

# Advanced VR – Unity XR

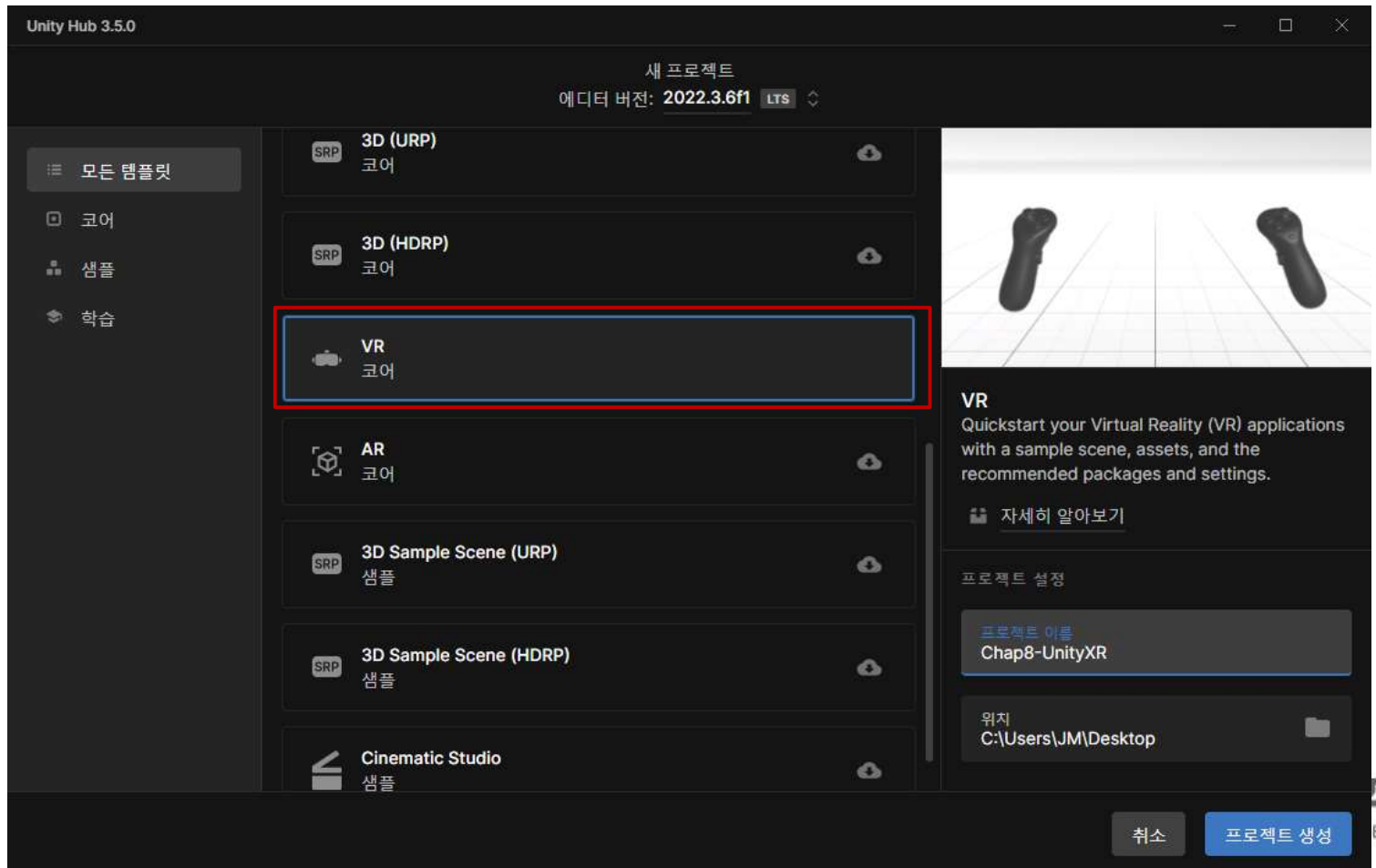
## (Chapter 6)

