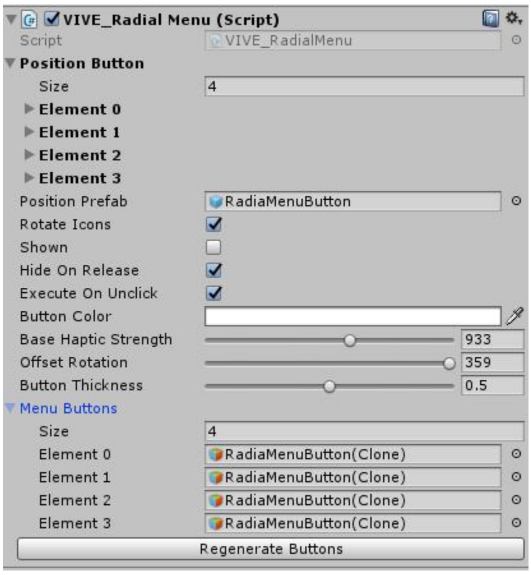
VIVE_UIEventInputModule is designed to work as you would expect how a Vive controller input to work. Including button presses, dragging, and touch events.

VIVE_Teleport - Implemented functionality of teleport in VR scene. Enable *SeachDownPoint* and *ConfirmDownPoint* function to implement teleport.



```
public void SeachDownPoint();
public void ConfirmDownPoint();
```

VIVE_RadialMenu - Set radial menu attribute, and binding keys to listener events.



VIVE_Radial Menu (Script)

Script: VIVE_RadialMenu

Position Button

- Size: 4
- **Element 0**
- **Element 1**
- **Element 2**
- **Element 3**

Position Prefab: RadiaMenuButton

Rotate Icons: ☑

Shown: □

Hide On Release: ☑

Execute On Unclick: ☑

Button Color:

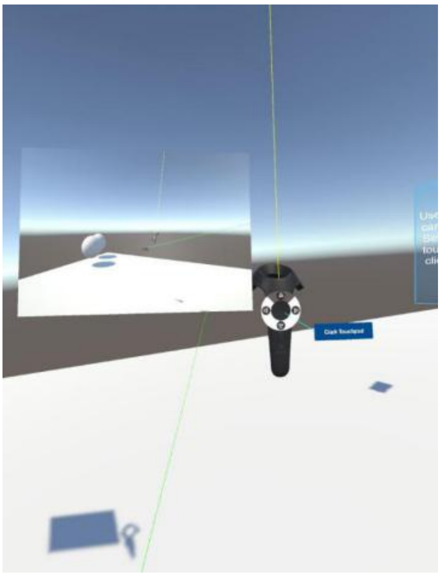
Base Haptic Strength: 933

Offset Rotation: 359

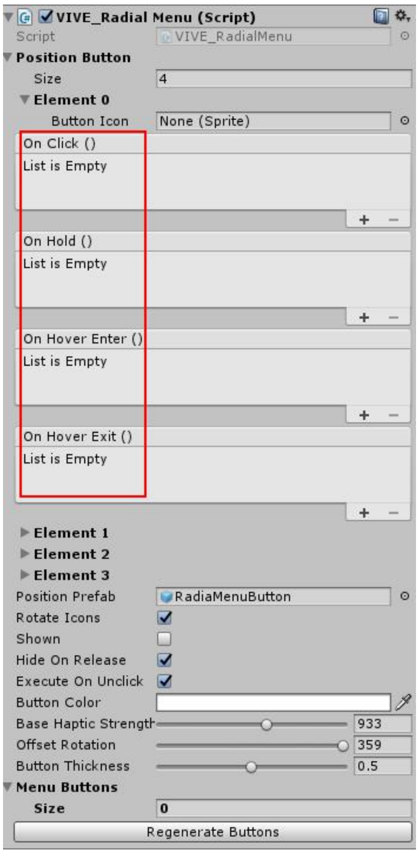
Button Thickness: 0.5

Menu Buttons

- Size: 4
- Element 0: RadiaMenuButton(Clone)
- Element 1: RadiaMenuButton(Clone)
- Element 2: RadiaMenuButton(Clone)
- Element 3: RadiaMenuButton(Clone)



Add different events to objects corresponding to *RadialMenuButtons* state.



