



# VR Capture Plugin

## Rift Component

Version 1.1

Created by RockVR

<http://www.rockvr.com/vrcapture>

Contact: [dev@rockvr.com](mailto:dev@rockvr.com)

# 1. Introduction and Overview

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

When you import *Rift Component* into your Unity project, the following files will be added:



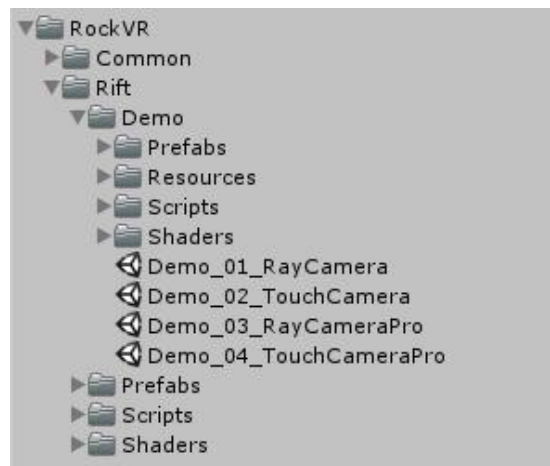
<i>RockVR/Rift/Demo</i>	Contains the scene file and all other assets for a fully functional demonstration of Video Capture with Rift interaction.
<i>RockVR/Rift/Prefabs</i>	Contains useful prefabs can be dragged and dropped in to your scene.
<i>RockVR/Rift/Scripts</i>	Contains the core Rift interaction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

This guide covers integrating VR Capture to your own Unity project with Oculus Rift 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](mailto:dev@rockvr.com). We will respond to you as quickly as possible.

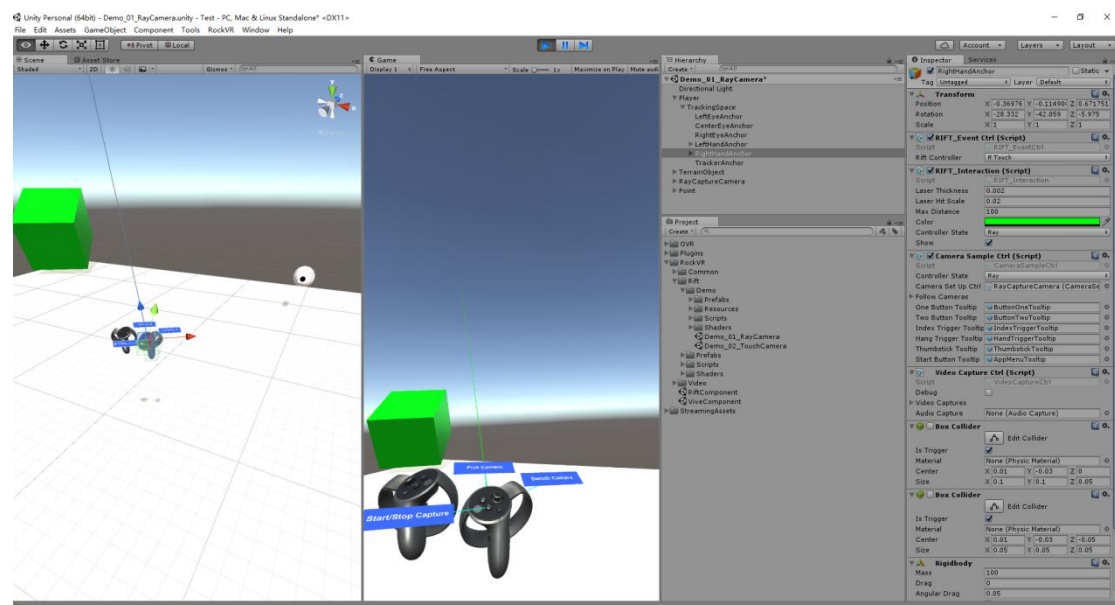
## 2. Demo Quick Start

VR Capture [RiftComponent](#) come with several demos to help you understand functionality quickly. Start by importing [Oculus](#) plugin then importing all [RiftComponent](#) package assets included demo scenes files.