**Jin-Mo Kim**

*jinmo.kim@hansung.ac.kr*

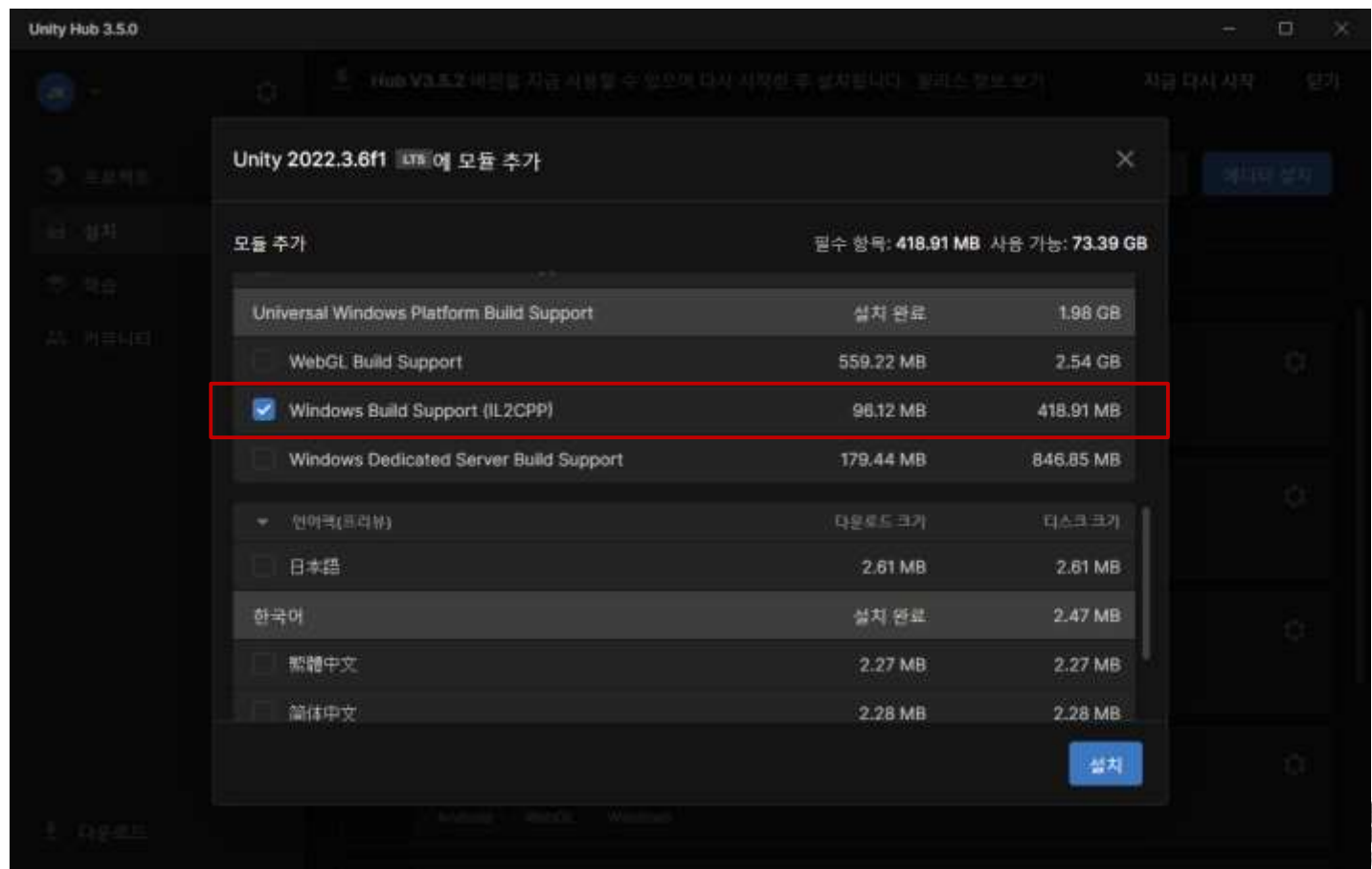
# Unity XR

- VR Template
  - VR 템플릿을 설치하여 설정



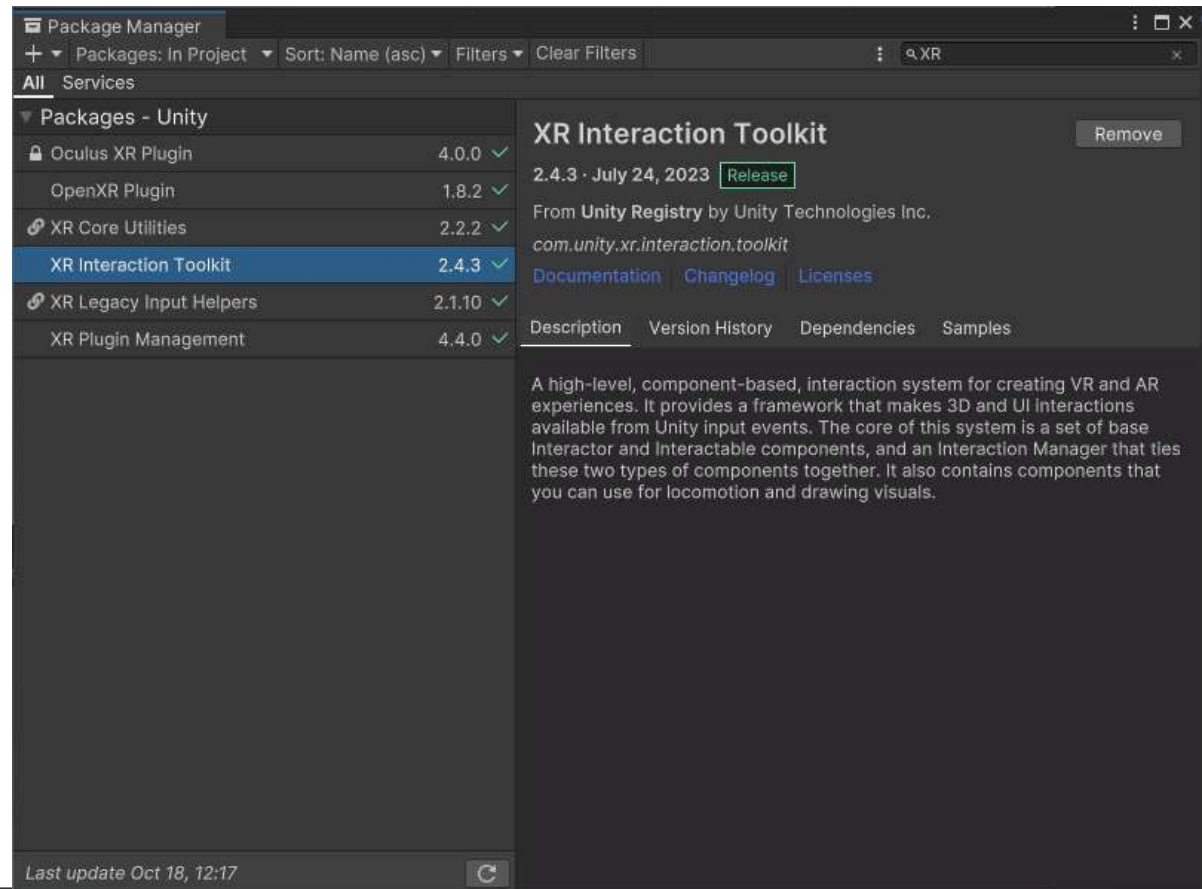
# Unity XR

- VR Template
