

# 오컬러스 퀘스트 추가 프로그래밍

## 1. 이준

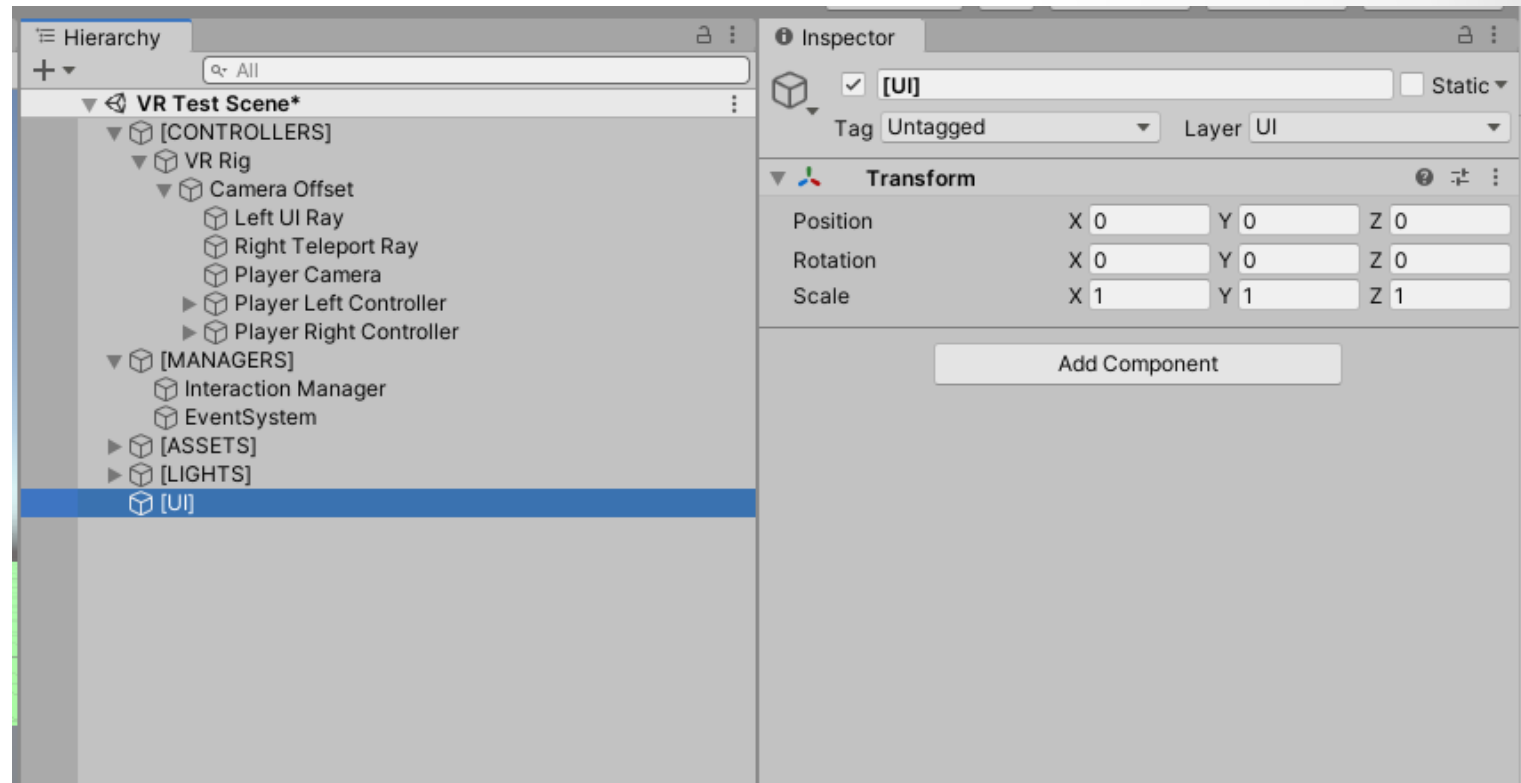
# 오컬러스 퀘스트 메뉴 UI

- 지난번 1차 강의때 메뉴 UI 작업

- 제가 드리는 파일을 다운로드 해서 확인 해보세요 (VR Setup)
- <https://drive.google.com/file/d/1izDsh5ljN64PvYfSazklkbDcxmGFoEC3/view?usp=sharing>
- 퀘스트 프로그래밍에 대해서 복습을 할 겸 World Space 카메라 모드 UI 와 Screen Space 카메라 모드를 각각 테스트 해보고 실습을 해볼 예정

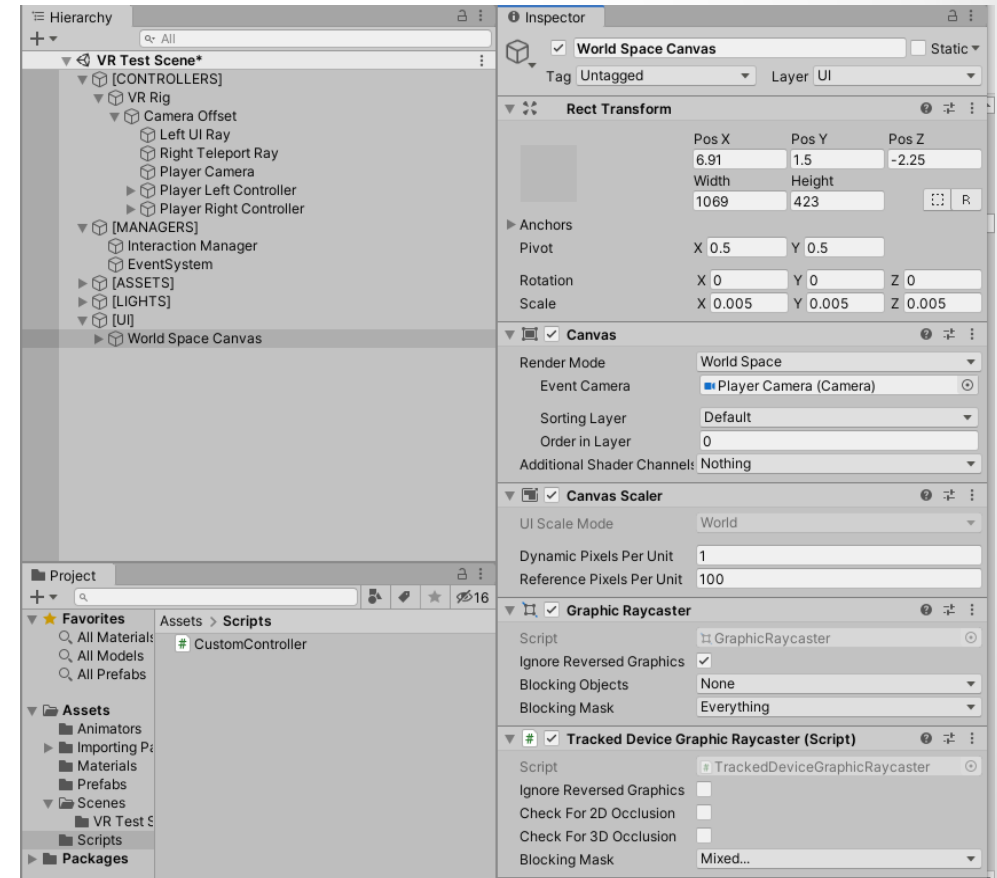
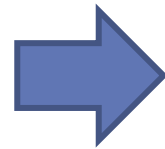
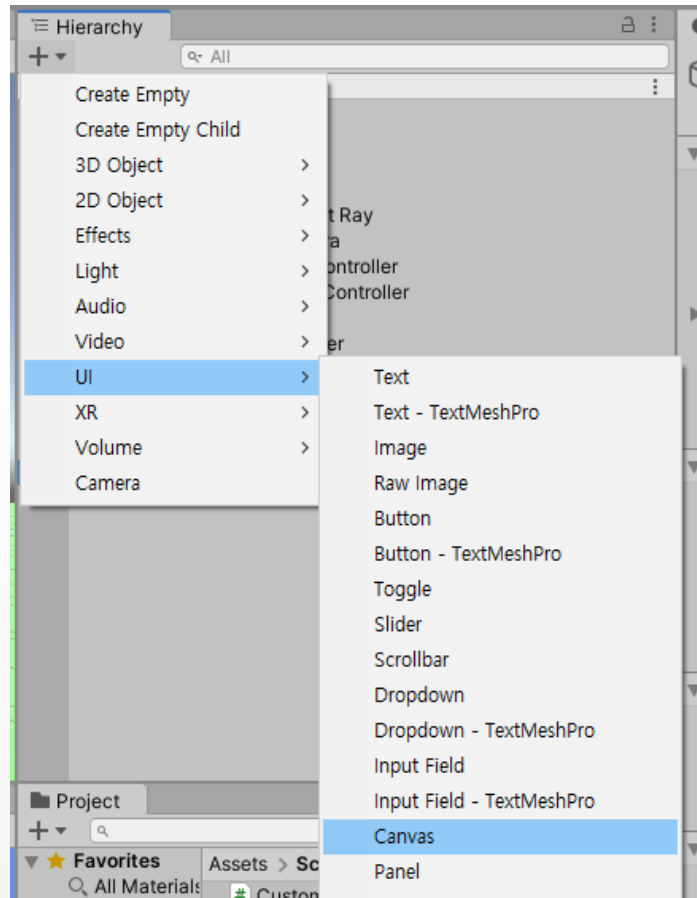
# 오컬러스 퀘스트 메뉴 UI