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

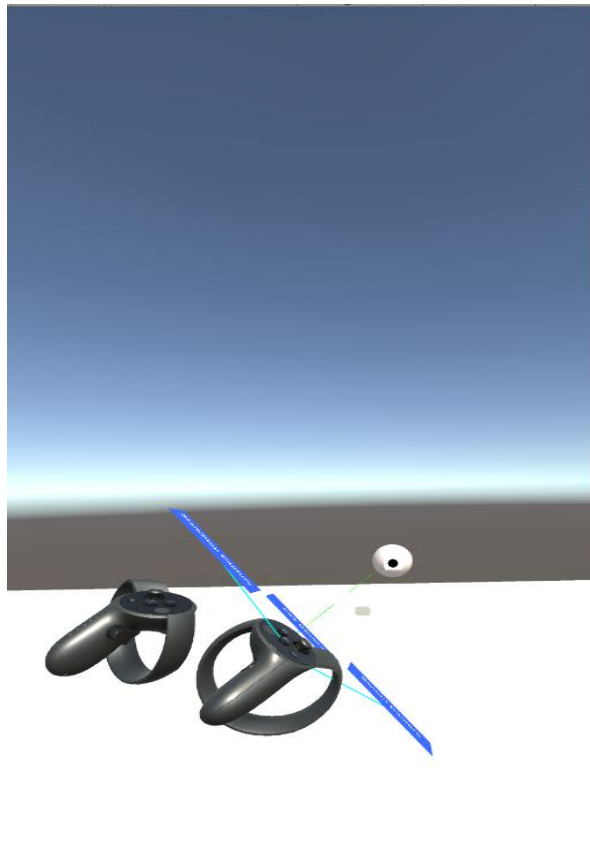
### 2.1 Laser Ray Controller Demo



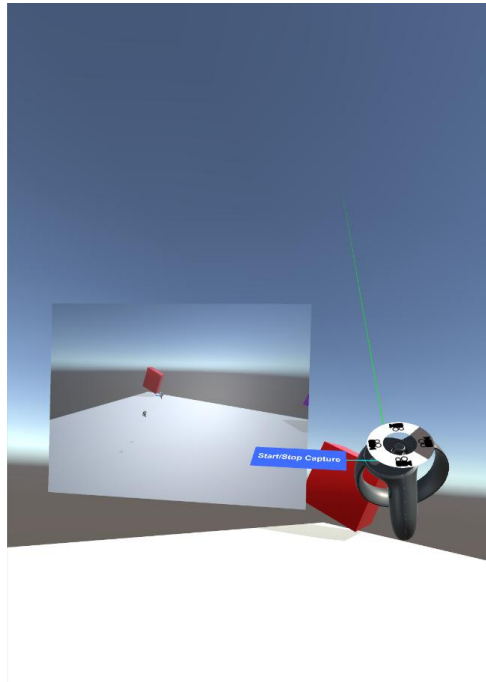
**Step 1:** Open the demo scene located in [/RockVR/Rift/Demo/Demo\\_01\\_RayCamera](#) :



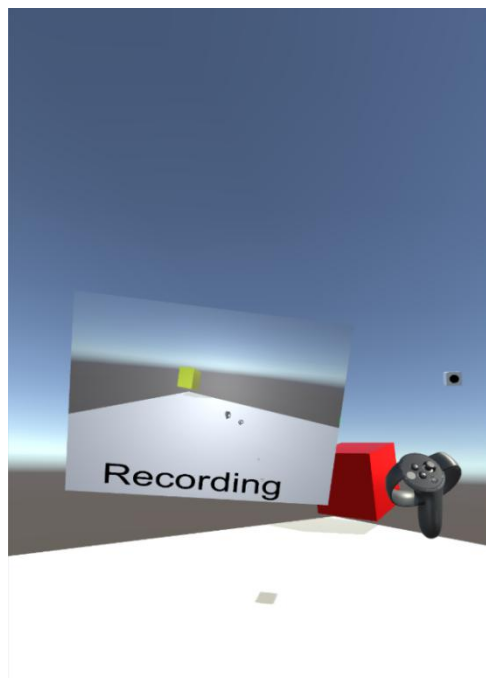
**Step 2:** Interact with recording camera. You need using laser ray to select the camera:



