가상현실 과제4

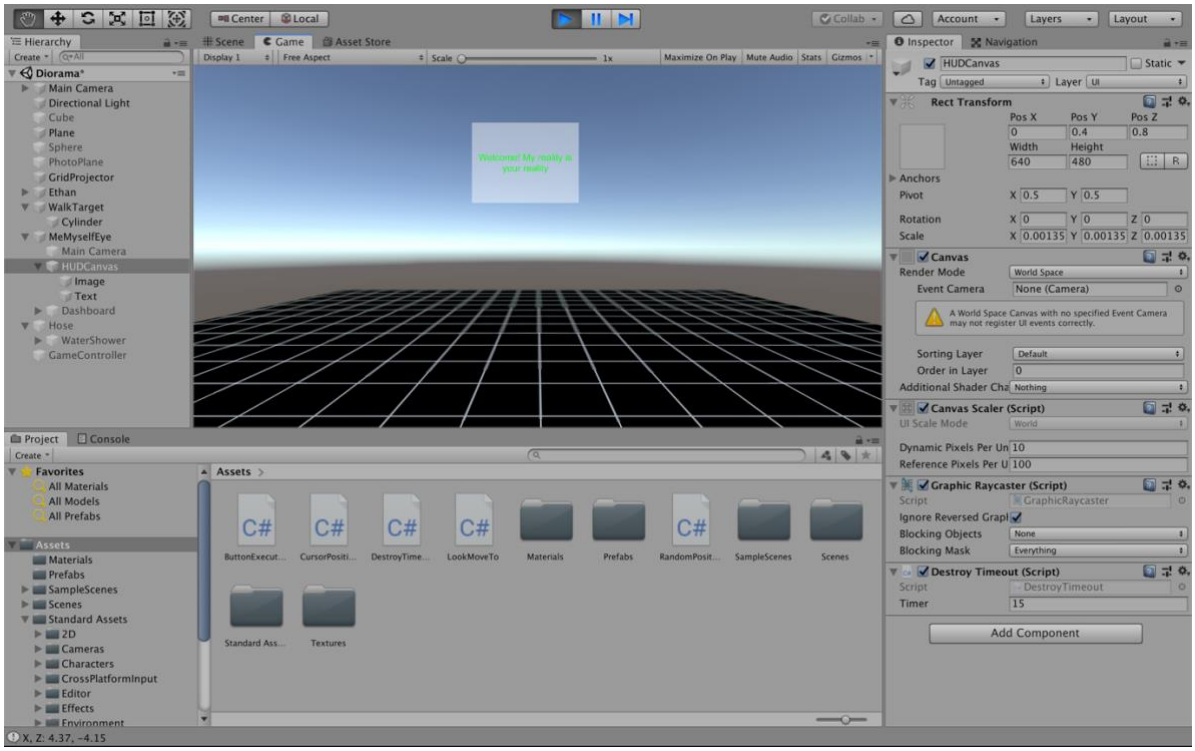
17011599 안정연

- Unity 버전 : 2018.4.9f1

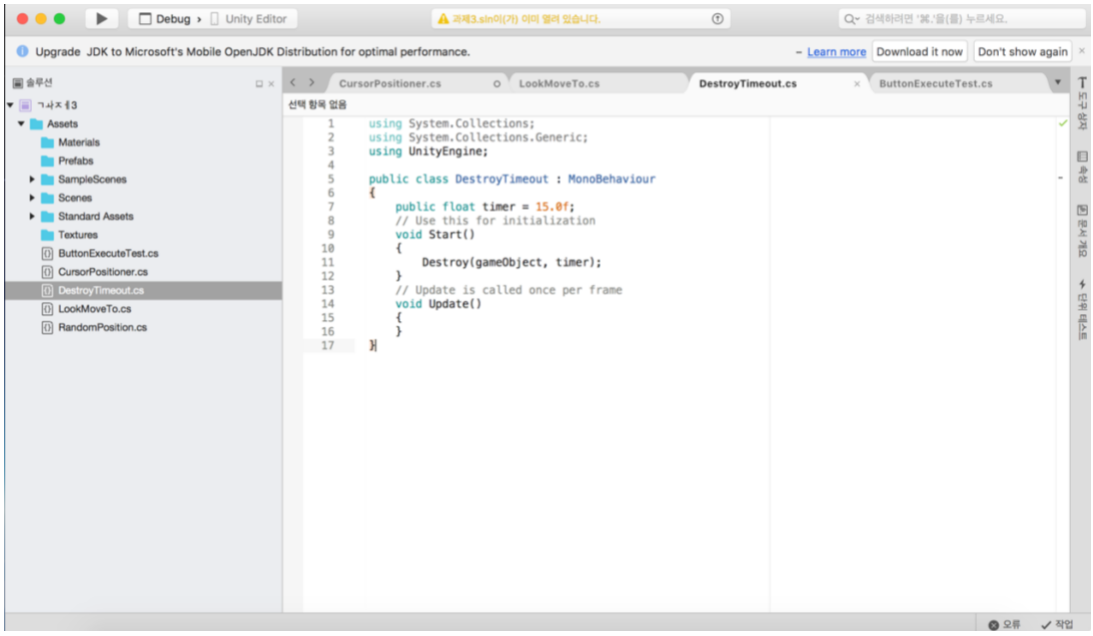
- 구현 기능 :

1. 윈드실드 HUD

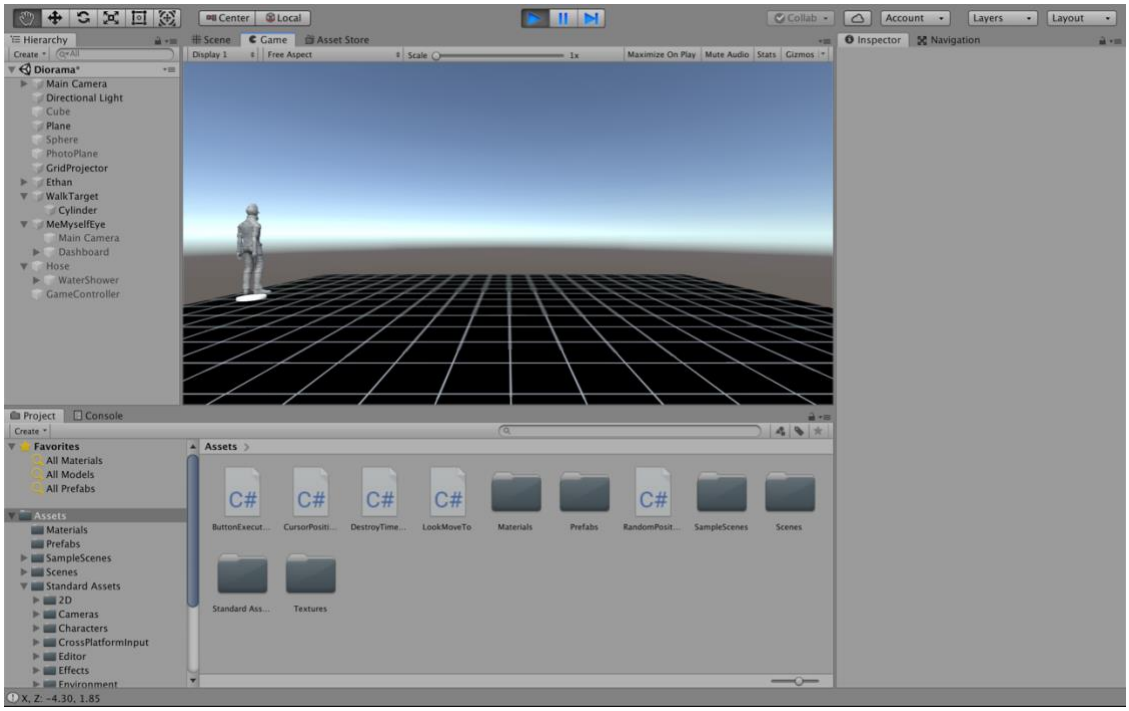
1) 프로젝트 패널에서 DefaultCanvas 프리팹을 계층 패널에 있는 MeMyselfEye 오브젝트로 드래그
2) 이름을 HUDCanvas 로 바꾼 뒤 HUDCanvas 를 선택하고, Rect Transform 컴포넌트에서 PosX, PosY, PosZ 를 (0, 0.4, 0.8)로 설정
3) HUDCanvas 아래에서 Text 를 "Welcome! My reality is your reality" 로 변경, 색상을 녹색으로 변경
4) 패널을 반투명하게 만든다. HUDCanvas 아래의 Image 에서 이미지를 선택하고, 컬러 대화상자에서 Alpha 채널을 255->115 로 변경