- CreateEmpty -> 이름을 [UI]로 변경
- 위치는 0, 0, 0으로 설정하기



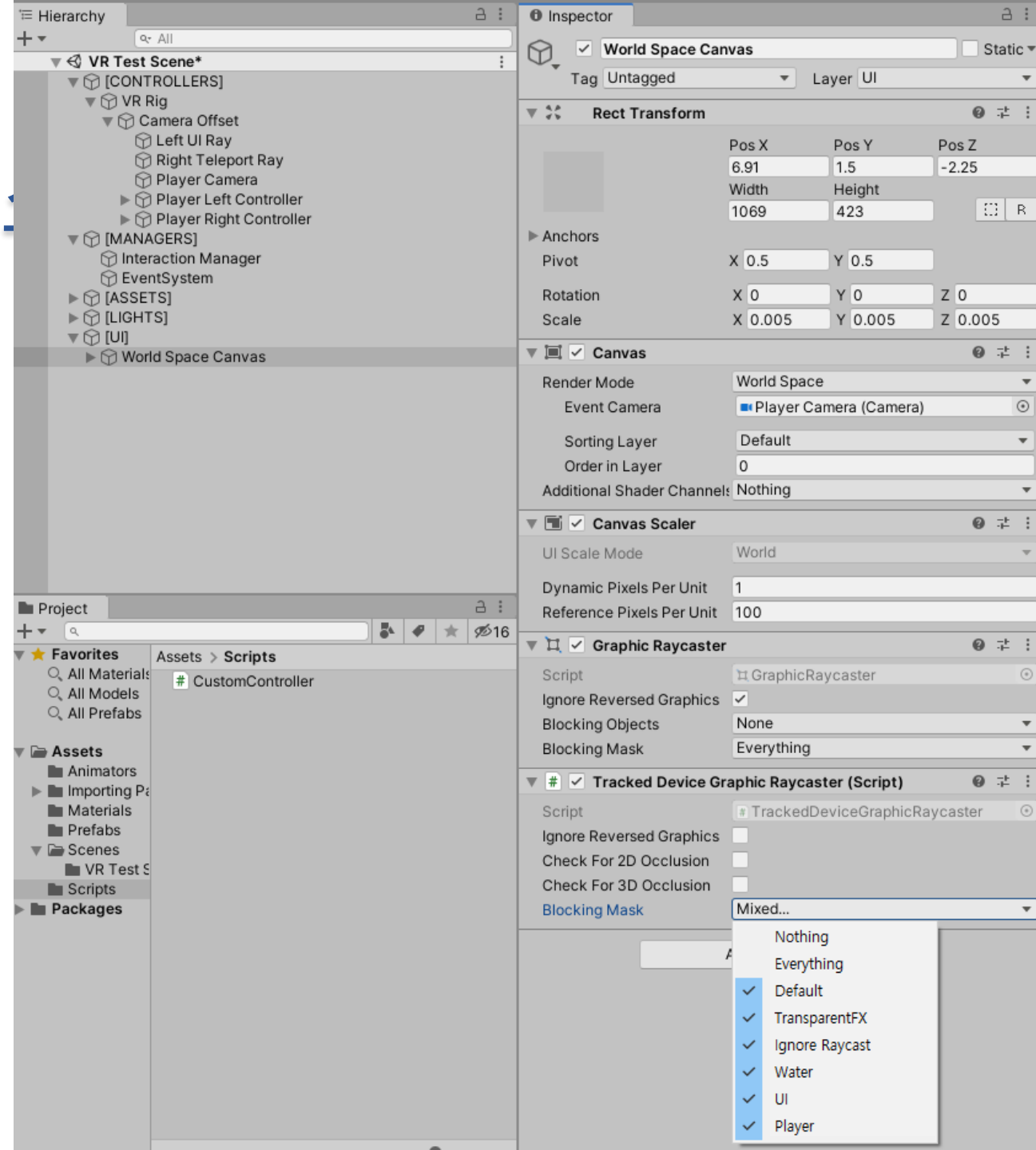
# 오컬러스 퀘스트 메뉴 UI

- 캔버스 추가 후 이름을 World Space Canvas 로 정하기
- Canvas의 위치 설정, World Space 설정 – Player Camera 설정



