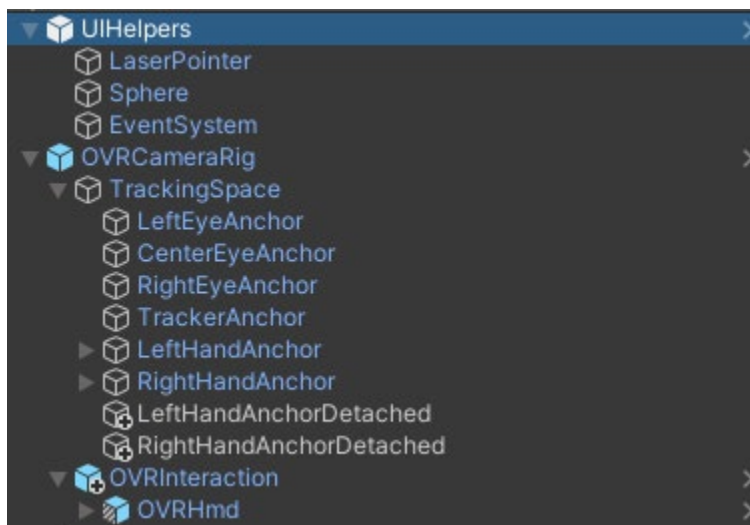
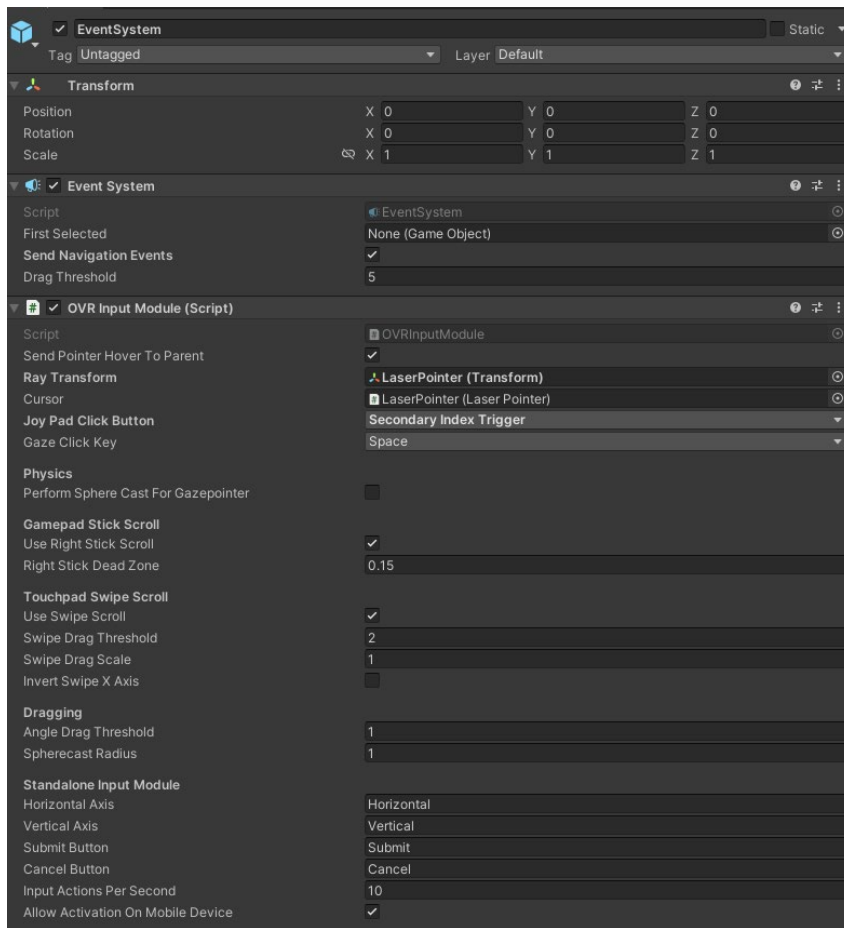


코드 설명서

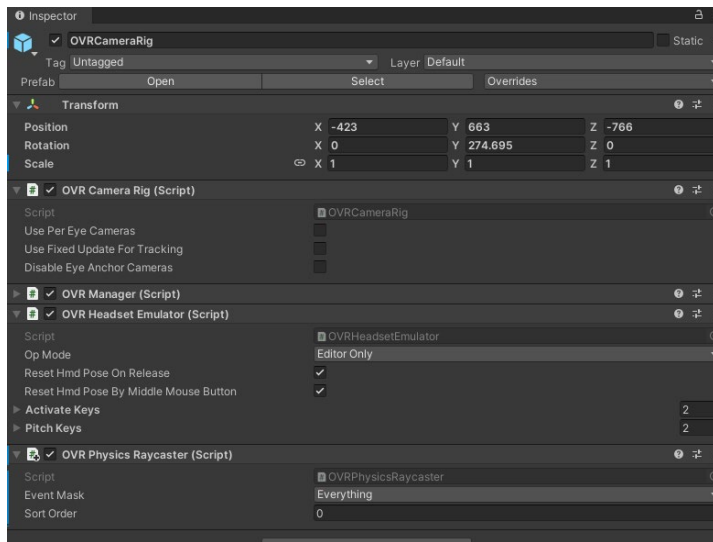
몸체의 움직임을 조정하기 위해서는 레버에 애니메이션/동작 스크립트를 활성화, 비활성화하는 코드를 추가하고, 활성화 조건을 위해 Raycast 를 적용해야 합니다.



