

# Virtual Reality in Unity 3D

Linya Ruan

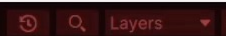
SRH Fachschulen

Stuttgart, 06. Juli, 2023



# Zeitplan für den Kurs

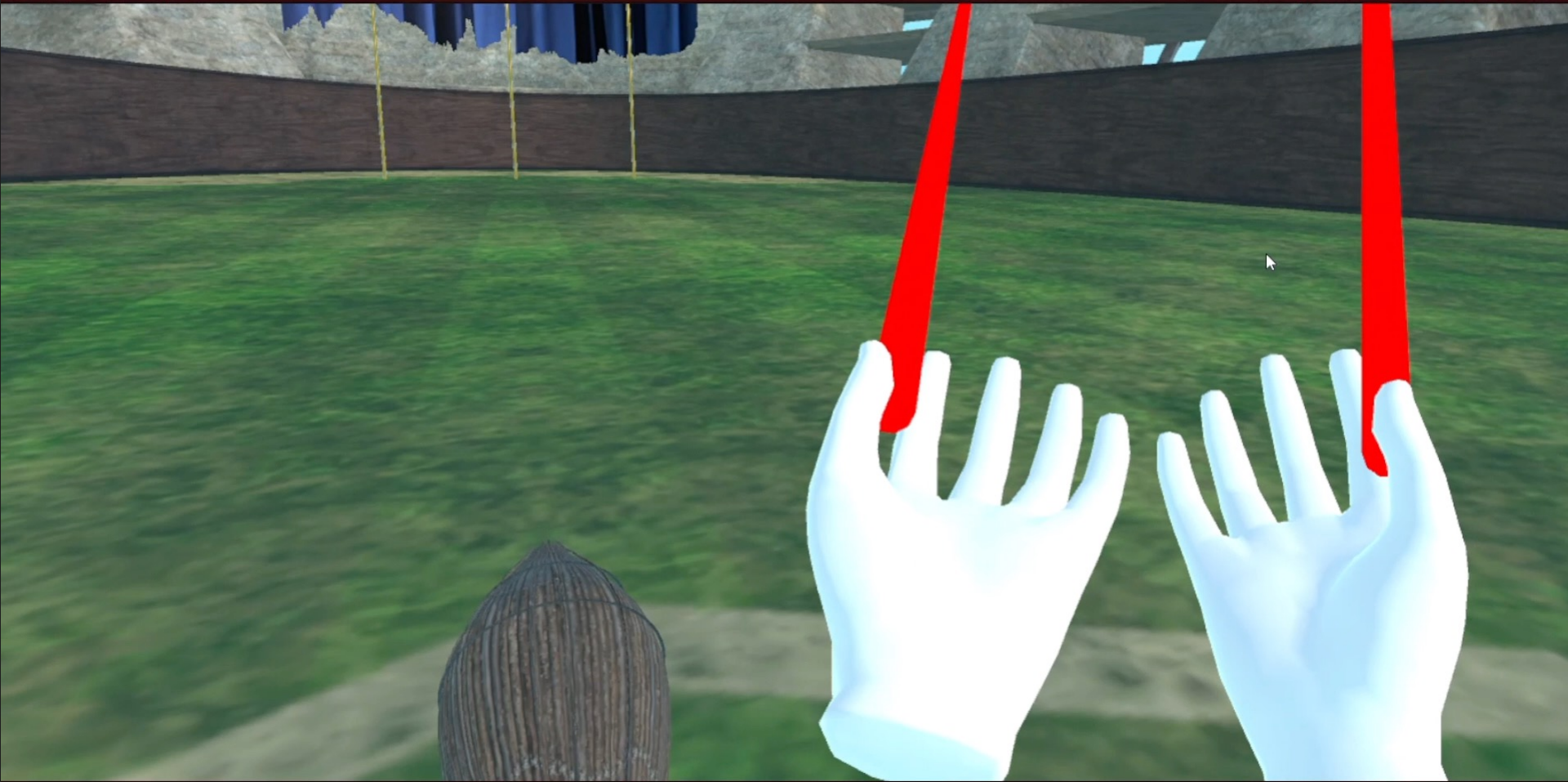
<b>06-07-2023</b>	<ul style="list-style-type: none"> <li>• Verbindung von Oculus Quest mit Unity 3D</li> <li>• VR Hand Animation</li> <li>• Programmierung : C# (Grundkennetnisse )</li> </ul>
-------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Display 1 Free Aspect

Scale 1x

Left Eye Play Focused Stat



# Install Input System package



The screenshot shows the Unity Package Manager interface. On the left, a list of packages is displayed, sorted by name. The 'Input System' package is highlighted, showing it is installed as a dependency at version 1.4.4. The right pane provides details for the 'Input System' package, including its version (1.4.4), release date (November 03, 2022), registry (Unity), and a description of its features. A 'Remove' button is visible at the bottom right.