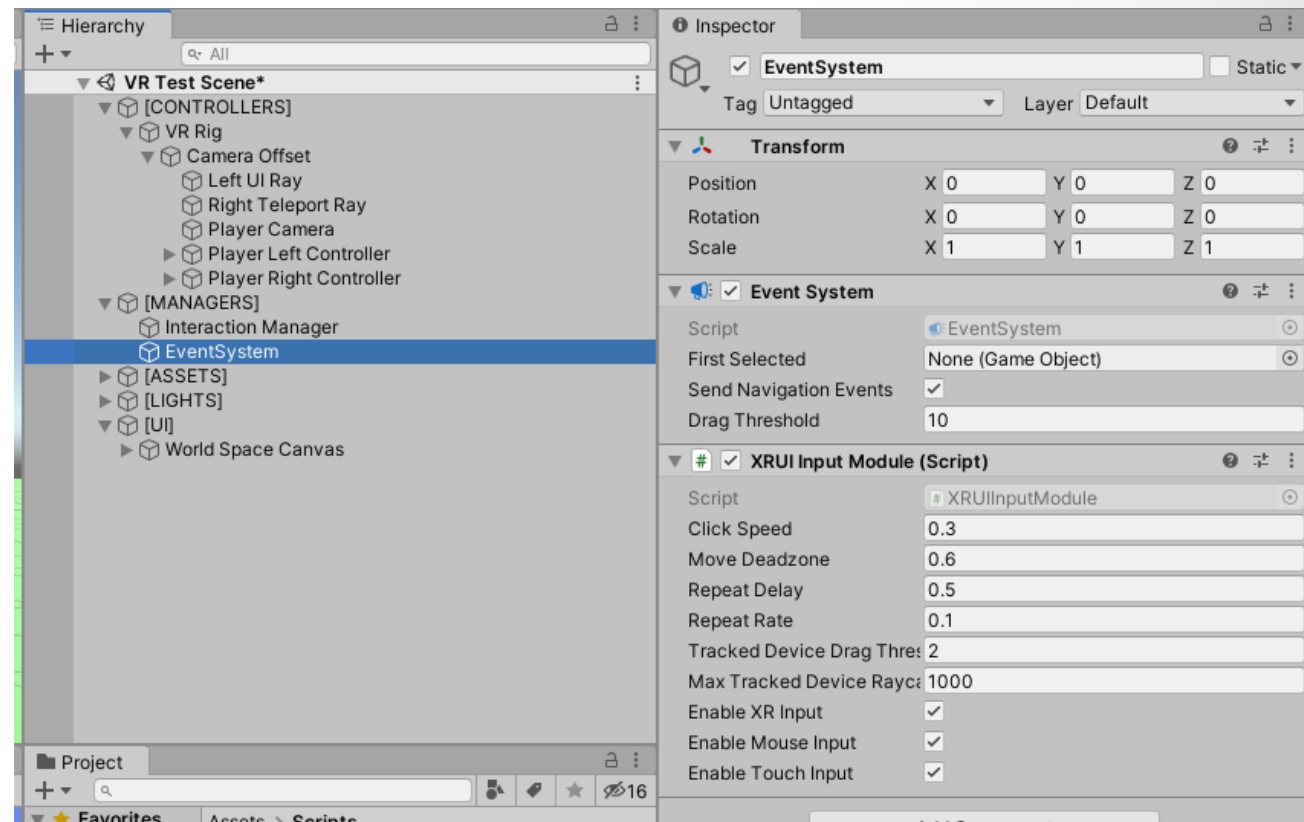
# 오컬러스 퀘스트

- Tracked Device Graphic Raycaster 추가후 Blocking Mask를 다음과 같이 설정하기



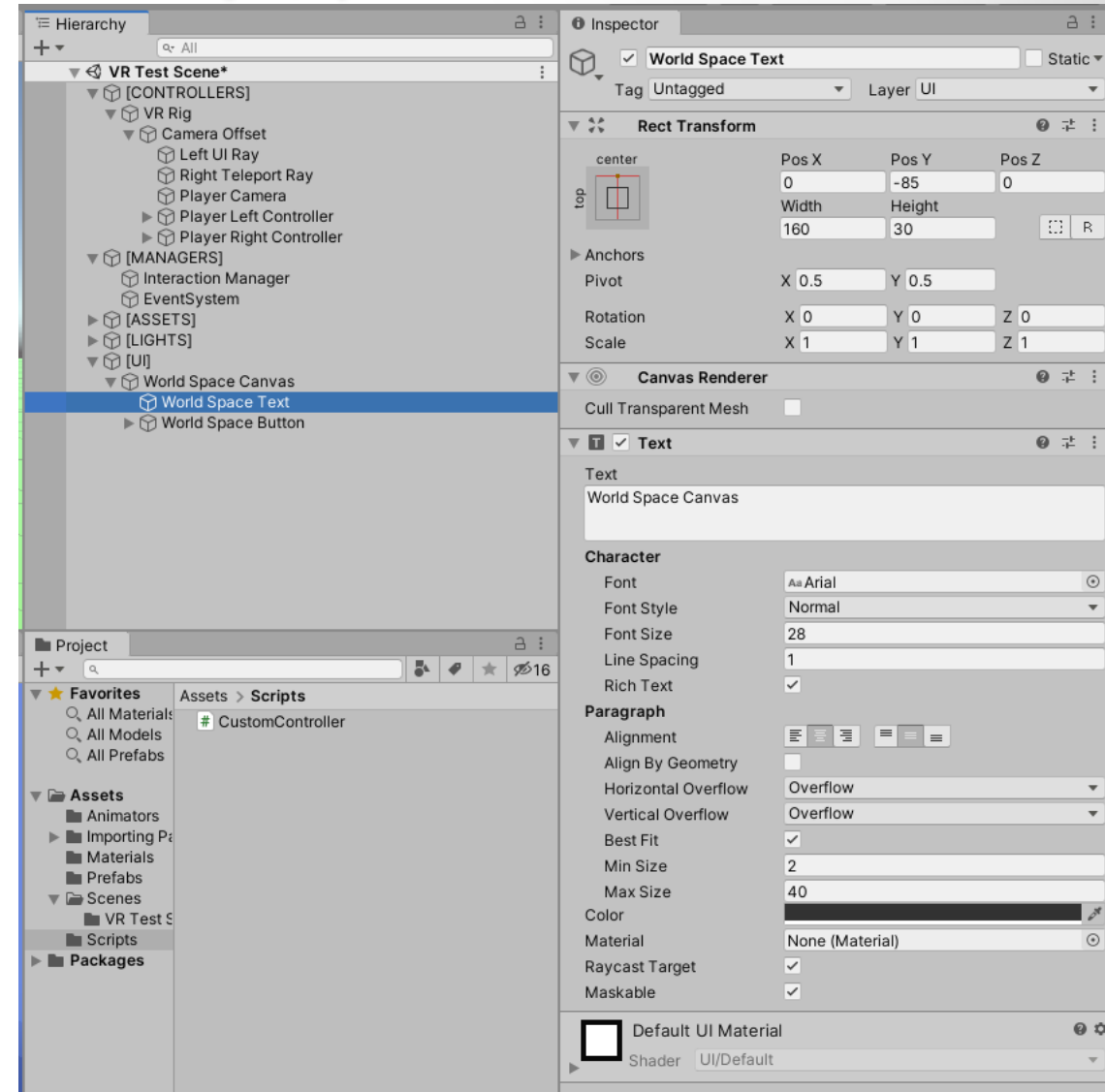
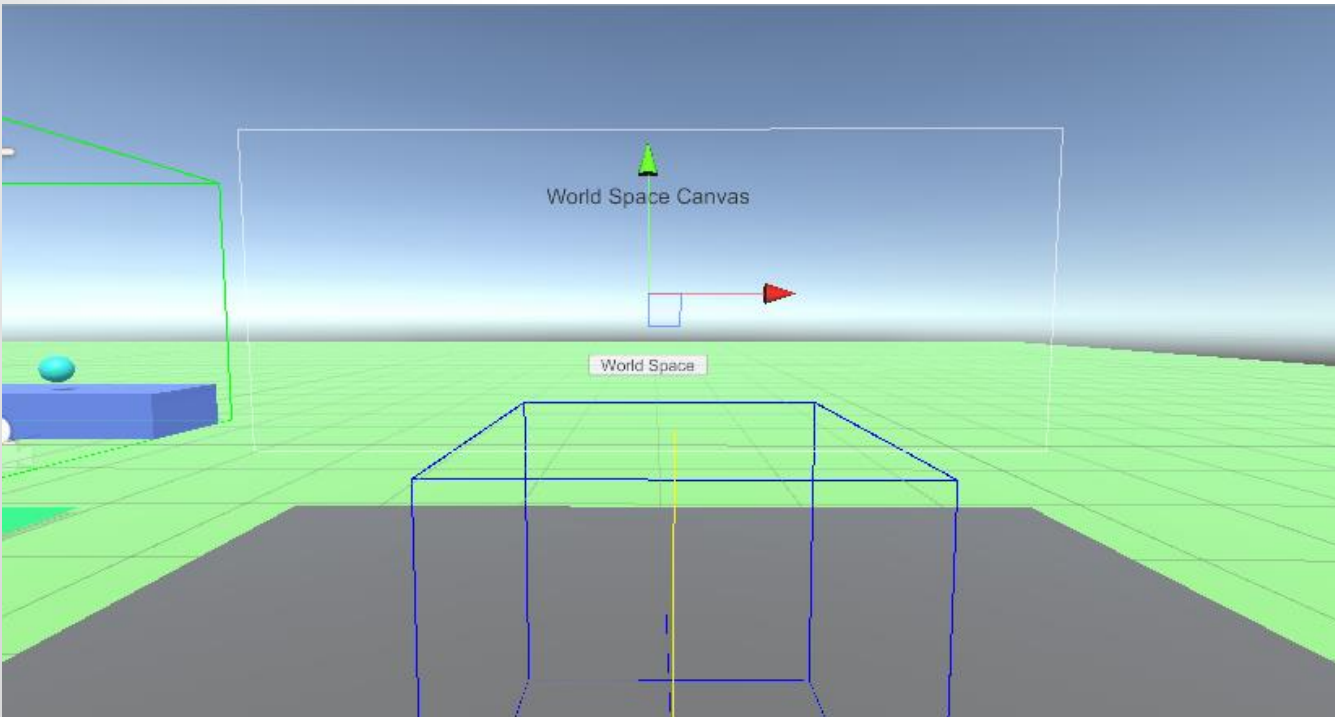