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

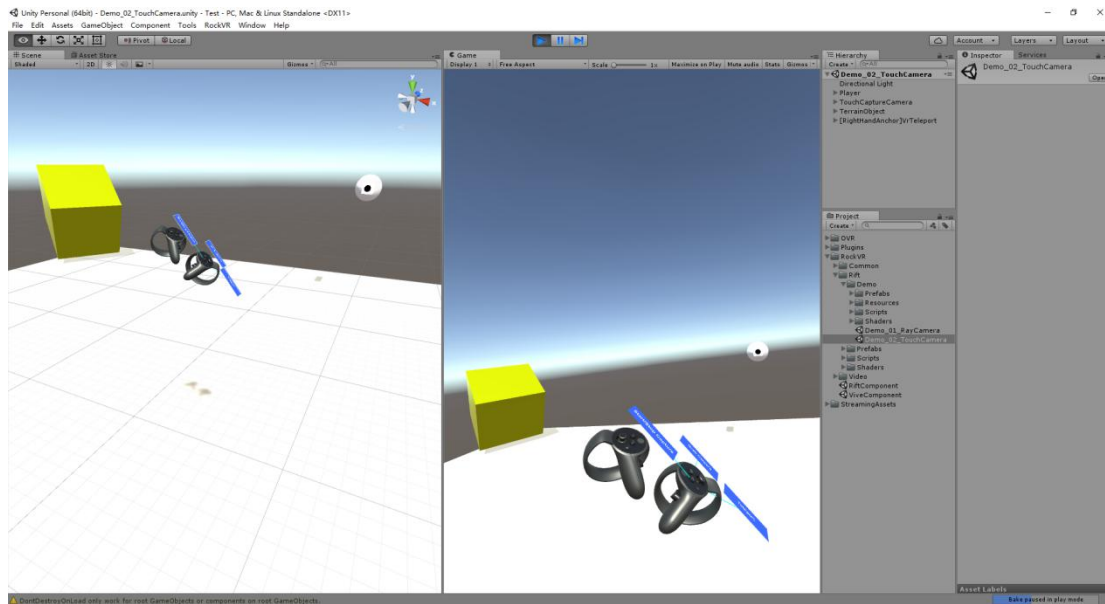


**Step 4:** After your perfect camera position set, press trigger to start video recording session:



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.