Package Name	Version	Status
Code Coverage	1.2.2	✓
Collections	1.2.4	
Core RP Library	12.1.8	
Deployment	1.1.0	
Economy	3.1.2	
Editor Coroutines	1.0.0	✓
FBX Exporter	4.1.3	
High Definition RP	12.1.8	
In App Purchasing	4.9.3	
Input System	1.4.4	✓
Installed as dependency	1.4.4	R
iOS 14 Advertising Support	1.0.0	
JetBrains Rider Editor	3.0.16	ⓘ
Live Capture	1.1.1	
Lobby	1.0.3	
Localization	1.3.2	
Magic Leap XR Plugin	6.4.1	
Mathematics	1.2.6	
ML Agents	2.0.1	

**Input System** Release

Unity Technologies

**Version 1.4.4 - November 03, 2022**

Registry Unity

*com.unity.inputsystem*

[View documentation](#) • [View changelog](#) • [View licenses](#)

A new input system which can be used as a more extensible and customizable alternative to Unity's classic input system in UnityEngine.Input.

**Samples**

Last update Jul 5, 19:17 ↻

Remove

# Quidditch VR

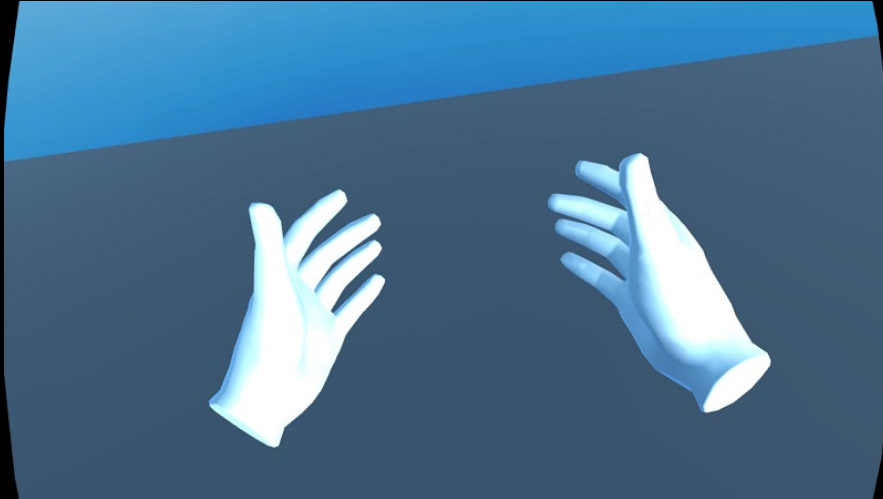


## ➤ Oculus Integration

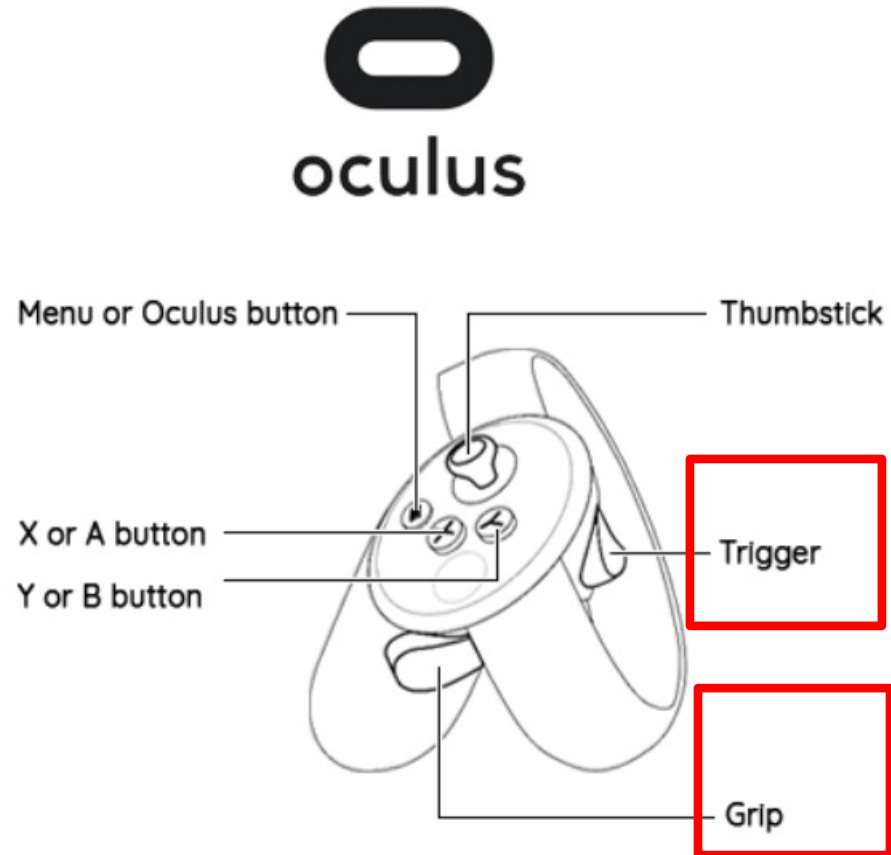
<https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022>