  - 빌드를 위해서는 IL2CPP 모듈을 추가



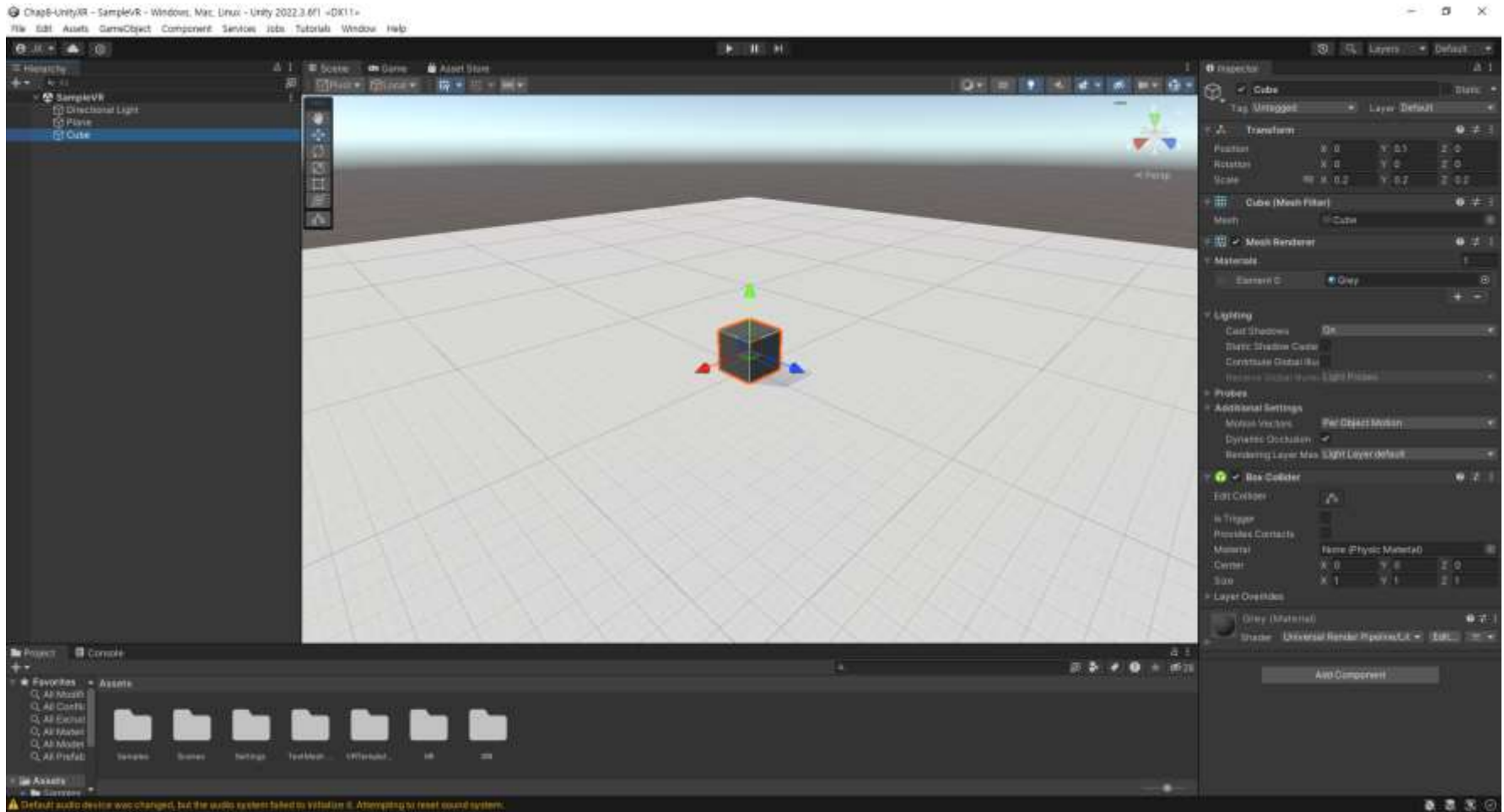
# Unity XR

- Packages
  - 필요한 패키지: XR Interaction Toolkit
    - VR Template은 기본적으로 필요한 패키지들을 모두 설치하고 있음