# 오컬러스 퀘스트 메뉴 UI

- EventSystem이 자동적으로 추가가 되는데, 이를 [MANAGERS] 객체 아래에 두고, XRUI Input Module로 설정을 확인하기!



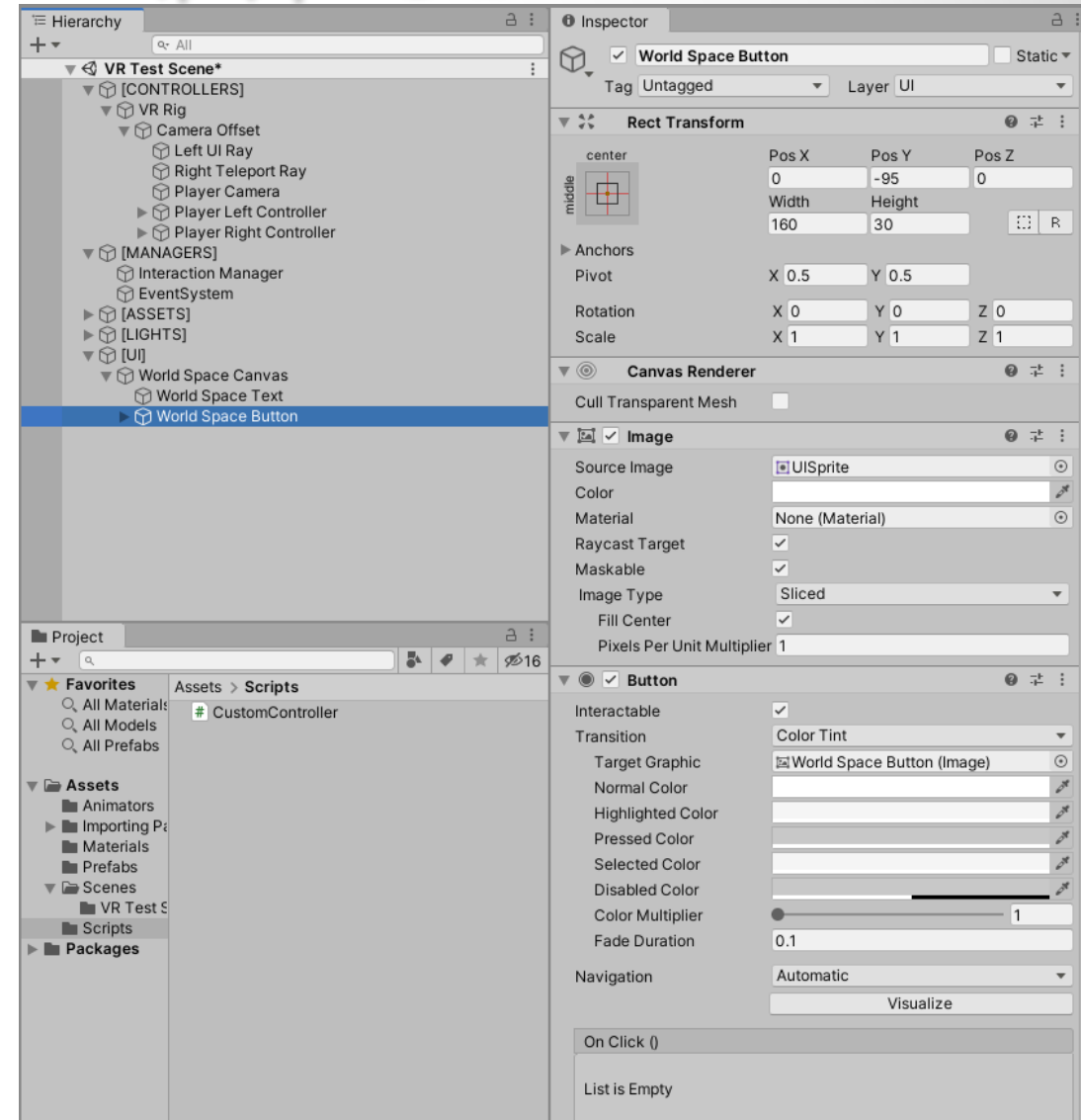
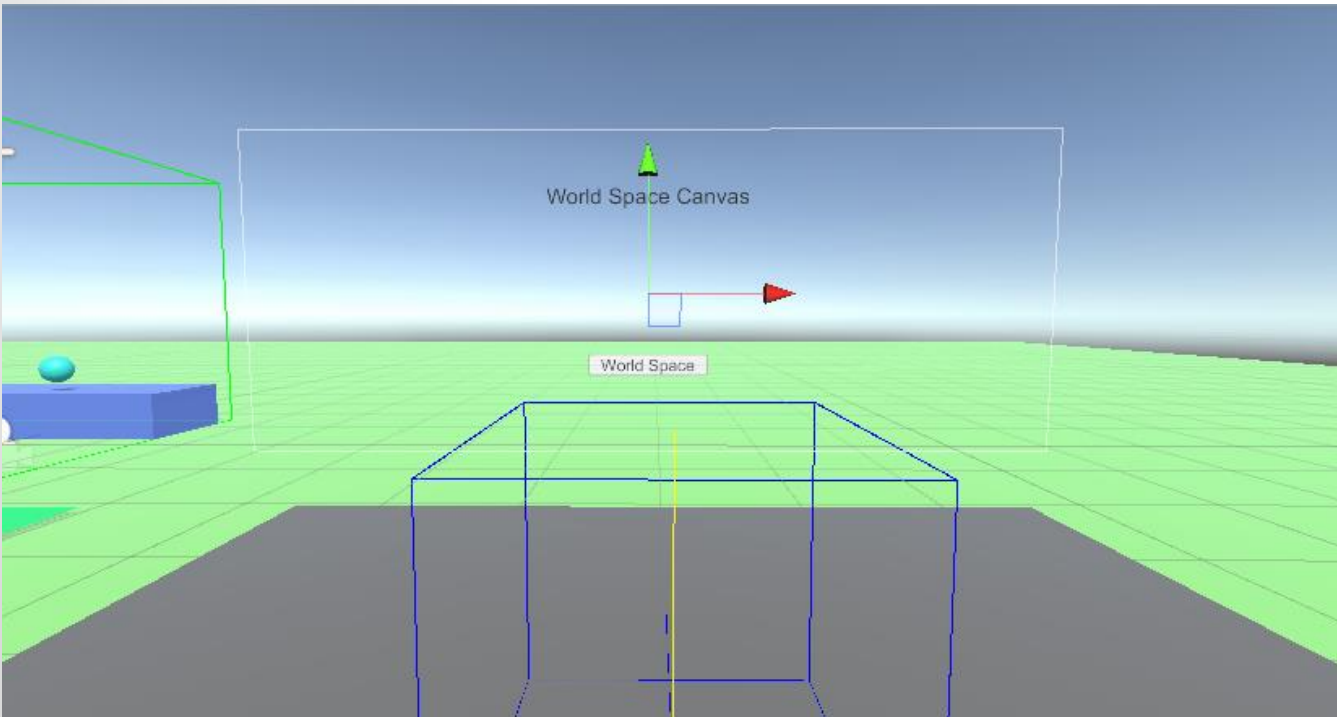
# 오컬러스 퀘스트 메뉴 UI