## ➤ Oculus CustomHands

<https://developer.oculus.com/documentation/unity/unity-sf-customhands/>

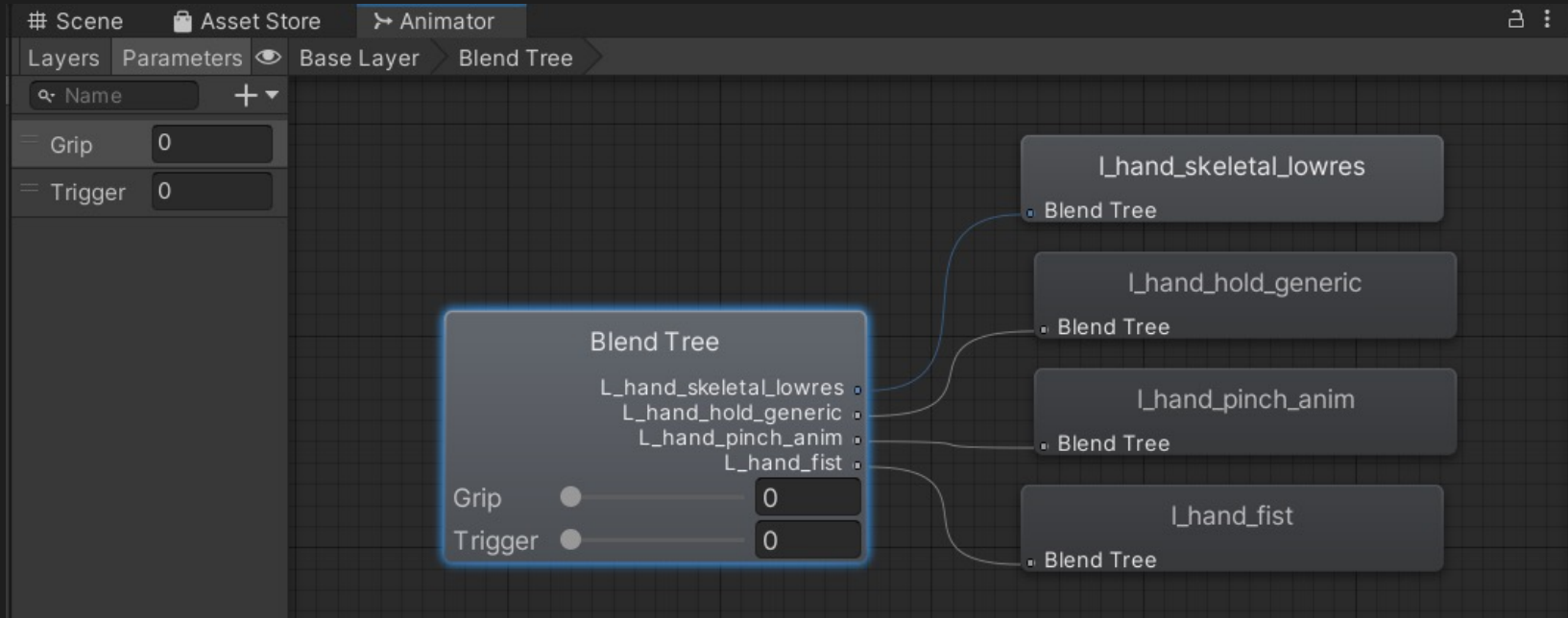


# oculus Controller





# Blend Tree (CustomHandLeft and CustomHandRight)



# Hand Animation



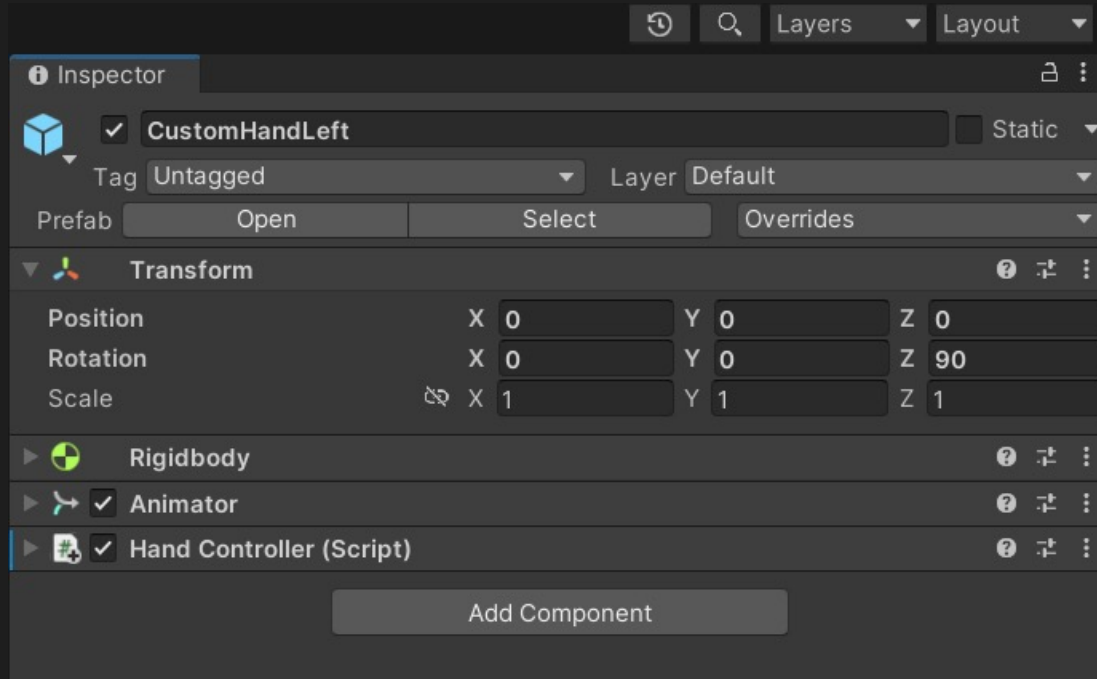
- Add XR Origin (Deviced-based -> XR Origin)
- Add CustomHandLeft and CustomHandRight
- Animator
  - Asset -> Samples -> XR Interaction Toolkit -> 2.0.4 -> Starter Assets -> XRI Default Input Actions
  - **XRI LeftHand** -> Add Actions (Grip) -> change Action Type(Value) -> change Control Type(Axis)
  - No Binding -> path(XR Controller) -> XR Controller (LeftHand)-> Optional Controls->grip
  - **XRI LeftHand** -> Add Actions (Trigger) -> change Action Type(Value) -> change Control Type(Axis)
  - No Binding -> path(XR Controller) -> XR Controller (LeftHand)-> Optional Controls->trigger (**same XRI Right Hand**)
  - **Save Asset**