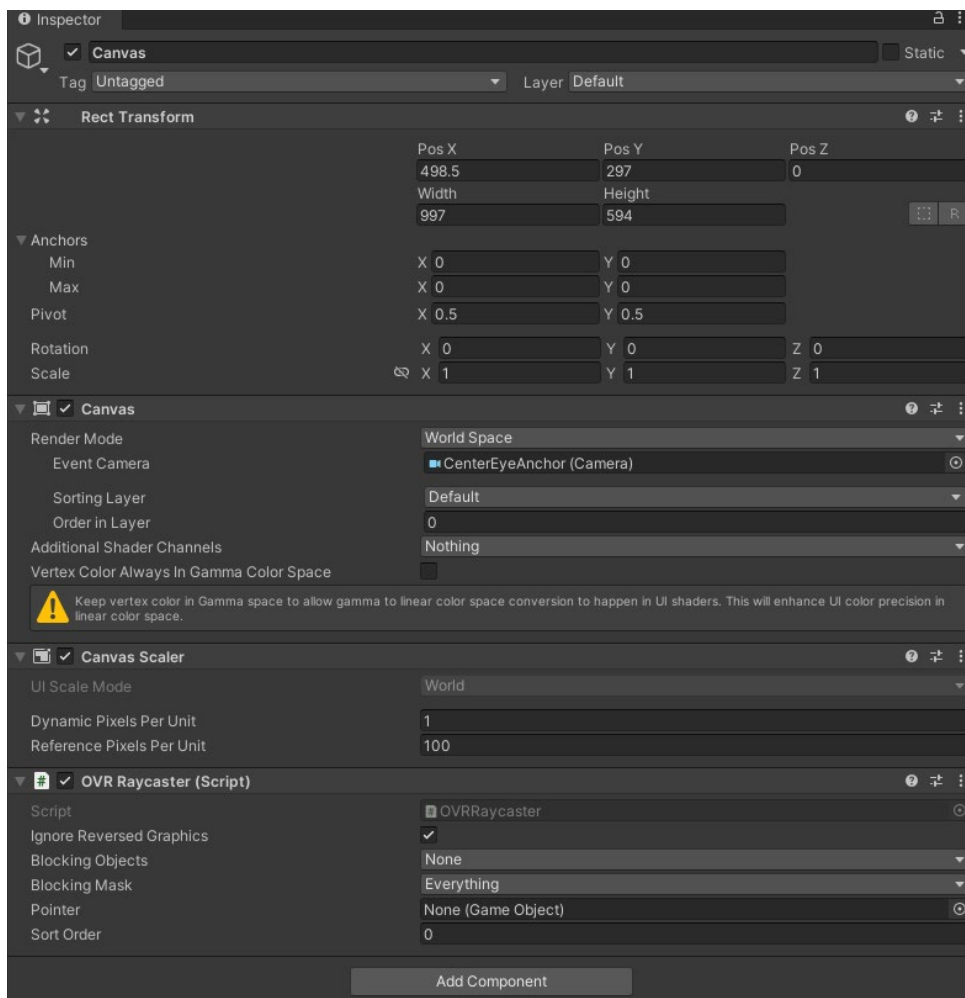
우선적으로 클릭 구현을 위해 Raycast 물리를 적용해야 합니다. 위와 같이 OVR 카메라를 세팅하고, UIHelpers 에셋을 불러옵니다.



UIHelpers 항목의 EventSystem 에서 Standard Input Module 을 비활성화하고 위와 같이 OVR Input 모듈을 세팅합니다.