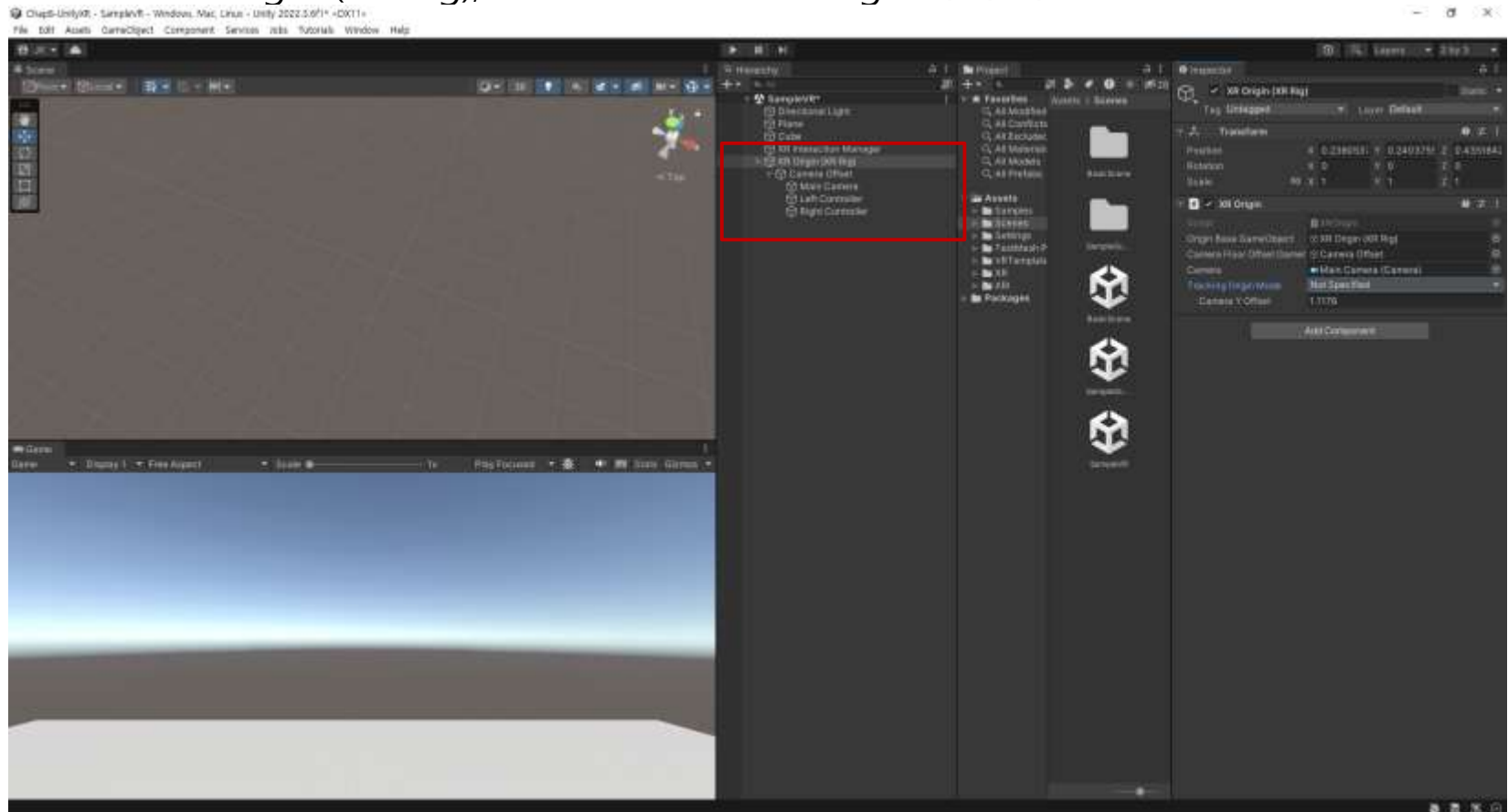
# Unity XR

- 기본 장면
  - Plane과 상호작용을 위한 Cube 생성 및 배치



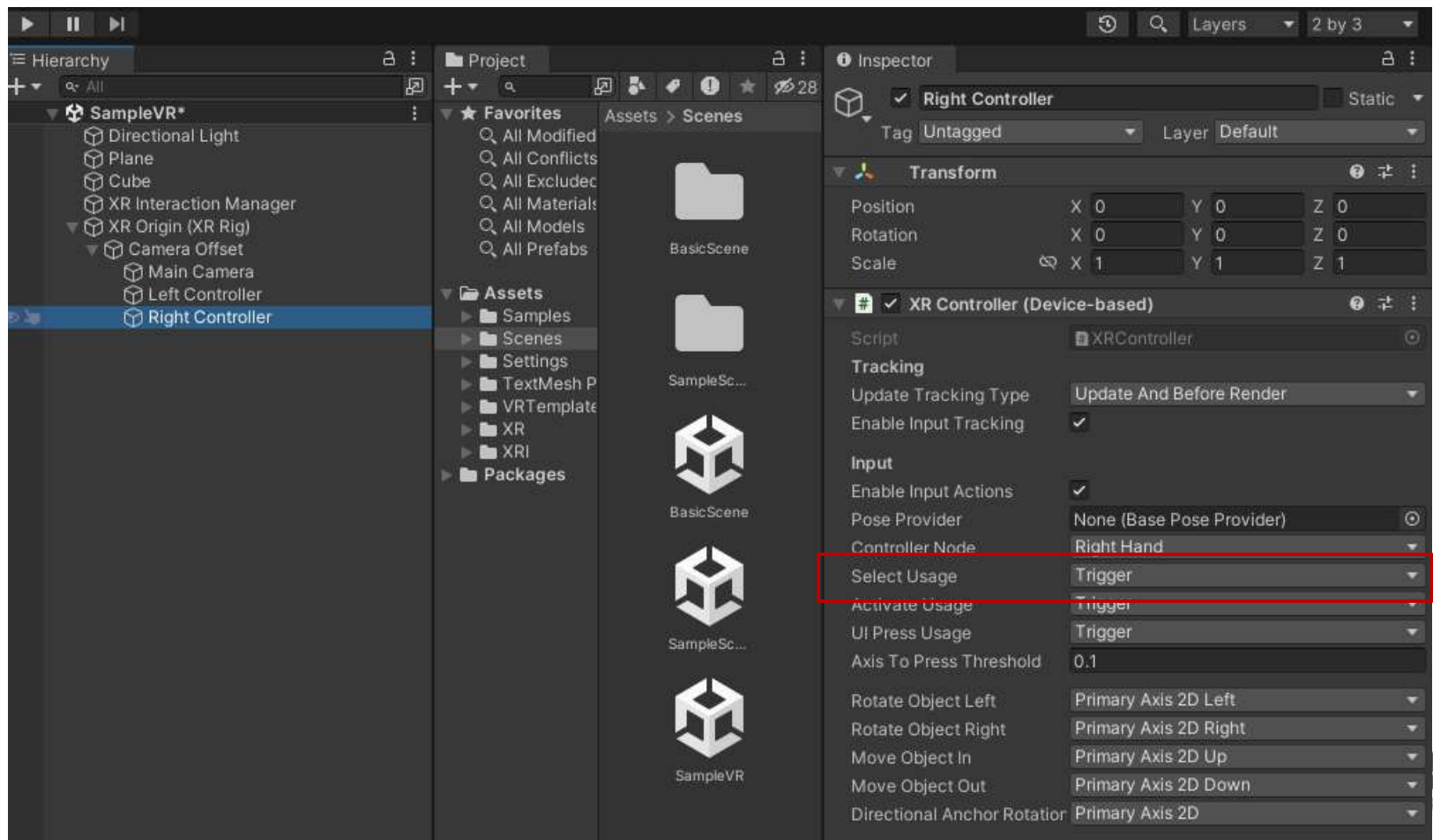
# Unity XR

- XR Setup
  - Hierarchy → XR → Device Based → XR Origin (VR) 추가
    - XR Origin (XR Rig), XR Interaction Manager이 추가