5) HUDCanvas 가 게임이 시작되고 타이머가 종료된 후에 사라지는 스크립트를 작성

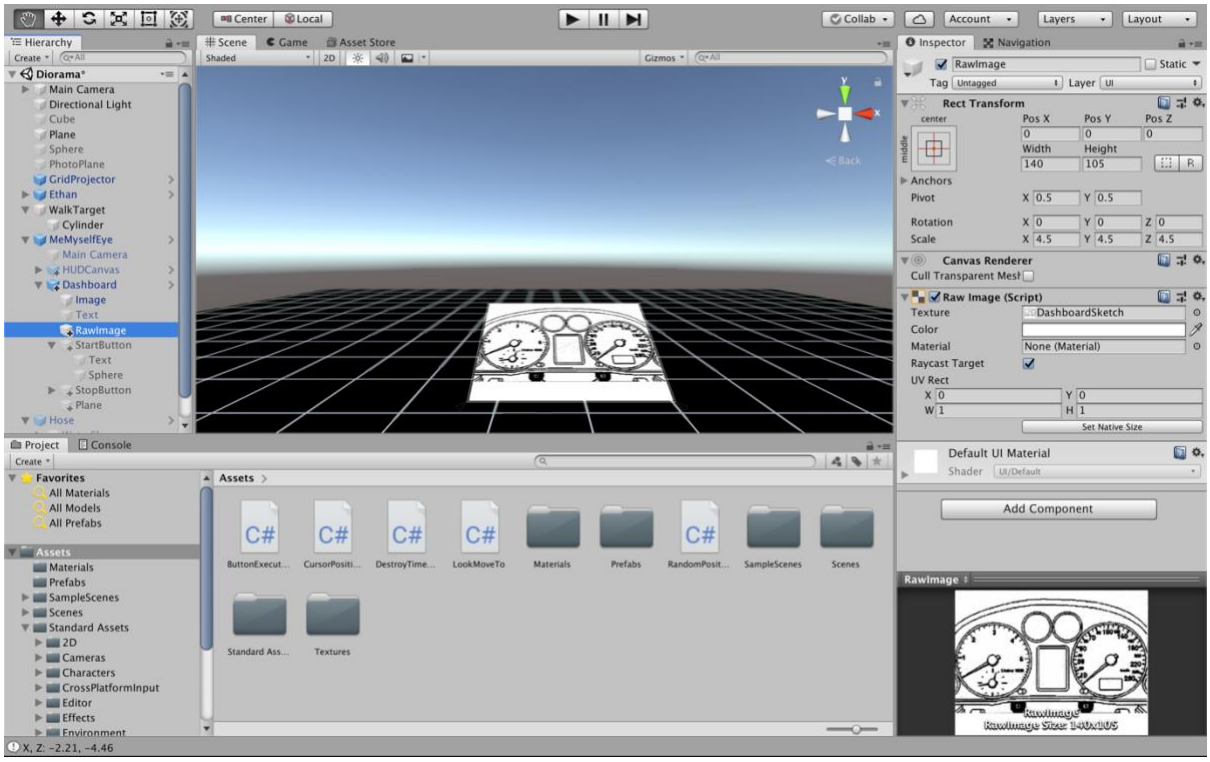


6) HUD 캔버스가 게임이 시작되고 15 초 후에 사라지는 것을 확인할 수 있다.

