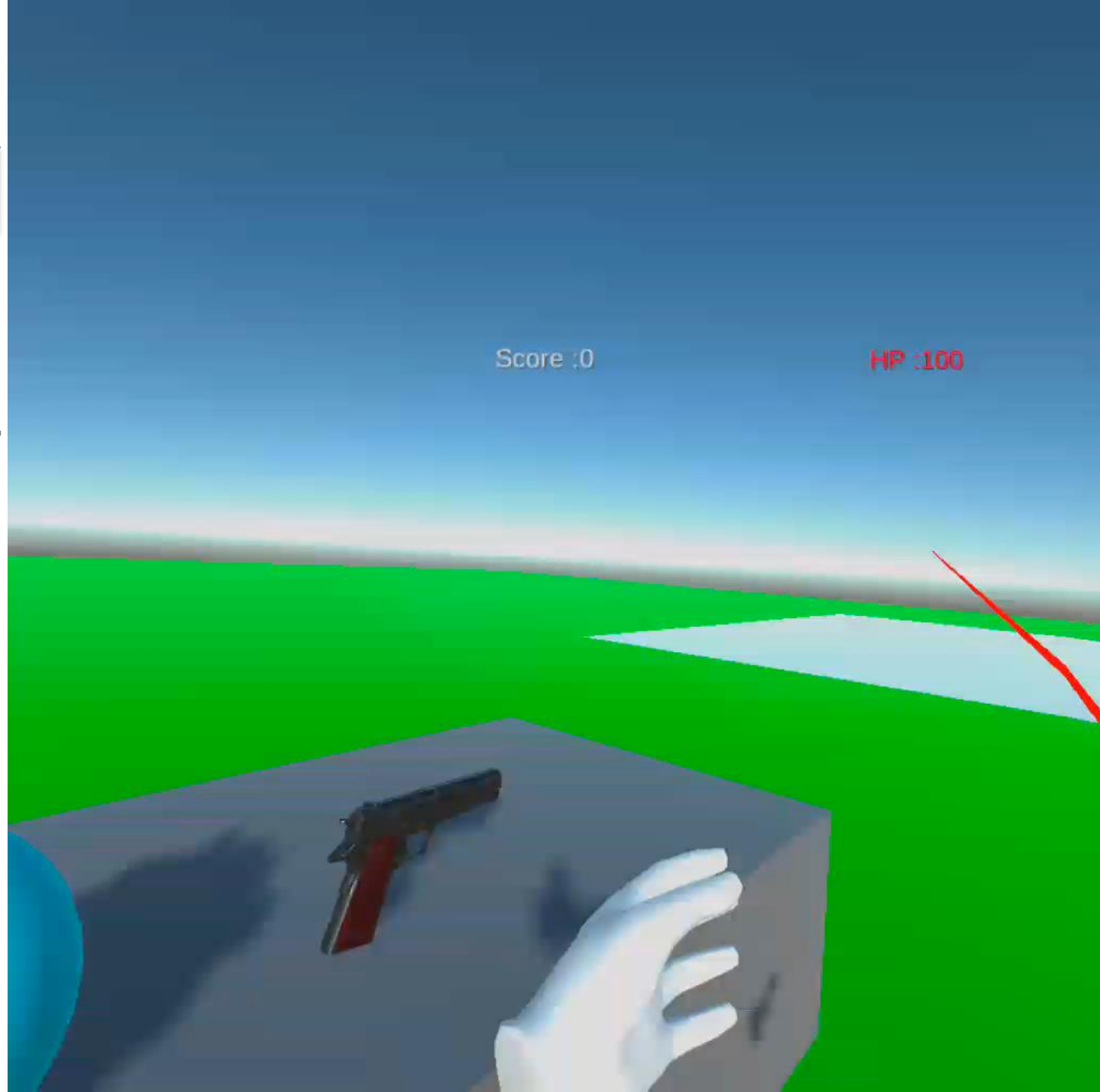


총싸움 전투 마지막

1. 이준

전투 시

- 어제 만든 게임 시스템 점검!
- 문제점 : 왼손 오른손 충돌
- 왼손으로 총을 잡고 있을때, 오른손 트리거로 총알 발사!
- 왼손으로 총을 잡고 있을때, 오른손으로 텔레포트시 그림 해제 되는 버그 발생!