# Hand Animation

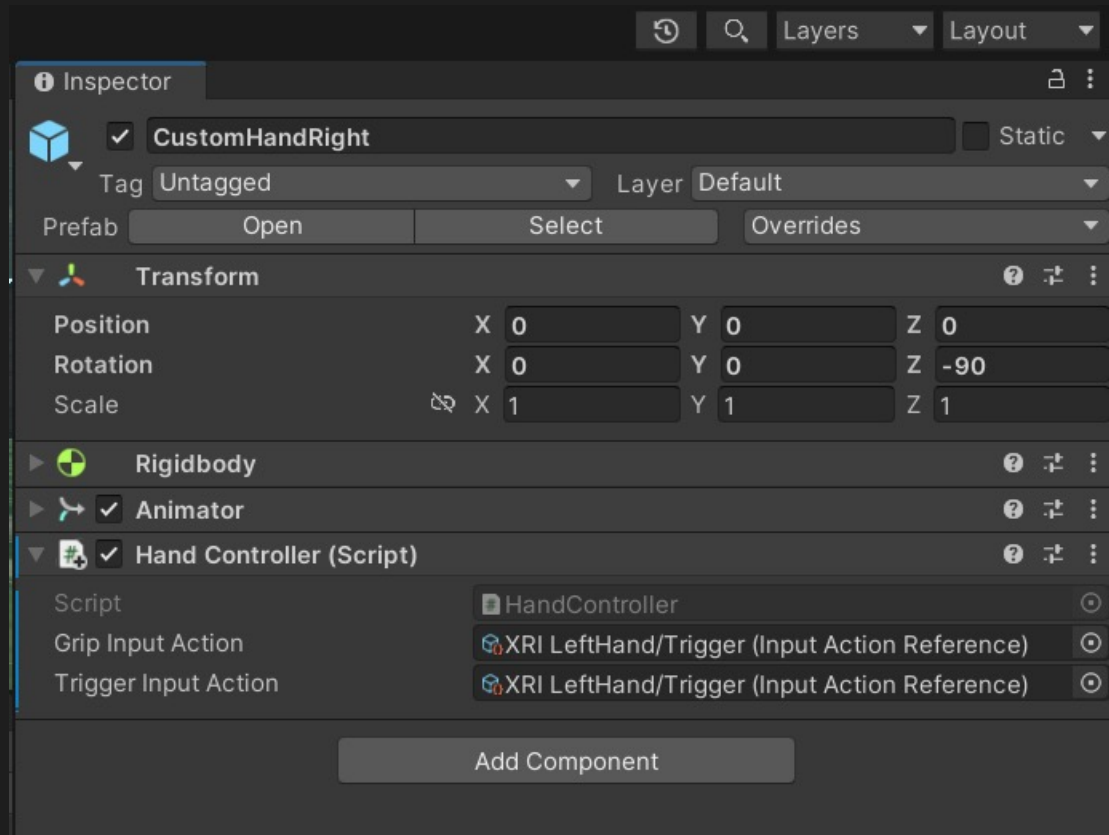


- Create C# Script (HandController)
- Add C# Script to CustomHandLeft and CustomHandRight



API	Explain
<b>SerializeReference( )</b>	UnityEngine.Object fields, by reference: [SerializeReference] attribute
<b>GetComponent ( )</b>	To find components attached to a particular GameObject myResults = GetComponent<ComponentType>()
<b>OnEnable ( )</b>	This function is called when the object becomes enabled and active.
<b>OnDisable ( )</b>	This function is called when the behaviour becomes disabled.

# Hand Animation



```
HandController.cs* X
Assembly-CSharp HandController

4 using UnityEngine;
5 using UnityEngine.InputSystem;
   Unity-Skript | 0 Verweise
6 public class HandController : MonoBehaviour
7 { // Add GripInputAction and TriggerInputAction
8     [SerializeField] InputActionReference GripInputAction;
9     [SerializeField] InputActionReference TriggerInputAction;
10    Animator HandAnimator;
   Unity-Nachricht | 0 Verweise
11    private void Awake()
12    {
13        // Get HandAnimation
14        HandAnimator = GetComponent<Animator>();
15    }
16
   Unity-Nachricht | 0 Verweise
17    private void OnEnable()
18    {
19        GripInputAction.action.performed += GripPressed;
20        TriggerInputAction.action.performed += TriggerPressed;
21    }
   2 Verweise
22    private void GripPressed(InputAction.CallbackContext obj)
23    {
24        HandAnimator.SetFloat("Grip", obj.ReadValue<float>());
25    }
26
   2 Verweise
27    private void TriggerPressed(InputAction.CallbackContext obj)
28    {
29        HandAnimator.SetFloat("Trigger", obj.ReadValue<float>());
30    }
31
   Unity-Nachricht | 0 Verweise
32    private void OnDisable()
33    {
34        GripInputAction.action.performed -= GripPressed;
35        TriggerInputAction.action.performed -= TriggerPressed;
36    }
```

# Hand Animation



- Add Input Action Manager to XR Origin
- Add XRI Default Input Actions (Input Action Asset)

## Finish Part 2