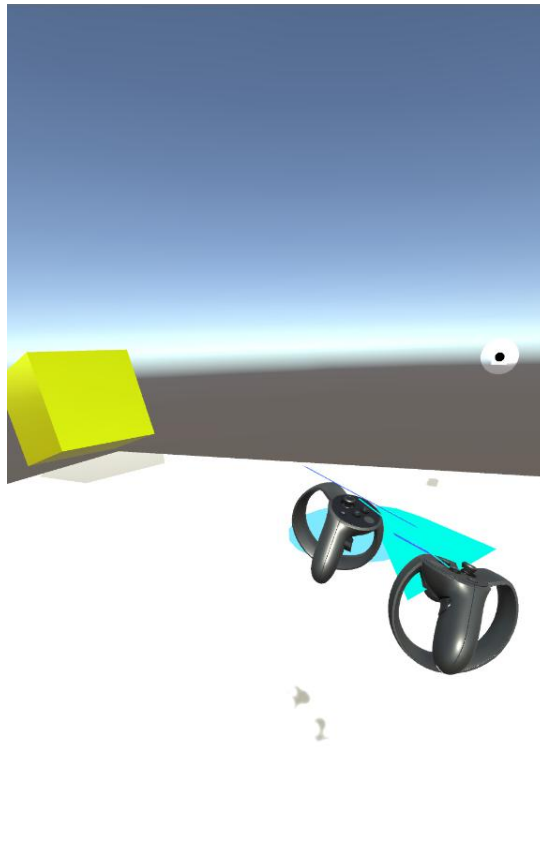
## 2.1 Touch Controller Demo



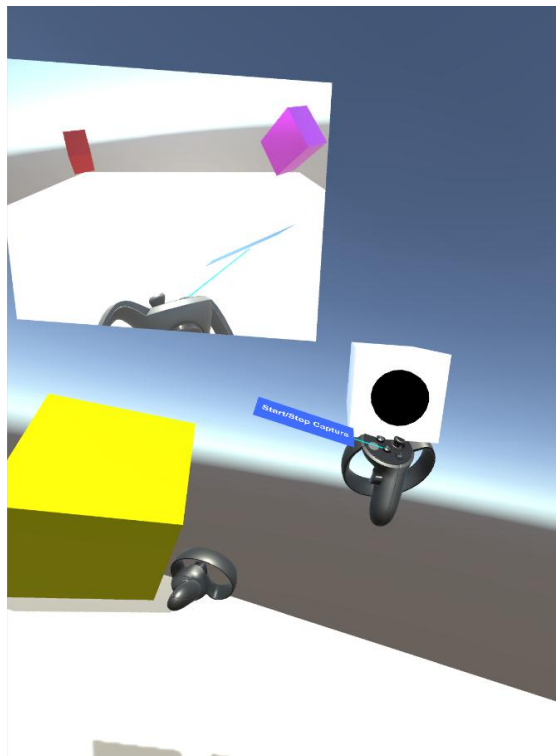
**Step 1:** Open the demo scene located in */RockVR/Rift/Demo/Demo\_02\_TouchCamera* :