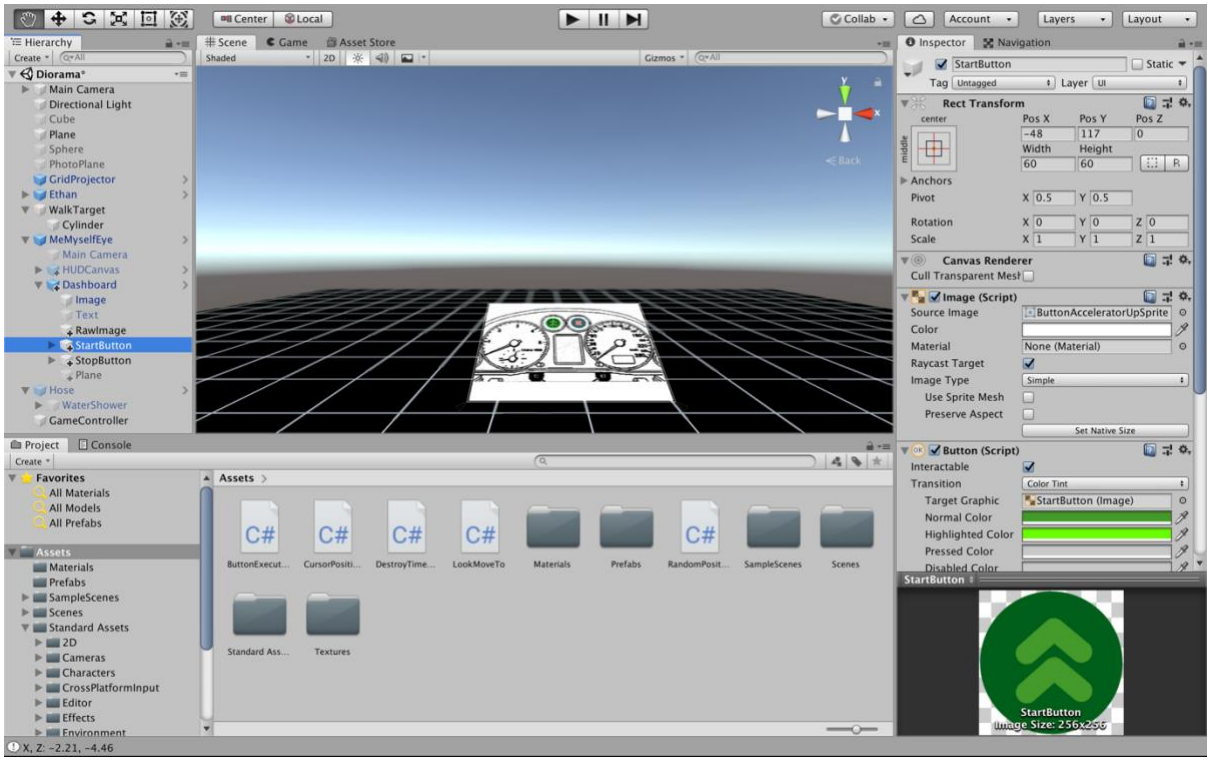


2. 인게임 대시보드

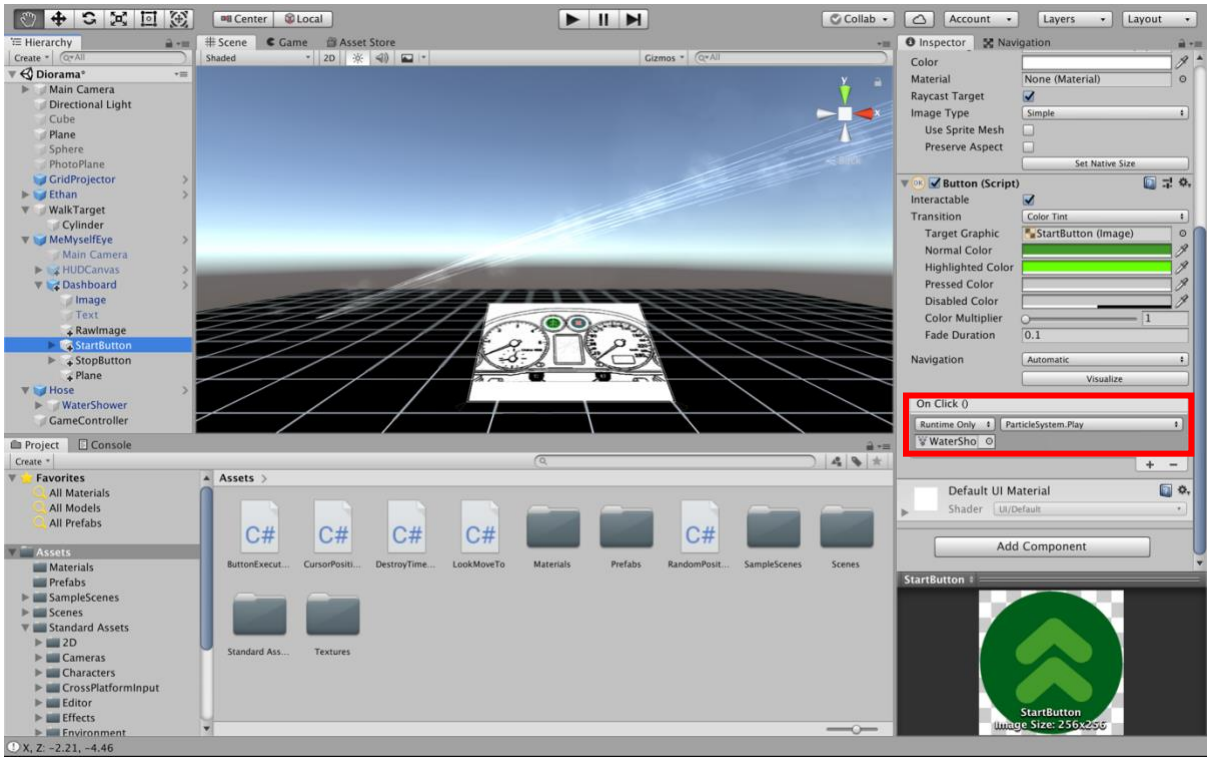
1) 자동차 대시보드 이미지 스케치 생성하기

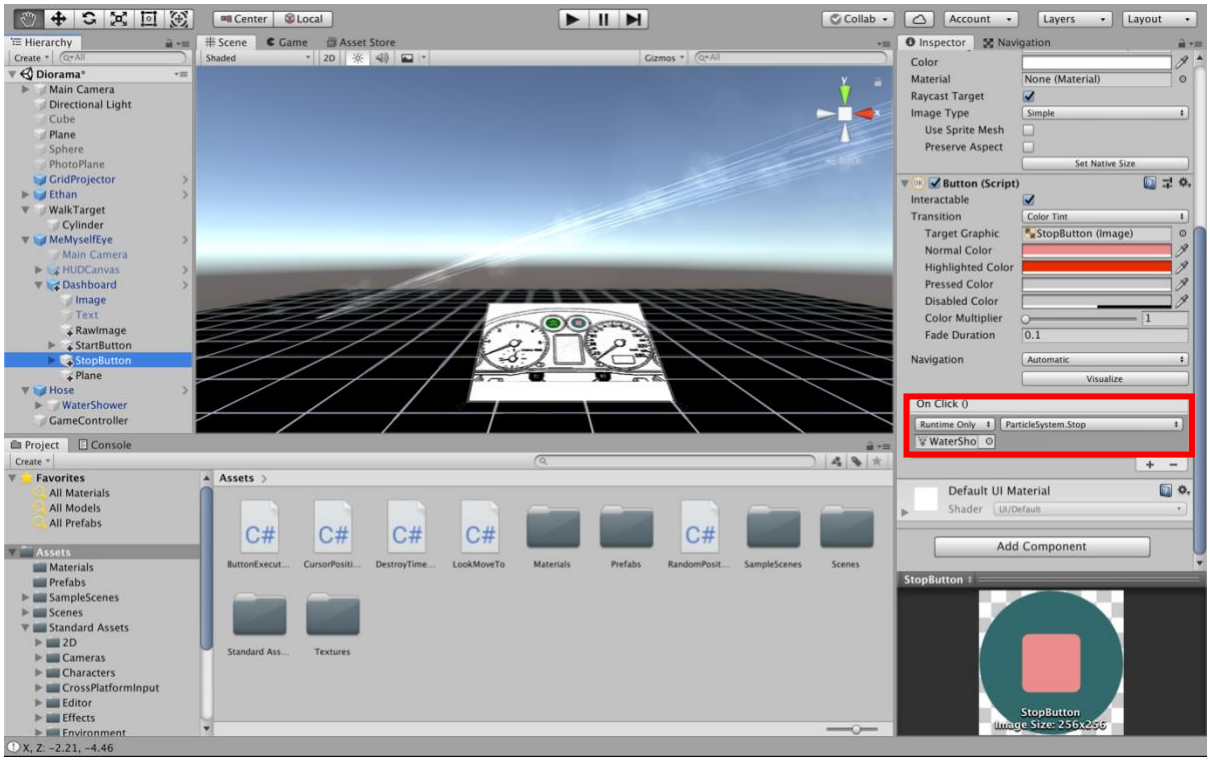


3) start 와 stop 버튼 추가



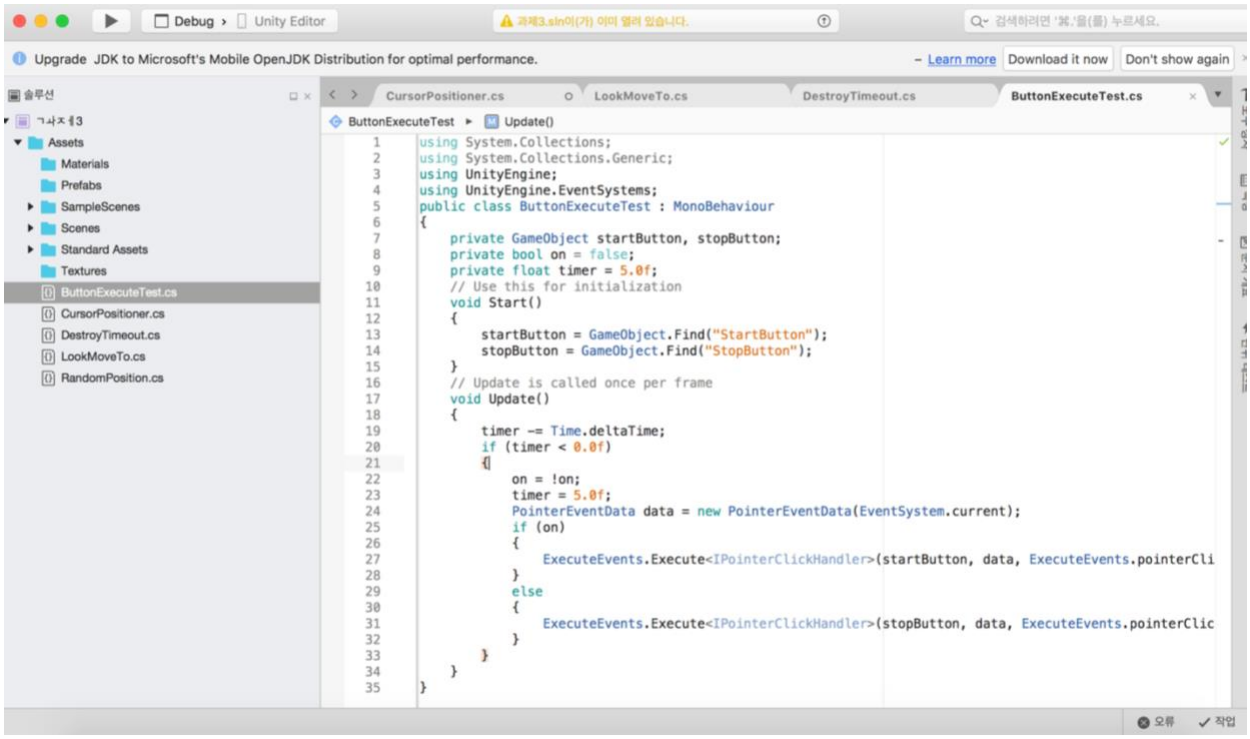
4) 물 호스와 버튼 연결


Hierarchy
Diorama*
Main Camera
Directional Light
Cube
Plane
Sphere
PhotoPlane
GridProjector
Ethan
WalkTarget
Cylinder
MeMyselfEye
Main Camera
HUDCanvas
Dashboard
Image
Text
RawImage
StartButton
StopButton
Plane
Hose
WaterShower
GameController
Scene
Game
Asset Store
Inspector
Navigation
Color
Material
None (Material)