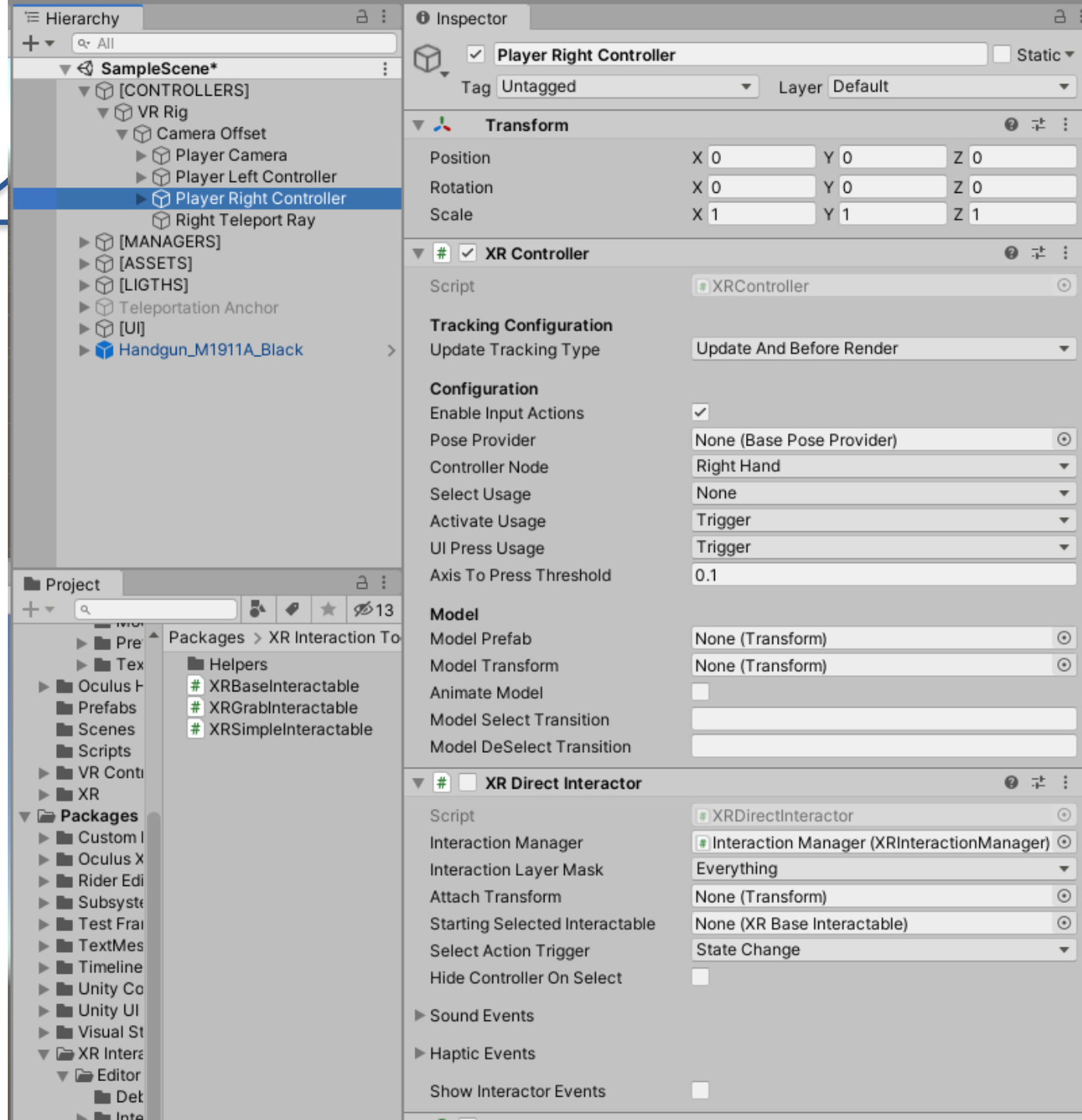


전투 시스템 만들기!