- World Space Canvas 배치
  - 아래에 텍스트와 버튼을 만듦



# 오컬러스 퀘스트 메뉴 UI

- World Space Canvas 배치
  - 아래에 텍스트와 버튼을 만듦



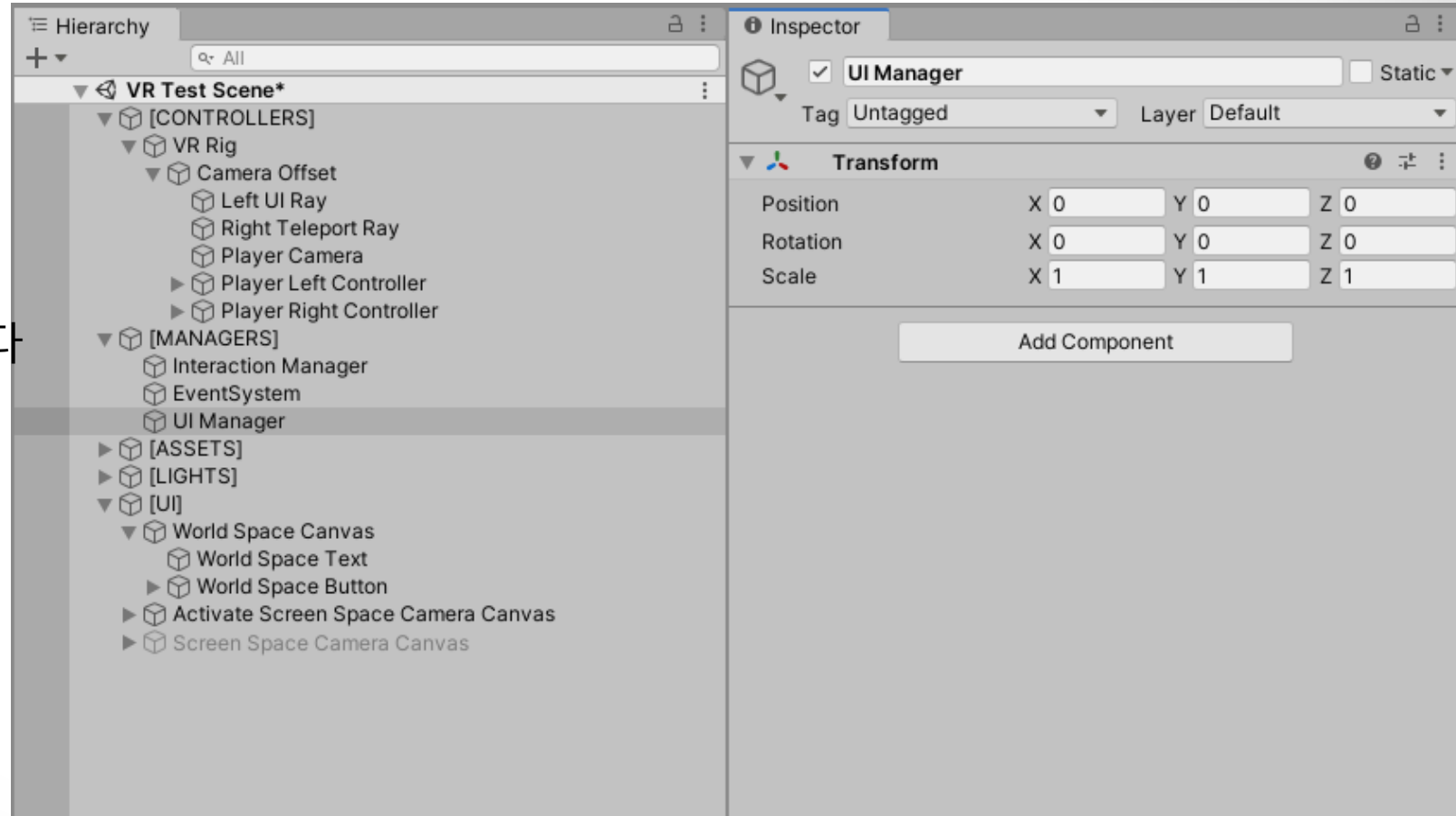


# 오컬러스 퀘스트 메뉴 UI

- [MANAGERS] 아래에 CreateEmptyObject 후 이름을 UI Manager라고 정함

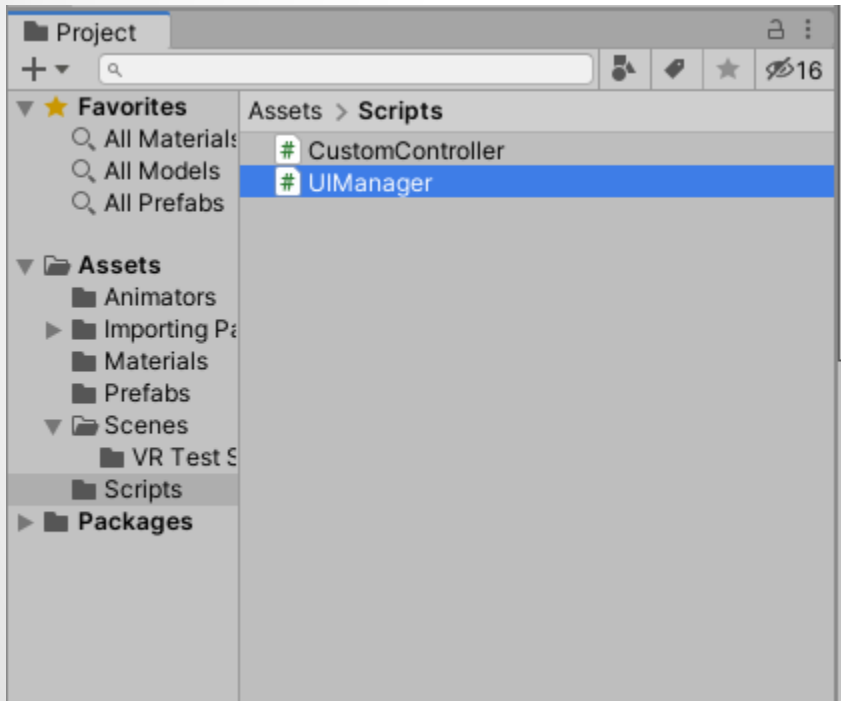
\n 이 키보드에 없는 경우  
\\n 으로 입력 하면 됩니다

한글 키보드 특징상 이렇게 표현이 됩니다



# 오컬러스 퀘스트 메뉴 UI

- UIManager.cs 파일 생성후 다음과 같이 코딩
- 버튼 클릭시 이동이 이루어지도록 함



```
UIManager.cs
Assembly-CSharp

using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class UIManager : MonoBehaviour
{
    // World Space UI & Button
    [SerializeField]
    private Button worldSpaceButton;
    [SerializeField]
    private Text worldSpaceText;

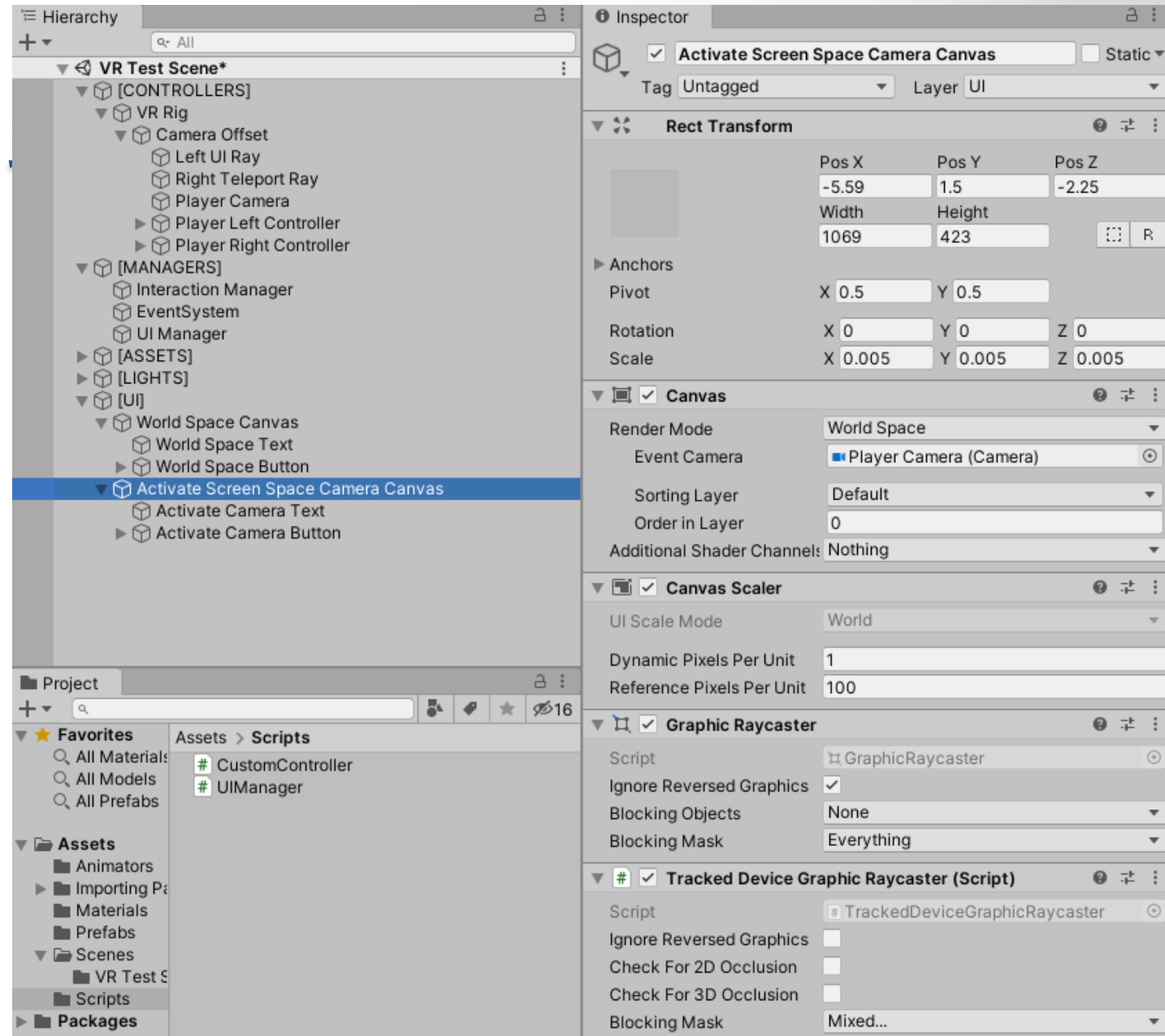
    // Start is called before the first frame update
    void Start()
    {
        worldSpaceButton.onClick.AddListener(() =>
        {
            worldSpaceText.text = "World Space Canvas\nButton Clicked";
        });
    }
}
```

# 오컬러스

- Canvas – Screen Space

- 설정 메뉴를 누르면 플레이어 화면을 따라 움직이는 형태로 구성할 수 있음 (일반 오컬러스 메뉴들)