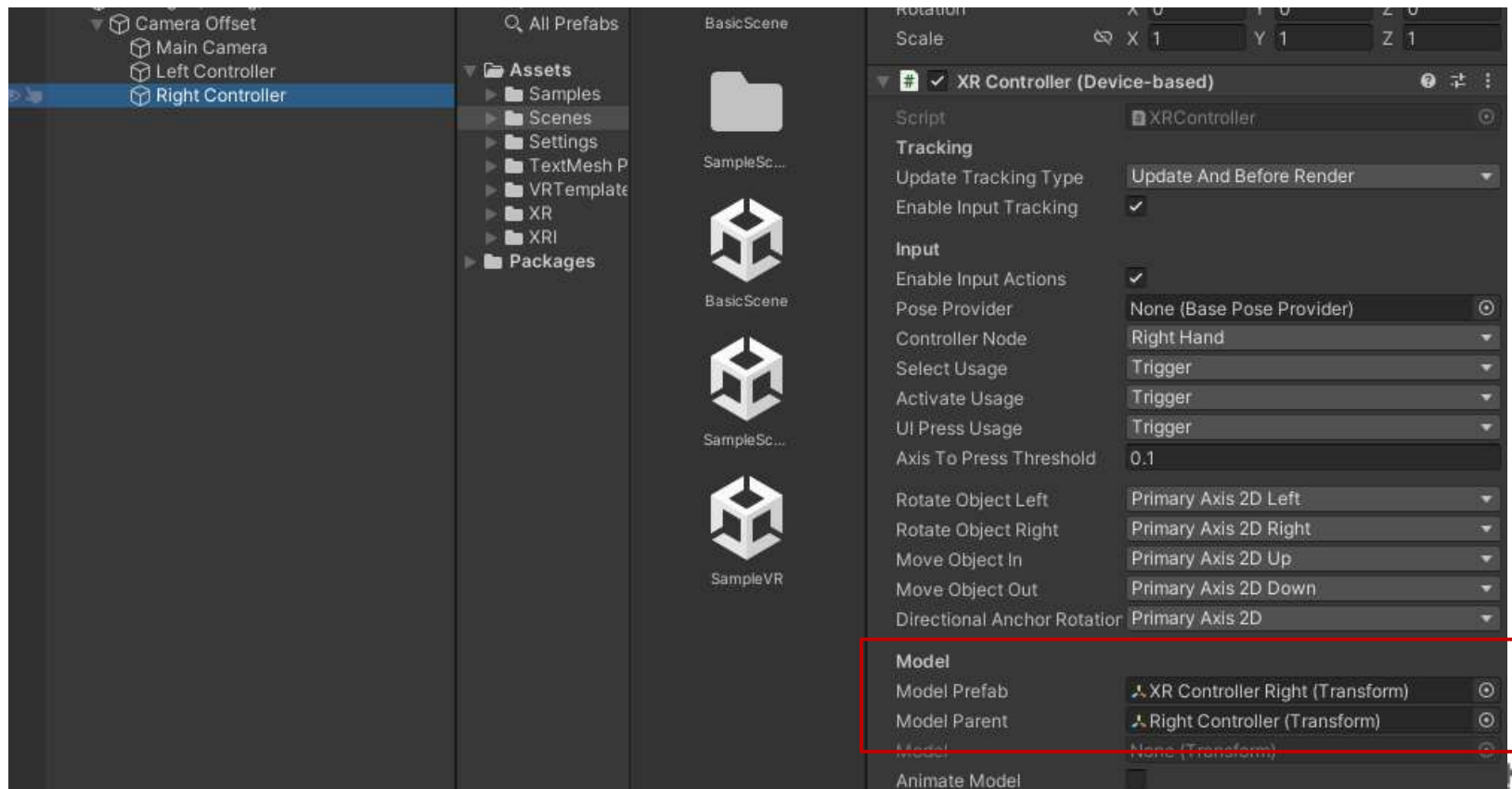
# Unity XR

- Controller
  - Left/Right Controller → Select Usage : Trigger로 변경



# Unity XR

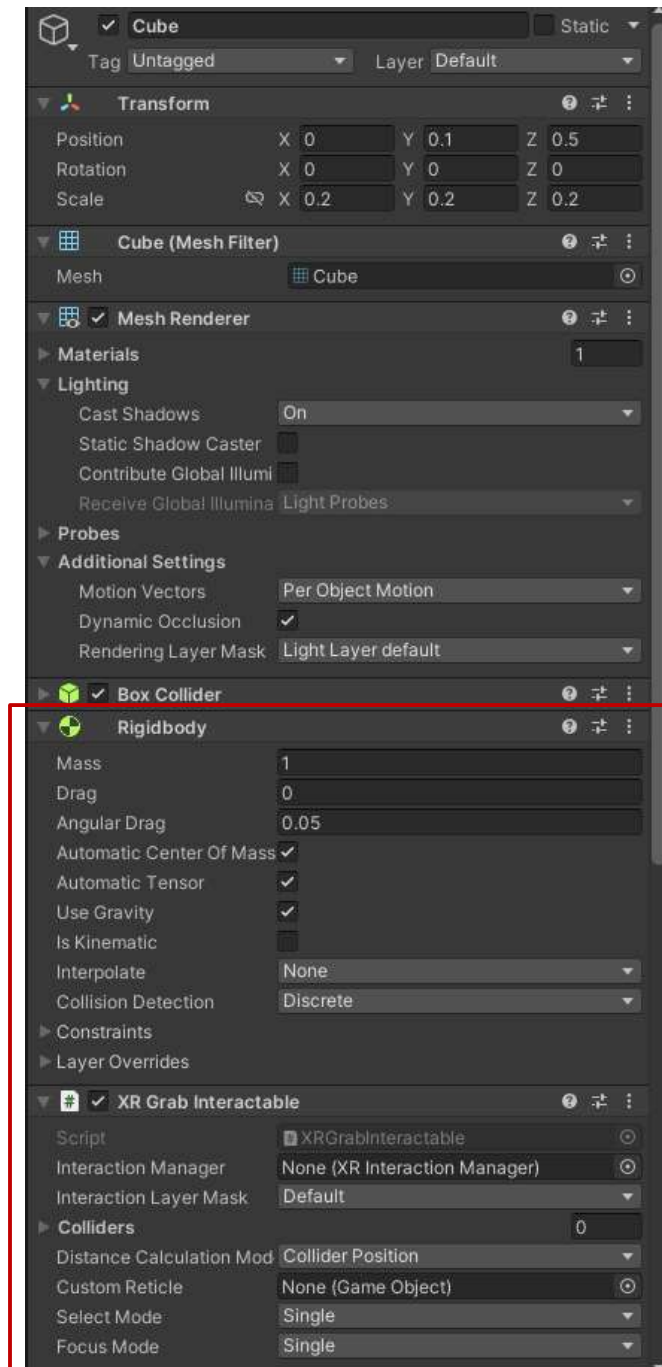
- Controller
  - Controller에 Model 연결





# Unity XR

- Interactable Object
  - Cube 객체에 대화형 속성 추가
    - XRGrabInteractable 컴포넌트 추가