- 개선 알고리즘
- 잡고 있는 손을 알아야함
- 오른손 : 텔레포트 레이로 이동만 잡기는 안되도록 하기!
- 왼손 : 왼손으로 총을 잡고 슈팅

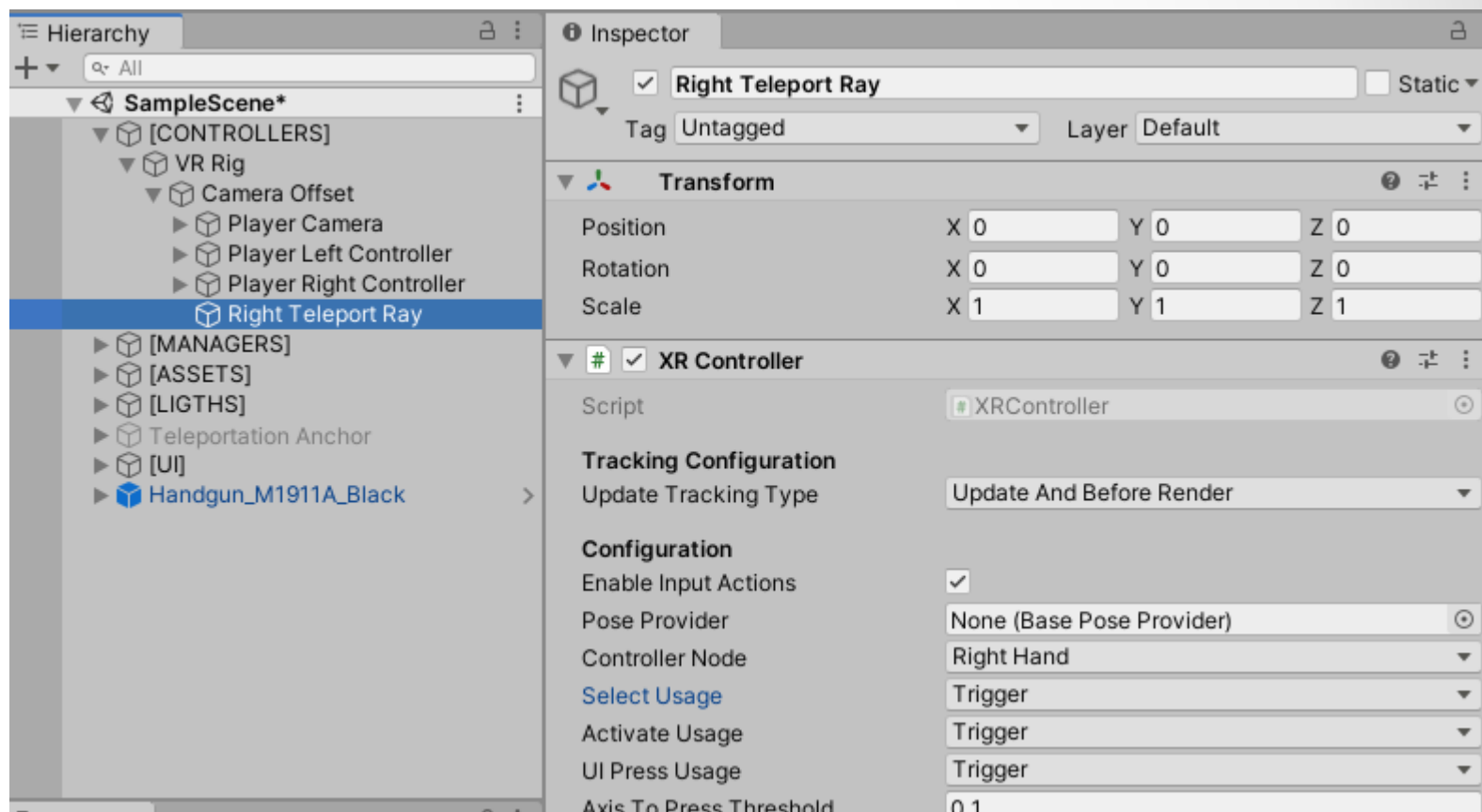
전투 시스템

- 오른손으로 잡기 해제처리
- XRController 에서 Select Usage를 None으로 변경
- XR Direct Interactor 비활성화



전투 시스템 만들기!

- 오른쪽 텔레포트 레이는 트리거로만 이동



전투

- CustomController 수정
- 컨트롤러 및 총이 어떤 손으로 잡고 있는지를 알기 위해서 enum 타입의 HandState를 추가함!
- 그리고 이를 변수로 가지는 HandGun을 가져옴!

```
XRGrabInteractable.cs CustomController.cs SimpleShoot.cs
Assembly-CSharp CustomController characteristics

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.XR;
5 using UnityEngine.XR.Interaction.Toolkit.UI;
6
7 public enum HandState { NONE = 0, RIGHT, LEFT };
8
9 public class CustomController : MonoBehaviour
10 {
11     public InputDeviceCharacteristics characteristics;
12     [SerializeField]
13     private List<GameObject> controllerModels;
14     private GameObject controllerInstance;
15     private InputDevice availableDevice;
16
17     public bool renderController ; // Hand와 Controller 사이를 변경할 변수
18     public GameObject handModel; // 핸드 모델
19     private GameObject handInstance; // 핸드 인스턴스
20
21     private Animator handModelAnimator; // 핸드 모델 애니메이션 변수
22
23     public GameObject HandGun;
24
25     bool triggerButton;
26
27
28     public HandState currentHand;
29
```

전

- CustomController 수정
- TryInitialize 함수에서 Left 및 Right 에 따라서 컨트롤의 손을 각각 설정

```
40 void TryInitialize()  
41 {  
42     List<InputDevice> devices = new List<InputDevice>();  