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

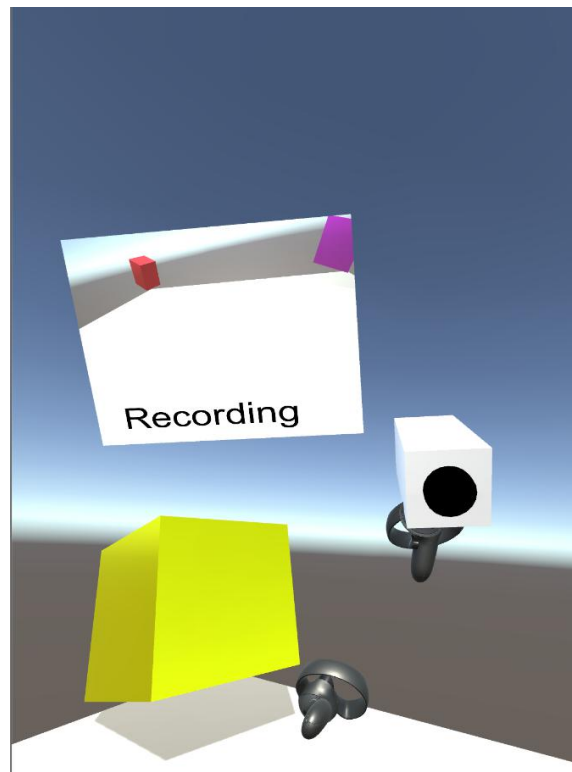


**Step 3:** Grab the camera and adjust to desired position and angle:





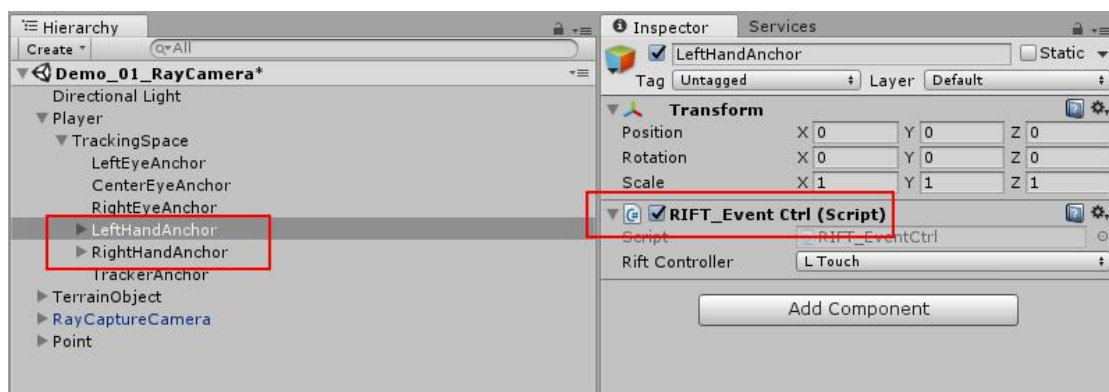
**Step 4:** Press the trigger to start video recording session, also you can grab camera and move it around while recording:



### 3. Integration Guide

[RIFT\\_Intertion](#) module implemented the core function of VR interaction control.

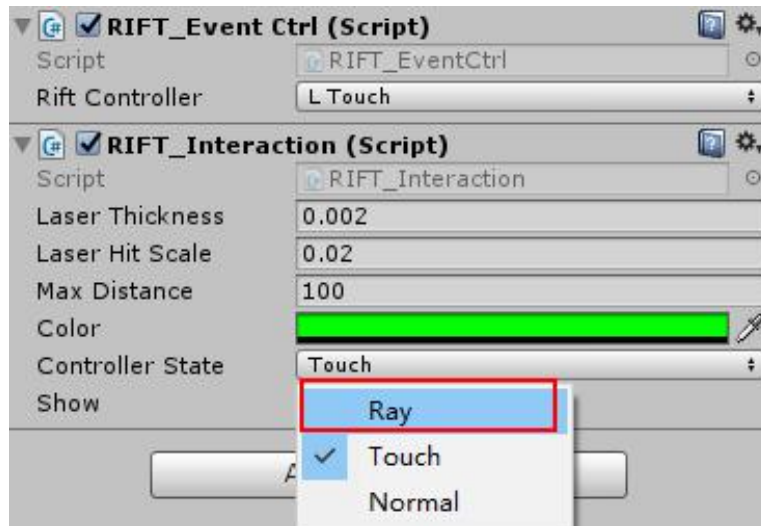
**Step 1:** Attach [RIFT\\_EventCtrl](#) script to the rift devices controller in the scene you want to control.



**Note:** [RIFT\\_EventCtrl](#) is based on the [RIFT\\_EventDelegate](#) and [Oculus](#) plugin, it is an event script to control the rift handle devices.

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

*RIFT\_Interaction* - Set the interaction patterns, choose Ray or Touch interaction type.



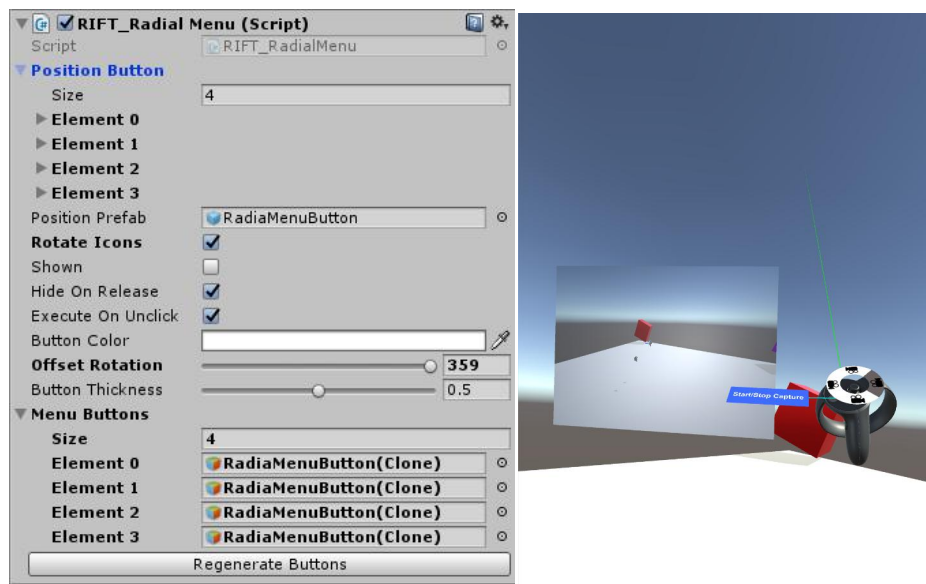
*RIFT\_Pointer* - This script encapsulates system event to take response for user interaction.