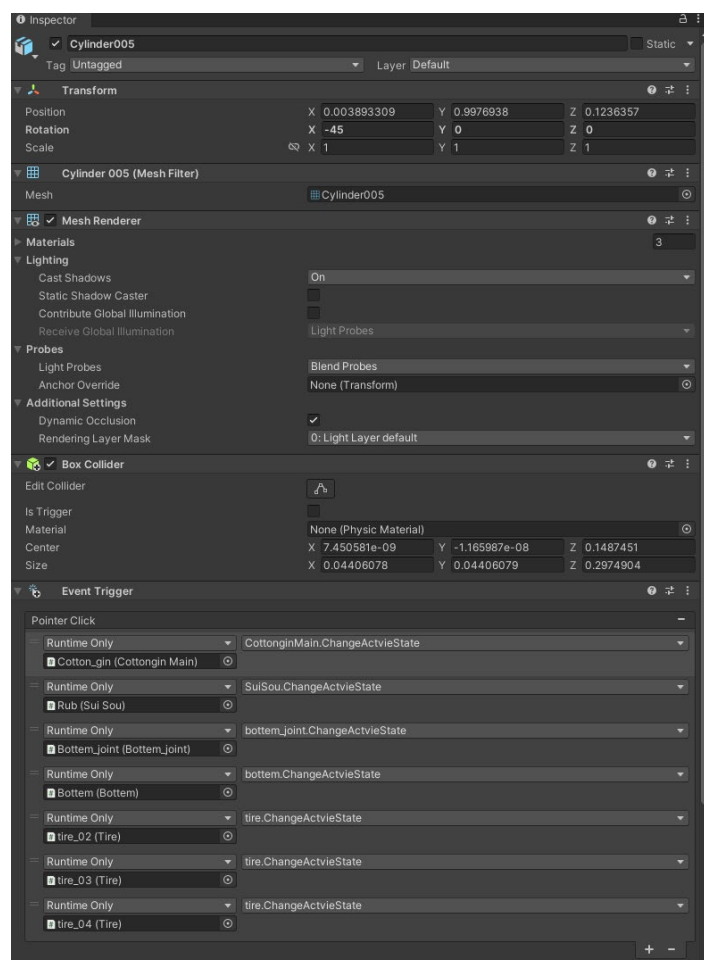


위와 같이 OVRCameraRig 에 OVR Physics Raycaster 스크립트를 추가합니다.



위와 같이 임의로 Canvas 를 생성한 다음, Hierachy 에서 조면기 본체를 Canvas 하위에 위치시킵니다.

이후 Canvas 에 OVR Raycaster 스크립트를 적용하면 정상적으로 오쿨러스 VR 의 컨트롤러 버튼으로 상호작용이 가능합니다.



위와 같이 실린더 5(레버의 손잡이 부분)에 Box Colider 를 추가하여 클릭 상호작용이 가능하게 합니다. 콜라이더의 크기를 키워 VR 컨트롤러의 판정 문제를 어느 정도 편리하게 해줄 수 있습니다.(생각보다 레버 누르기가 작게 보여서 쉽지 않음)

다음으로 이벤트를 트리거를 추가하고 “Pointer Click”에 움직임을 구현한 스크립트를 적용합니다. 레버를 누르면 활성화 상태를 바꿔줘야 하고, 각각 이것을 구현한 함수의 이름은 “ChangeActiveState”로 고정했습니다.

```

5  using System.Collections;
6  using System.Collections.Generic;
7  using UnityEngine;
8
9  public class CottonginMain : MonoBehaviour
10 {
11     public Animator animator;
12     // Start is called before the first frame update
13     void Start()
14     {
15         animator = GetComponent<Animator>();
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21
22     }
23     public void ChangeActiveState ()
24     {
25         print("dddd2");
26         if (animator.GetBool("UpperActive"))
27         {
28             animator.SetBool("UpperActive", false);
29             print("dddd");
30         }
31         else
32         {
33             animator.SetBool("UpperActive", true);
34             print("dddd1");
35         }
36     }
37
38 }

```

CottonginMain.cs

코튼진(조면기) 본체에 적용되는 스크립트로, 애니메이터 animator 를 불러오고
ChangeActivateState 함수를 통해 조면기 상단의 애니메이션 활성화 상태를 조정합니다.

애니메이터의 UpperActive 는 불 변수인데, 불 변수가 변할때마다 트랜지션이 적용되어
조면기 상단이 움직였다가/멈췄다가를 반복하게 됩니다.

```

4
5 using System.Collections;
6 using System.Collections.Generic;
7 using UnityEngine;
8
9 public class tire : MonoBehaviour
10 {
11     public float rotationSpeed = 2.0f; // 회전 속도
12     public bool alive = false;
13     public float suisoutime = 0.0f;
14
15
16 void Update()
17 {
18     // Z축 방향으로 회전
19     if(alive)
20     {
21         transform.Rotate(0, 0, rotationSpeed);
22     }
23
24     //if (suisoutime > 60f)
25     //{
26         // suisoutime = 0.0f;
27     //}
28
29 }
30
31 public void ChangeActiveState()
32 {
33     print("Z축 회전");
34     if (alive)
35     {
36         alive = false;
37         print("SuisousOnOff");
38     }
39     else
40     {
41         alive = true;
42         print("SuisousOnOn");
43     }
44 }
45 }

```

Tire.cs

타이어(조면기 상단을 제외한 나머지 바퀴)에 적용되는 스크립트로, 움직임이 애니메이션 대신 스크립트 형태로 구현됩니다. 이 경우 각각 ChangeActivateState 로 변화하는 alive 값을 받아와 동작을 활성화할지, 아니면 멈추게 할지 정하는 식으로 구현하였습니다.