- World Space Canvas를 복제 (Ctrl+D) 후 다음과 같이 이름 변경

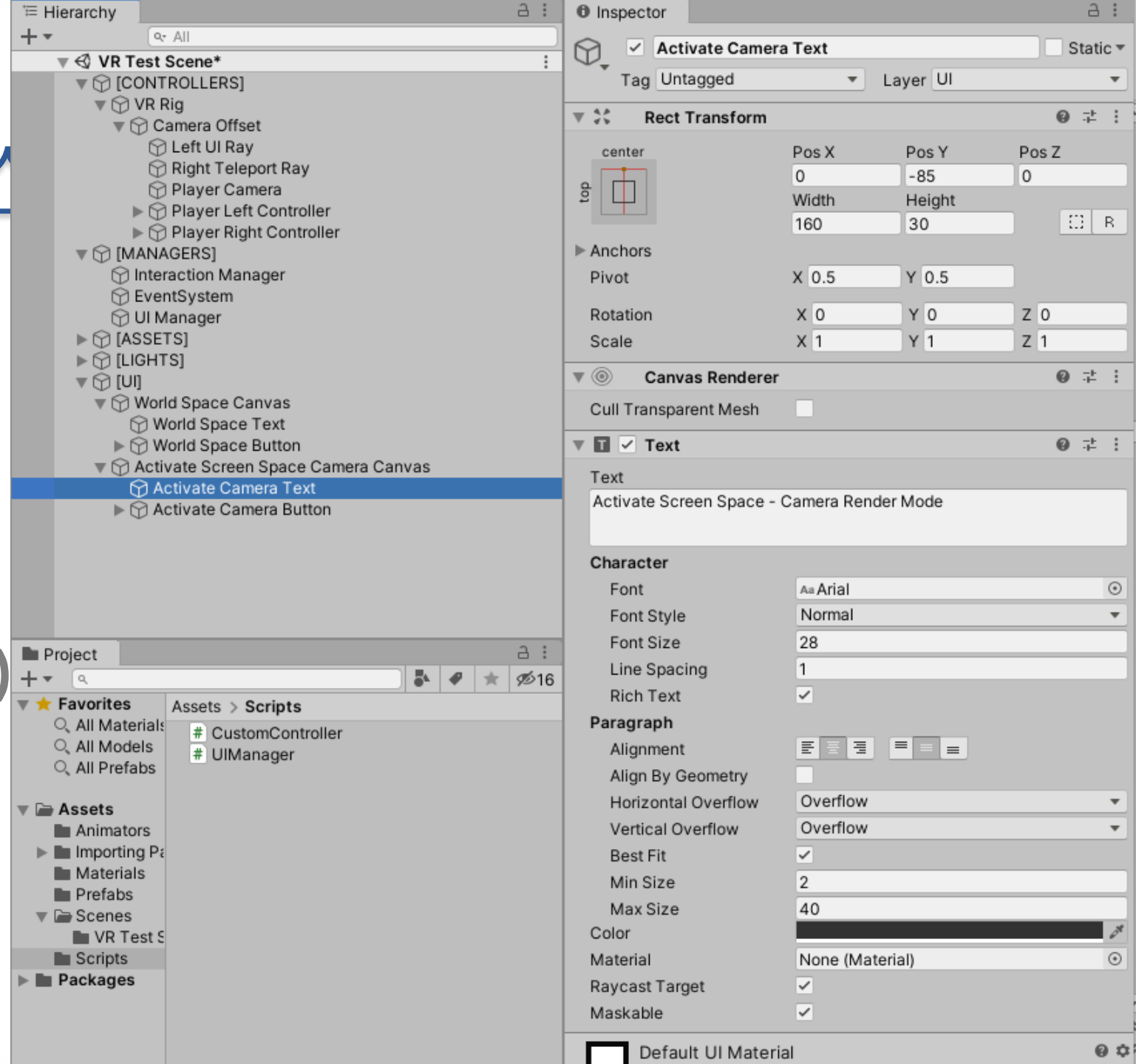


# 오쿨러스

- Canvas – Screen Space

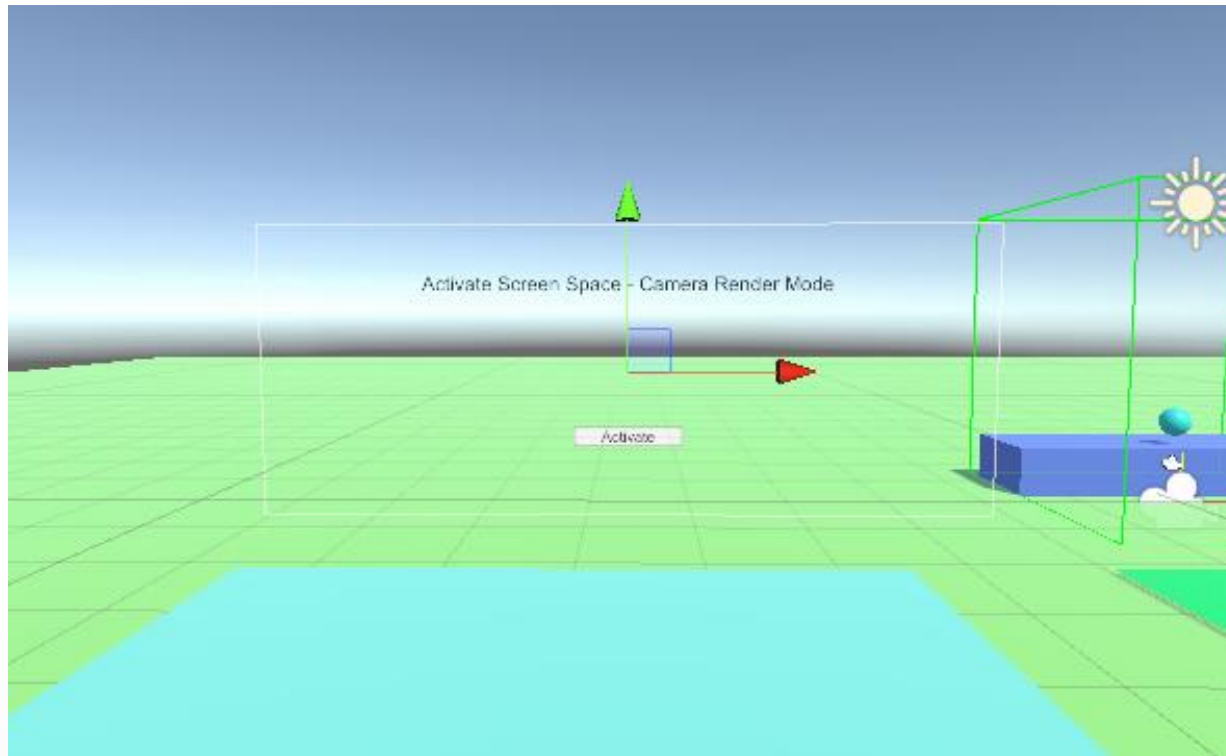
- 설정 메뉴를 누르면 플레이어 화면을 따라 움직이는 형태로 구성할 수 있음 (일반 오쿨러스 메뉴들)

- World Space Canvas를 복제 (Ctrl+D) 후 다음과 같이 이름 변경



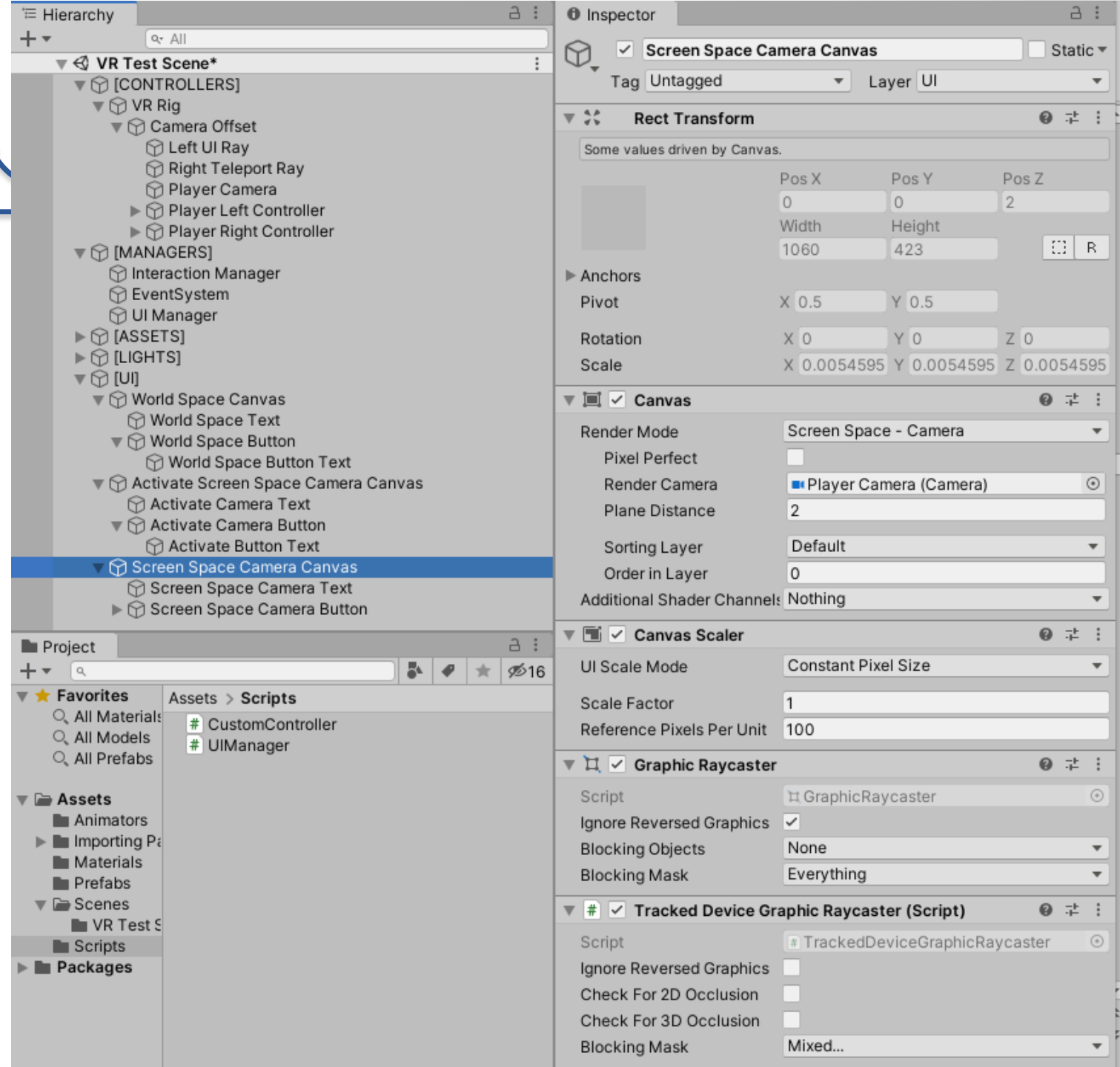
# 오컬러스 퀘스트 메뉴 UI

- 이 화면에서 버튼을 누르면, Screen Space Canvas 가 활성화 되고 메뉴 선택후 돌아가기 버튼을 누르면 다시 돌아 감



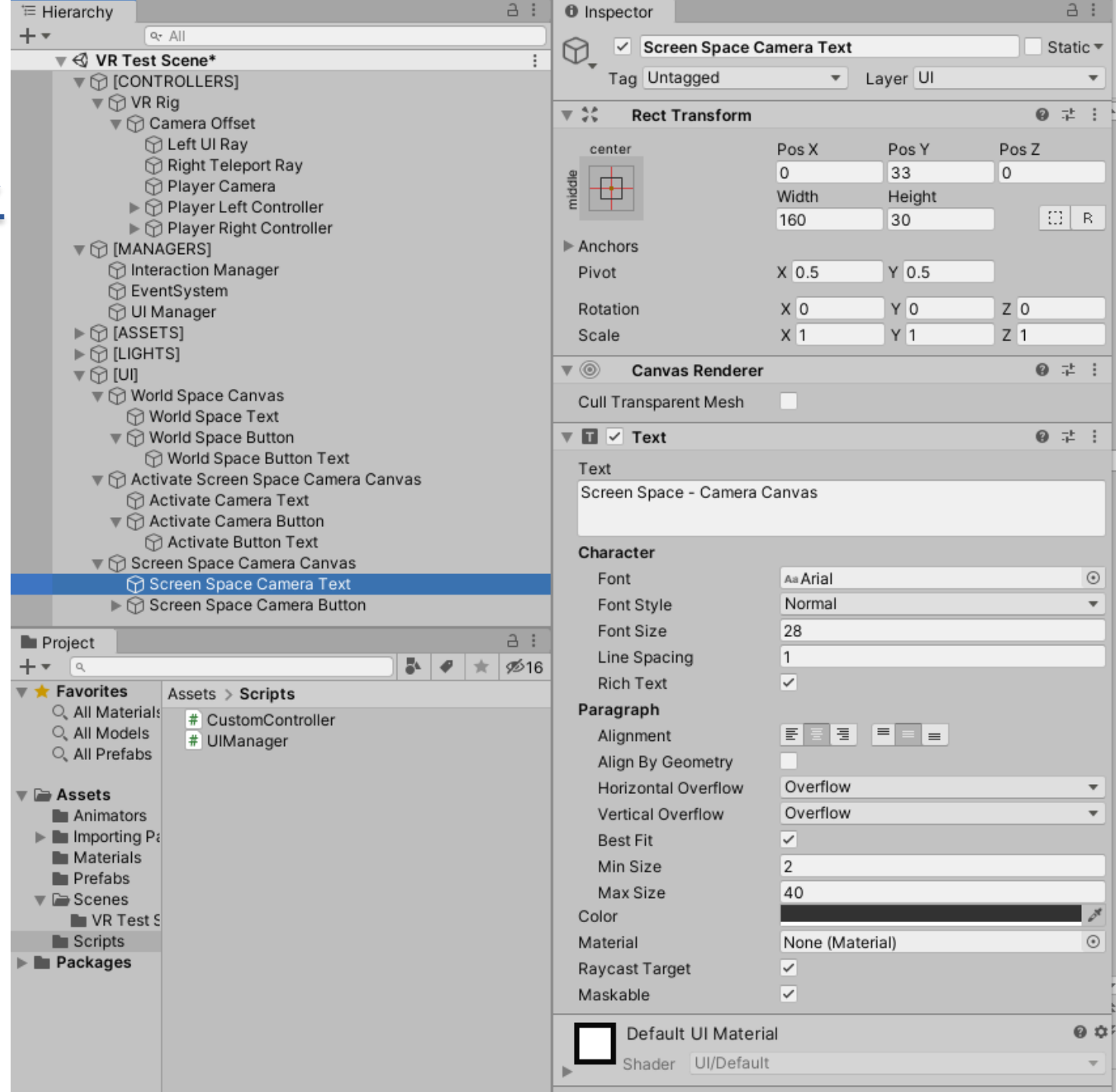
# 오컬러스

- 다시 WorldSpace Canvas를 복사하고 다음과 같이 이름을 변경함
- Screen Space –Camera 설정
- 카메라에 의존적이기 때문에 자동적으로 위치는 0, 0, 0에 위치함