Raycast Target
Image Type
Simple
Use Sprite Mesh
Preserve Aspect
Set Native Size
Button (Script)
Interactable
Transition
Color Tint
Target Graphic
StartButton (Image)
Normal Color
Highlighted Color
Pressed Color
Disabled Color
Color Multiplier
Fade Duration
Navigation
Automatic
Visualize
On Click ()
Runtime Only
ParticleSystem.Play
Default UI Material
Shader
Add Component
StartButton
Image Size: 256x256
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Materials
Prefabs
SampleScenes
Scenes
Standard Assets
2D
Cameras
Characters
CrossPlatformInput
Editor
Effects
Environment
ButtonExecut...
CursorPositi...
DestroyTime...
LookMoveTo
RandomPosit...
Standard Ass...
Textures
X, Z: -2.21, -4.46


Hierarchy
Diorama*
Main Camera
Directional Light
Cube
Plane
Sphere
PhotoPlane
GridProjector
Ethan
WalkTarget
Cylinder
MeMyselfEye
Main Camera
HUDCanvas
Dashboard
Image
Text
RawImage
StartButton
StopButton
Plane
Hose
WaterShower
GameController
Scene
Game
Asset Store
Inspector
Navigation
Color
Material
None (Material)
Raycast Target
Image Type
Simple
Use Sprite Mesh
Preserve Aspect
Set Native Size
Button (Script)
Interactable
Transition
Color Tint
Target Graphic
StopButton (Image)
Normal Color
Highlighted Color
Pressed Color
Disabled Color
Color Multiplier
Fade Duration
Navigation
Automatic
Visualize
On Click ()
Runtime Only
ParticleSystem.Stop
Default UI Material
Shader
Add Component
StopButton
Image Size: 256x256
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Materials
Prefabs
SampleScenes
Scenes
Standard Assets
2D
Cameras
Characters
CrossPlatformInput
Editor
Effects
Environment
ButtonExecut...
CursorPositi...
DestroyTime...
LookMoveTo
RandomPosit...
Standard Ass...
Textures
X, Z: -2.21, -4.46