      - Rigidbody 속성을 자동으로 추가됨

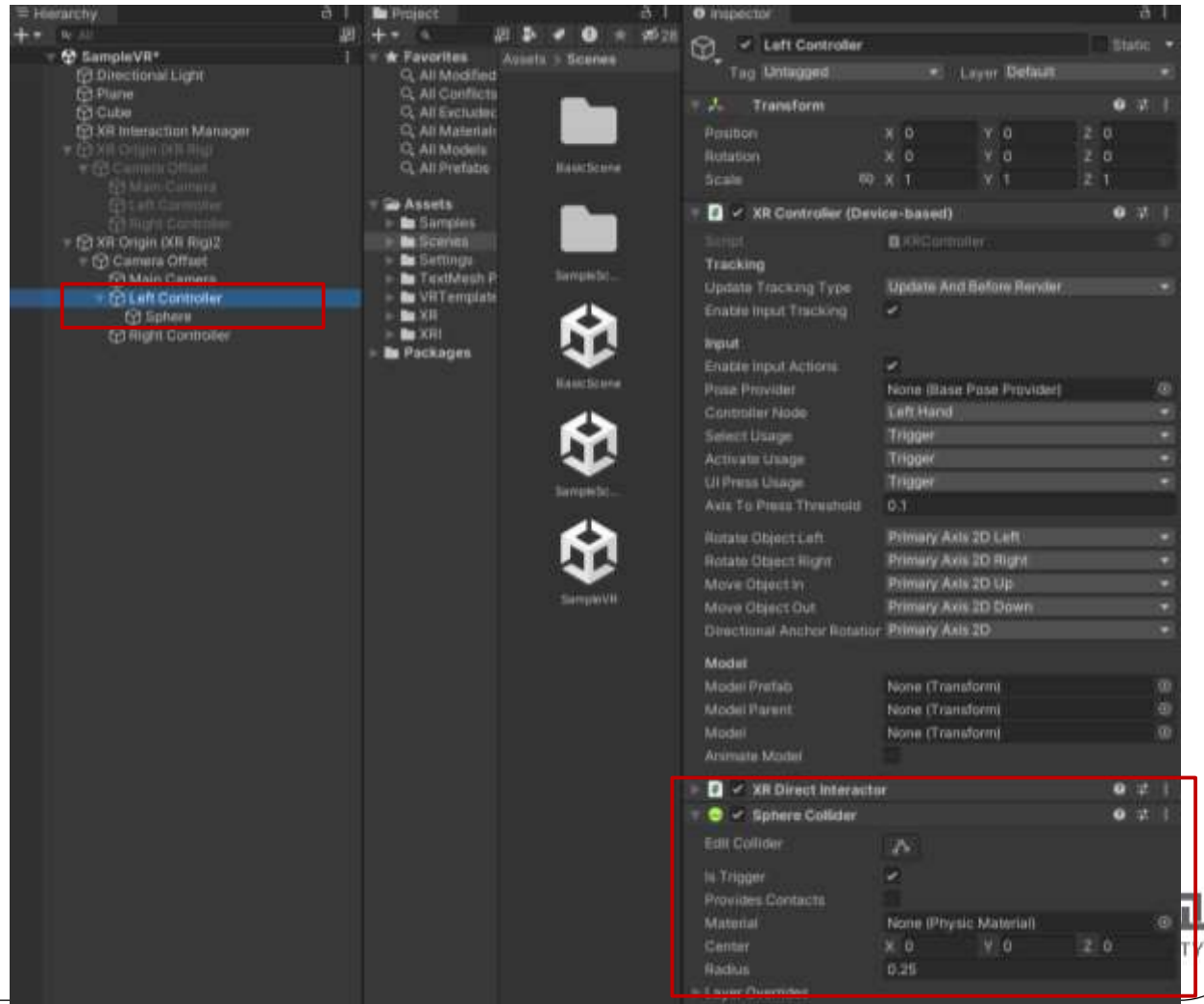


# Unity XR

- Direct Interaction
  - Controller 속성 중
    - XRRayInteractor,
    - LineRenderer,
    - XRInteractorLineVisual 삭제
  - XRDirectInteractor 속성 추가
  - Sphere Collider 속성 추가
- Controller 자식으로 구 객체 추가