# 오컬러스

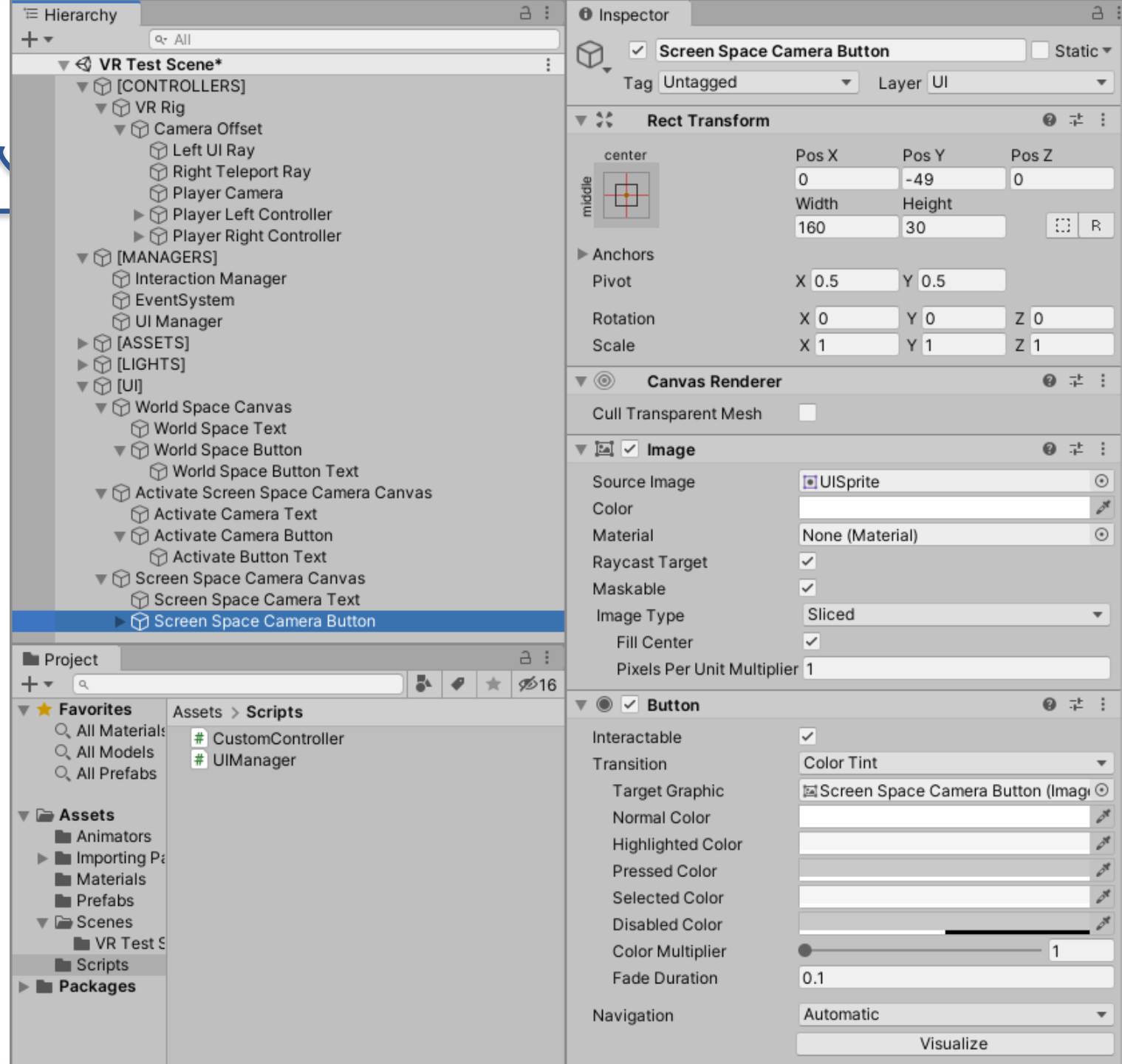
- Screen Space Camera Text 설정





# 오컬러스

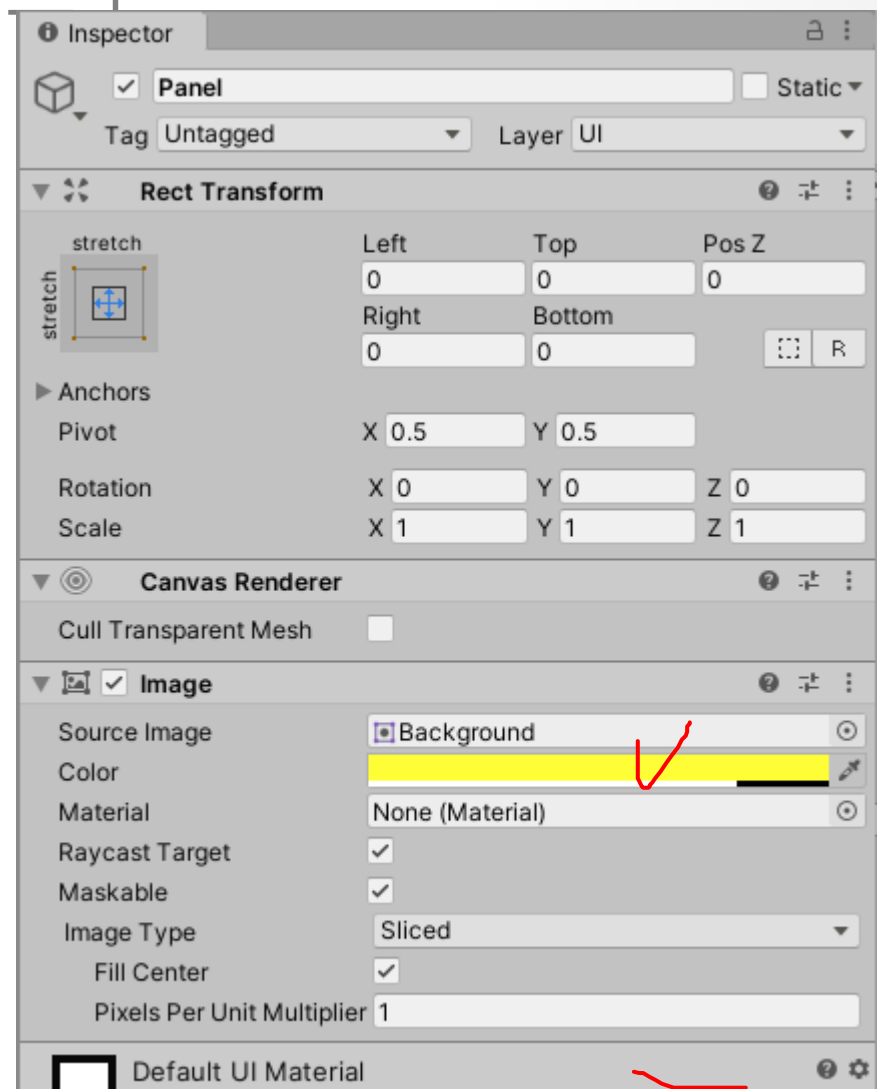
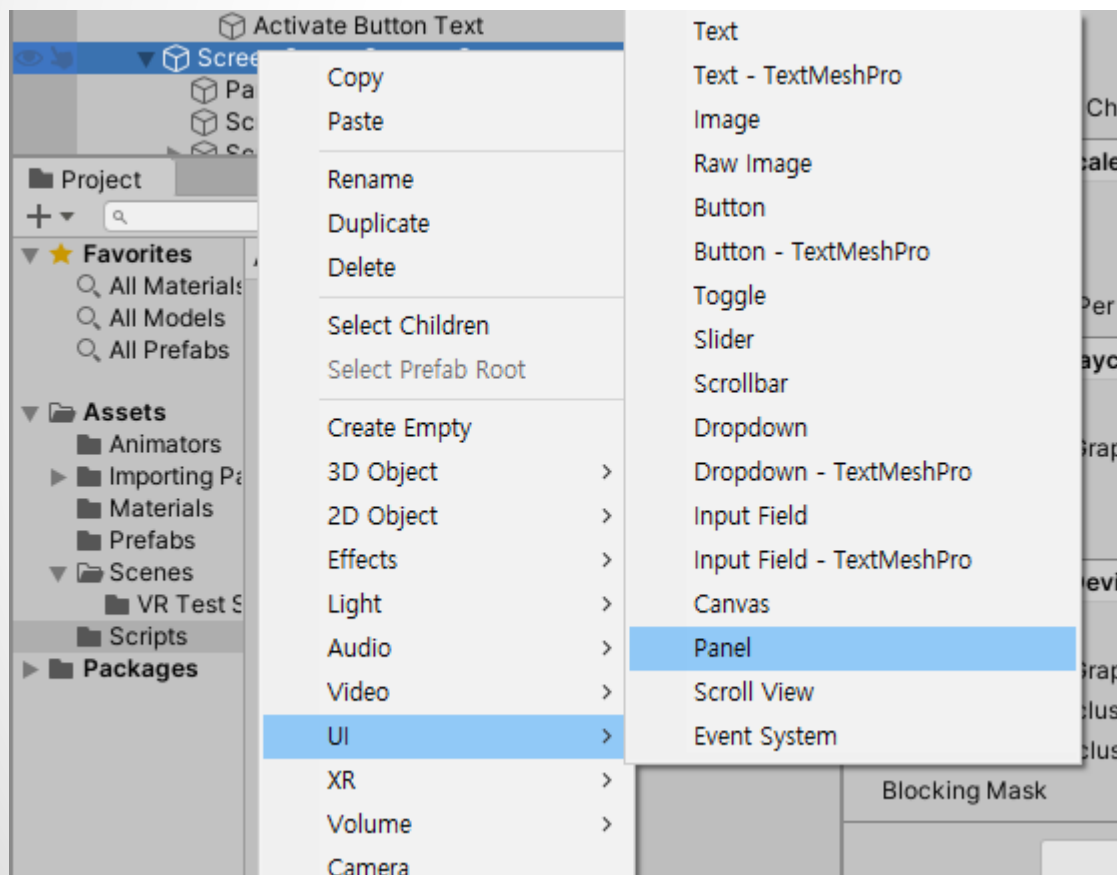
- Screen Space Camera Button 설정