43     InputDevices.GetDevicesWithCharacteristics(characteristics, devices);  
44     foreach (var device in devices)  
45     {  
46         Debug.Log($"Available Device Name: {device.name}, Characteristic: { device.characteristics}")  
47     }  
48     if (devices.Count > 0)  
49     {  
50         availableDevice = devices[0];  
51         GameObject currentControllerModel;  
52         if(availableDevice.name.Contains("Left"))  
53         {  
54             currentControllerModel = controllerModels[1];  
55             currentHand = HandState.LEFT;  
56         }  
57         else if(availableDevice.name.Contains("Right"))  
58         {  
59             currentControllerModel = controllerModels[2];  
60             currentHand = HandState.RIGHT;  
61         }  
62         else  
63         {  
64             currentControllerModel = null;  
65             currentHand = HandState.NONE;  
66         }  
67         if (currentControllerModel)  
68         {  
69             controllerInstance = Instantiate(currentControllerModel, transform);  
70         }  
71         else  
72         {  
73             Debug.LogError("Didn't get suitable controller model");  
74             controllerInstance = Instantiate(controllerModels[0], transform);  
75         }  
76         handInstance = Instantiate(handModel, transform); // 핸드 인스턴스추가  
77         handModelAnimator = handInstance.GetComponent<Animator>(); // 핸드 모델 애니메이션 설정  
78     }  
79 }
```

전투 시스템 만들기!