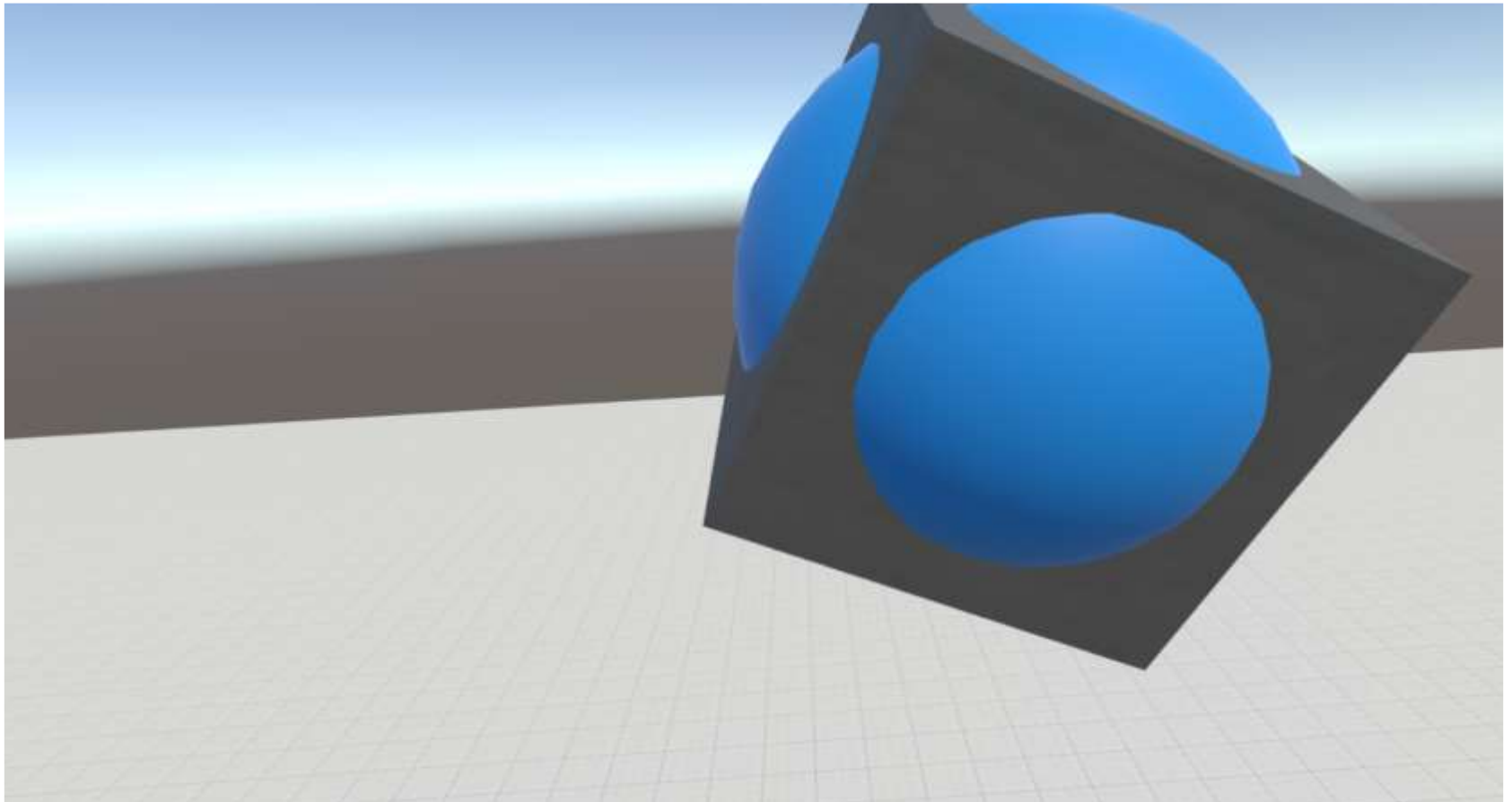
# Unity XR

- Direct Interaction



# Unity XR

- Direct Interaction



# Unity XR

- Script
  - Project → C# Script → cshSelect.cs 생성
  - Cube 객체에 등록

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Rendering.Universal;

public class cshSelect : MonoBehaviour
{
    public void CubeEnter(int val)
    {
        Debug.Log("Enter: " + val);
    }

    public void CubeExit(bool val)
    {
        Debug.Log("Exit: " + val);
    }
}
```

# Unity XR

- Controller Event Handler
  - XRDirectInteractor
    - Interactor Event
      - Select → Entered, Exited 속성 추가