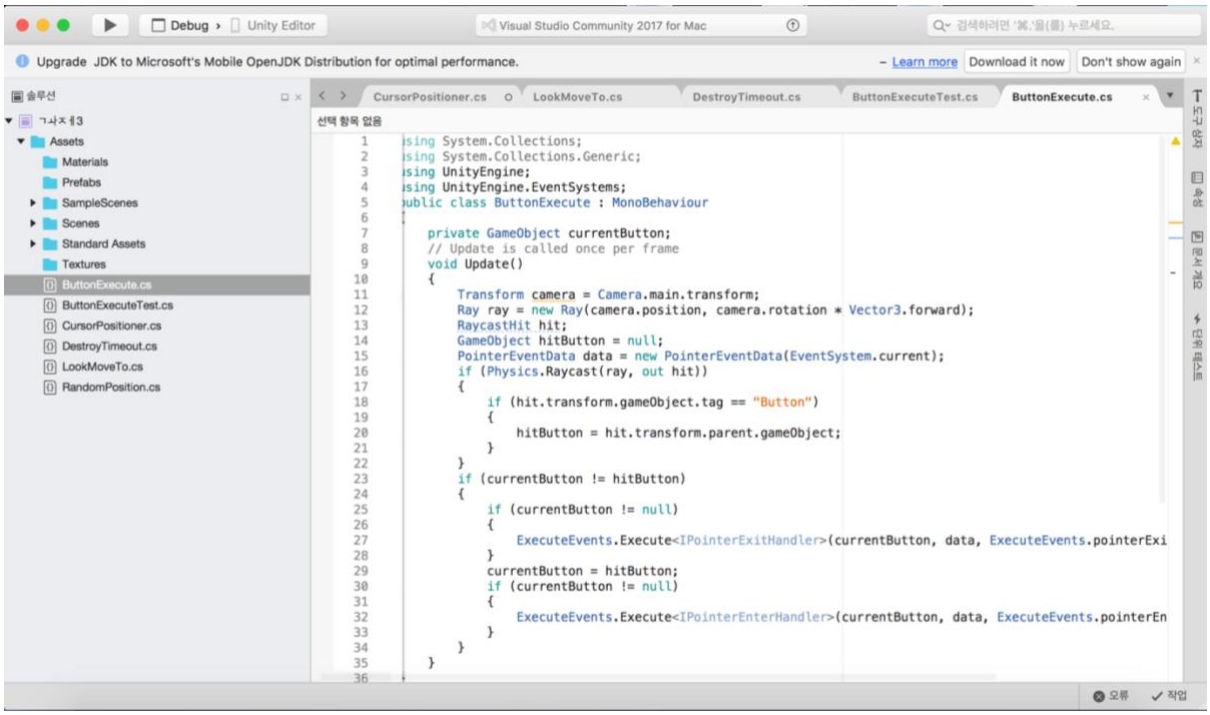
5) 스크립트에서 버튼 동작시키기

– GameController 오브젝트에 ButtonExecuteTest 스크립트를 만든 후 작성(호스가 5 초마다 켜자고 꺼지게 함)



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
public class ButtonExecuteTest : MonoBehaviour
{
    private GameObject startButton, stopButton;
    private bool on = false;
    private float timer = 5.0f;
    // Use this for initialization
    void Start()
    {
        startButton = GameObject.Find("StartButton");
        stopButton = GameObject.Find("StopButton");
    }
    // Update is called once per frame
    void Update()
    {
        timer -= Time.deltaTime;
        if (timer < 0.0f)
        {
            on = !on;
            timer = 5.0f;
            PointerEventData data = new PointerEventData(EventSystem.current);
            if (on)
            {
                ExecuteEvents.Execute<IPointerClickHandler>(startButton, data, ExecuteEvents.pointerCli
            }
            else
            {
                ExecuteEvents.Execute<IPointerClickHandler>(stopButton, data, ExecuteEvents.pointerClic
            }
        }
    }
}
```

6) 버튼 강조하기



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;
public class ButtonExecute : MonoBehaviour
{
    private GameObject currentButton;
    // Update is called once per frame
    void Update()
    {
        Transform camera = Camera.main.transform;
        Ray ray = new Ray(camera.position, camera.rotation * Vector3.forward);
        RaycastHit hit;
        GameObject hitButton = null;
        PointerEventData data = new PointerEventData(EventSystem.current);
        if (Physics.Raycast(ray, out hit))
        {
            if (hit.transform.gameObject.tag == "Button")
            {
                hitButton = hit.transform.parent.gameObject;
            }
        }
        if (currentButton != hitButton)
        {
            if (currentButton != null)
            {
                ExecuteEvents.Execute<IPointerExitHandler>(currentButton, data, ExecuteEvents.pointerExi
            }
            currentButton = hitButton;
            if (currentButton != null)
            {
                ExecuteEvents.Execute<IPointerEnterHandler>(currentButton, data, ExecuteEvents.pointerEn
            }
        }
    }
```