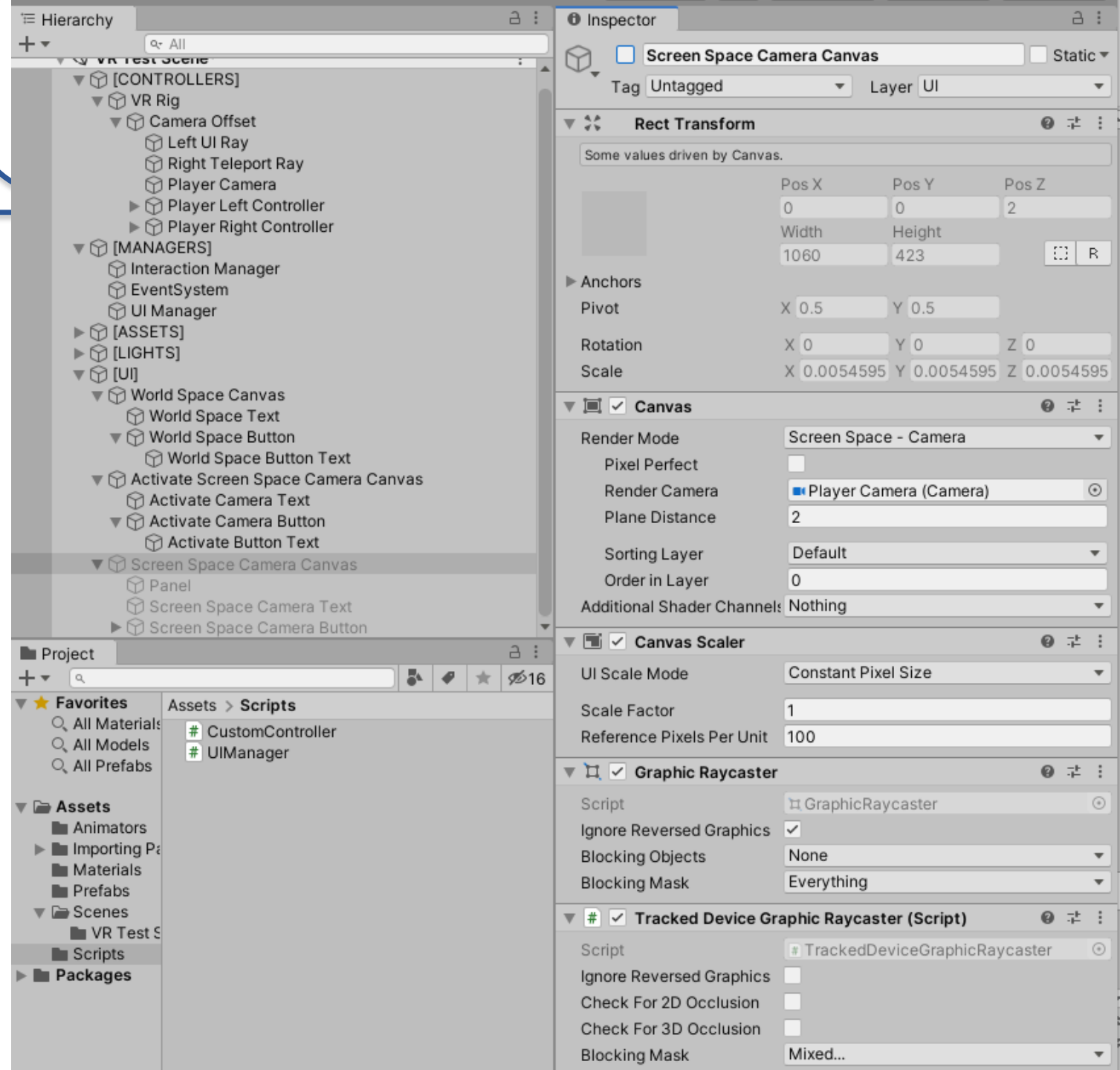
# 오컬러스 퀘스트 메뉴 UI

- 패널을 추가하고 Image 의 색을 노랑색으로 설정 (구분하기 쉽게 하기 위해서)



# 오컬러스

- Screen Space Camera Canvas 는 비활성화



# 오컬러스 퀘스트 메뉴 UI

- UIManager.cs 파일 수정하기
- Screen Space 버튼 클릭 및 캔버스 접근을 위한 변수 생성

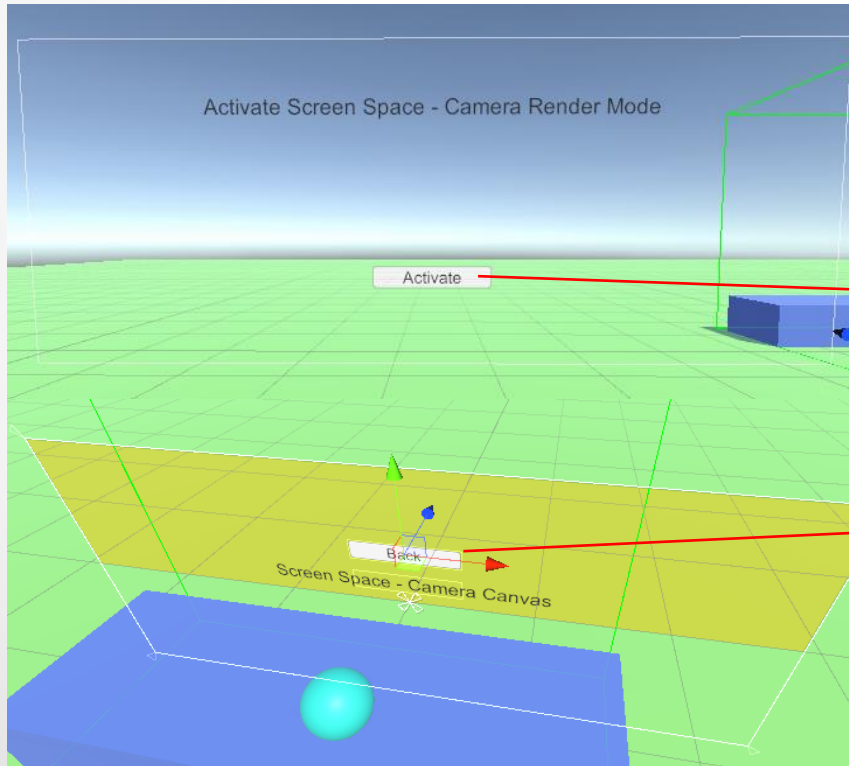
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class UIManager : MonoBehaviour
{
    // World Space UI & Button
    [SerializeField]
    private Button worldSpaceButton;
    [SerializeField]
    private Text worldSpaceText;
    [SerializeField]
    private Button activateScreenSpaceCameraCanvasButton;

    // Screen Space - Camera UI & Button
    [SerializeField]
    private GameObject screenSpaceCameraCanvas;
    [SerializeField]
    private Button screenSpaceCameraButton;
```

# 오컬러스 퀘스트 메뉴 UI

- UIManager.cs 파일 수정하기
- Start 함수에 해당 부분을 수정



```
// Start is called before the first frame update
private void Start()
{
    #region UI EVENT
    worldSpaceButton.onClick.AddListener(() =>
    {
        worldSpaceText.text = "World Space Canvas\nButton Clicked";
    });

    activateScreenSpaceCameraCanvasButton.onClick.AddListener(() =>
    {
        screenSpaceCameraCanvas.SetActive(true);
    });

    screenSpaceCameraButton.onClick.AddListener(() =>
    {
        screenSpaceCameraCanvas.SetActive(false);
    });
    #endregion
}
```

# 오컬러스 퀘스트 메뉴 UI

- UI Manager 의 객체 설정후 (꼼꼼히) 테스트해보기!