- SimpleShoot 클래스에서
- 현재 HandGun을 잡은 손을 표시하는 currentGrab 변수를 추가

```
7 public class SimpleShoot : MonoBehaviour
8 {
9
10     public GameObject bulletPrefab;
11     public GameObject casingPrefab;
12     public GameObject muzzleFlashPrefab;
13     public Transform barrelLocation;
14     public Transform casingExitLocation;
15
16
17     public float shotPower = 100f;
18
19     public bool isGrab = false ;
20
21     public AudioClip fireClip; //총 발사 사운드 클립
22     AudioSource fireAudio;    //핸드건에 추가한 오디오소스컴포넌트를 담을 변수
23
24     public HandState currentGrab;
25
26 }
```


전투

- CustomController 의 Update 에서 발사를 다음과 같이 수정

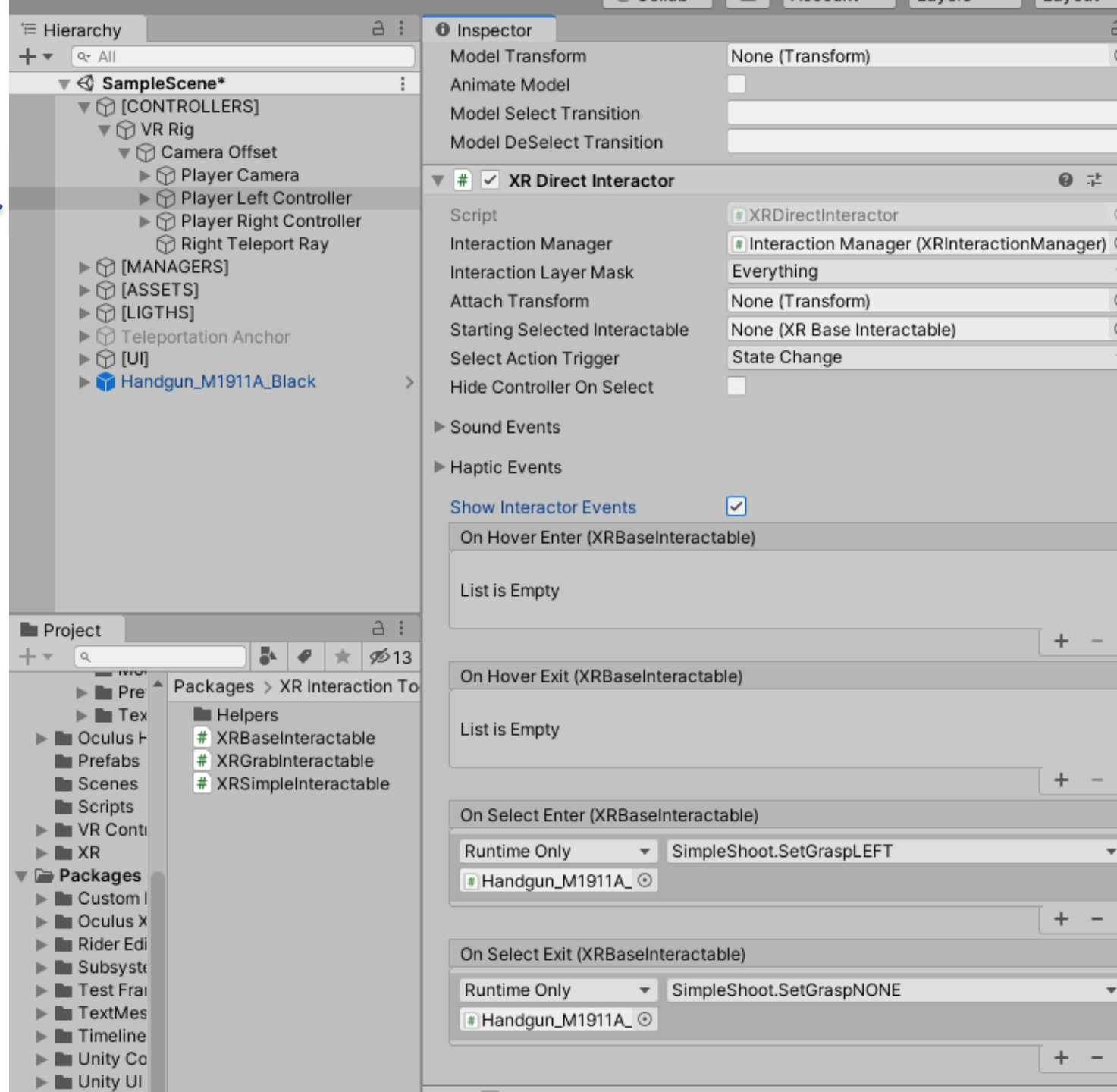
```
84 void Update()
85 {
86     if (!availableDevice.isValid)
87     {
88         TryInitialize();
89     }
90     if (renderController)
91     {
92         handInstance.SetActive(false);
93         controllerInstance.SetActive(true);
94     }
95     else
96     {
97         handInstance.SetActive(true);
98         controllerInstance.SetActive(false);
99         UpdateHandAnimation(); // 핸드 애니메이션은 여기서만 수행
100     }
101
102     if (HandGun != null)
103     {
104         bool menuButtonValue;
105         if (availableDevice.TryGetFeatureValue(CommonUsages.triggerButton, out menuButtonValue) && menuButtonValue)
106         {
107             if(triggerButton == false && currentHand == HandGun.GetComponent<SimpleShoot2>().currentGrab)
108             {
109                 HandGun.GetComponent<SimpleShoot2>().Shoot();
110                 triggerButton = true;
111             }
112         }
113         else
114         {
115             triggerButton = false;
116         }
117     }
118
119     if (FindObjectOfType<GameManager>().isGameOver)
120     {
121         bool menuButtonValue;
122         if (availableDevice.TryGetFeatureValue(CommonUsages.menuButton, out menuButtonValue) && menuButtonValue)
123         {
124             FindObjectOfType<GameManager>().RestartGame();
125         }
126     }
127 }
128
```