*RIFT\_UIEventInputModule* is designed to work as you would expect how a Rift controller input to work. Including button presses, dragging, and touch events.

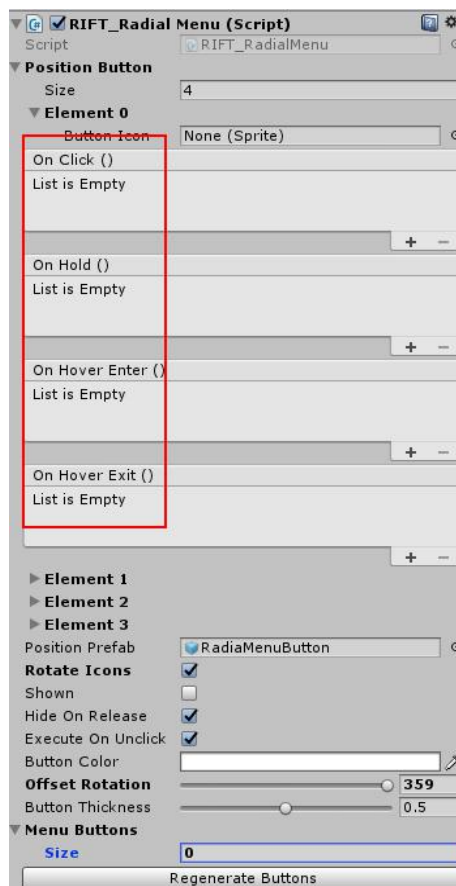
*RIFT\_Teleport* - Implemented functionality of teleport in VR scene. Enable *SeachDownPoint* and *ConfirmDownPoint* function to implement teleport.

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

*RIFT\_RadialMenu* - Set radial menu attribute, and binding keys to event listener.



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



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

Create *RIFT\_EventCtrl* object.

```
private RIFT_EventCtrl eventCtrl;
```

Register to create a delegated events.

```
void OnEnable()
{
    if (eventCtrl != null)
    {
        eventCtrl.eventDelegate.OnPressButtonPrimaryHandTrigger += OnPressButtonPrimaryHandTrigger;
        eventCtrl.eventDelegate.OnPressButtonPrimaryHandTriggerUp += OnPressButtonPrimaryHandTriggerUp;
        eventCtrl.eventDelegate.OnPressButtonOneDown += OnPressButtonOneDown;
        eventCtrl.eventDelegate.OnPressButtonPrimaryIndexTrigger += OnPressButtonPrimaryIndexTrigger;
        eventCtrl.eventDelegate.OnPressButtonPrimaryIndexTriggerUp += OnPressButtonPrimaryIndexTriggerUp;
        eventCtrl.eventDelegate.OnTouchPrimaryThumbstick += OnTouchPrimaryThumbstick;
        eventCtrl.eventDelegate.OnTouchPrimaryThumbstickUp += OnTouchPrimaryThumbstickUp;
        eventCtrl.eventDelegate.OnPressPrimaryThumbstick += OnPressPrimaryThumbstick;
        eventCtrl.eventDelegate.OnPressPrimaryThumbstickDown += OnPressPrimaryThumbstickDown;
        eventCtrl.eventDelegate.OnPressPrimaryThumbstickUp += OnPressPrimaryThumbstickUp;
    }
}
```

## 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](mailto:dev@rockvr.com) and we will provide more help to let you share your awesome game more efficient.