VIVE_Radial Menu (Script)

Script: VIVE_RadialMenu

Position Button

- Size: 4
- **Element 0**
  - Button Icon: None (Sprite)
  - On Click ()
    List is Empty
  - On Hold ()
    List is Empty
  - On Hover Enter ()
    List is Empty
  - On Hover Exit ()
    List is Empty
- **Element 1**
- **Element 2**
- **Element 3**

Position Prefab: RadiaMenuButton

Rotate Icons: ☑

Shown: □

Hide On Release: ☑

Execute On Unclick: ☑

Button Color:

Base Haptic Strength: 933

Offset Rotation: 359

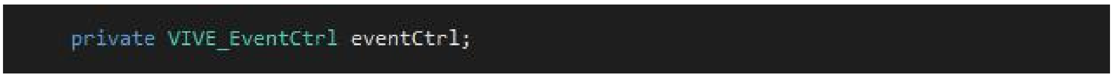
Button Thickness: 0.5

Menu Buttons

- **Size**: 0

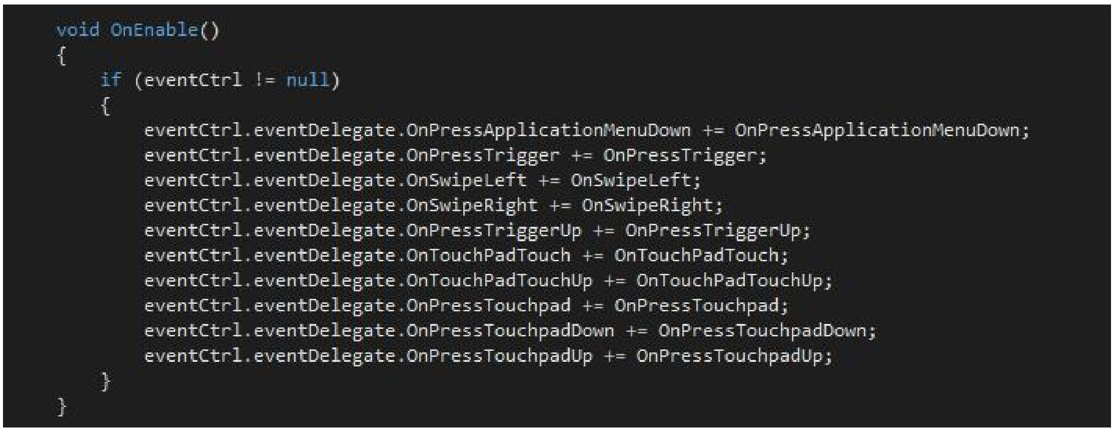
Step 3: Create a control management script to manage the handle events.

Create *VIVE_EventCtrl* object.



```
private VIVE_EventCtrl eventCtrl;
```

Register to create a delegated events.



```
void OnEnable()
{
    if (eventCtrl != null)
    {
        eventCtrl.eventDelegate.OnPressApplicationMenuDown += OnPressApplicationMenuDown;
        eventCtrl.eventDelegate.OnPressTrigger += OnPressTrigger;
        eventCtrl.eventDelegate.OnSwipeLeft += OnSwipeLeft;
        eventCtrl.eventDelegate.OnSwipeRight += OnSwipeRight;
        eventCtrl.eventDelegate.OnPressTriggerUp += OnPressTriggerUp;
        eventCtrl.eventDelegate.OnTouchPadTouch += OnTouchPadTouch;
        eventCtrl.eventDelegate.OnTouchPadTouchUp += OnTouchPadTouchUp;
        eventCtrl.eventDelegate.OnPressTouchpad += OnPressTouchpad;
        eventCtrl.eventDelegate.OnPressTouchpadDown += OnPressTouchpadDown;
        eventCtrl.eventDelegate.OnPressTouchpadUp += OnPressTouchpadUp;
    }
}
```

4. Feedback

If you have any feedback to *VR Capture* plugin, please email us directly, your suggestion will be very valuable to us. If you plan integrate a plugin into your game, please contact us by *dev@rockvr.com* and we will provide more help to let you share your awesome game more efficient.