- SimpleShoot 에서 다음의 함수들을 선언해줘서 왼손 혹은 오른손으로 잡았는지를 알고 있으면 로직상 관리하기가 편리함

```
55 public void SetGraspState(HandState state)
56 {
57     currentGrab = state;
58 }
59
60 public void SetGraspNONE()
61 {
62     if(!GetComponent<XRGrabInteractable>().isSelected)
63         currentGrab = HandState.NONE;
64 }
65
66
67 public void SetGraspLEFT()
68 {
69     currentGrab = HandState.LEFT;
70 }
71
72 public void SetGraspRIGHT()
73 {
74     currentGrab = HandState.RIGHT;
75 }
76
```

전투

- Player Left Controller에서
- On Select Enter와 Exit를 각각
- SetGraspLEFT 와 SetGraspNONE 설정



전투 시스템 만들기!

- 적들이 범위에 들어와야 공격 하도록 변경!
- `Vector3.distance` 를 사용하여 20 안에 들어와야 공격을 하도록 변경해보자

```
31 // Update is called once per frame
32 void Update()
33 {
34     timeAffterSpawn += Time.deltaTime;
35
36     if(timeAffterSpawn >= spwanRate)
37     {
38         timeAffterSpawn = 0f;
39
40         if (!FindObjectOfType<GameManager>().isGameOver &&
41             Vector3.Distance(target.transform.position, transform.position) <= 20.0f)
42         {
43             GameObject bullet = Instantiate(bulletPrefab, transform.position, transform.rotation);
44             bullet.transform.LookAt(target);
45             fireAudio.PlayOneShot(fireClip);
46         }
47
48         spwanRate = Random.Range(spwanRateMin, spwanRateMax);
49     }
50
51 }
52
```

전투 시

넓은 지역에서 BulletSpawner와
전투 만들어 보기!

다음 영상을 보고 유사하게 혹은 더
재미있게 만들어 보기!

참고로 TeleportArea.cs 를 Plane에
적용 하였고 TeleportArea 와
TeleportAnchor (회색 부분들)은
비활성화 했음!

Plane의 크기를 10, 10, 10으로 늘림

