

Unity3D Script – UI

Team Project

UI 스크립트 튜토리얼

기초적인 설명은 유니티 정식 홈페이지에 존재

<https://unity3d.com/kr/learn/tutorials/s/user-interface-ui>

UI 개요

- canvas: 모든 UI는 Canvas 내에 소속됨
- 모든 UI에는 해당 UI와 특정 스크립트를 연결시킬 수 있음
- 4개의 핵심 UI 모듈에 대한 사용법

- 1) Button
- 2) Check Box
- 3) Radio Box
- 4) Slider

수업 코드

- GameManger를 만들고
이곳에 4개의 UI별로 수
행하는 업무들을 함수로
개별 정의

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

0 references

```
public class Button_Script : MonoBehaviour {
```

5 references

```
public static bool bCheck_Parameter = true;
```

4 references

```
public static int bHardware = 0;
```

1 reference

```
[SerializeField] private InputField Name_Input;
```

1 reference

```
[SerializeField] private InputField Password_Input;
```

1 reference

```
[SerializeField] private Slider Slider_Value;
```

```
// Use this for initialization
```

0 references

```
void Start () {
```

```
}
```

```
// Update is called once per frame
```

0 references

```
void Update () {
```

```
}
```

0 references

```
public void MyButtonScript(){
```

```
    Debug.Log("My Button");
```

```
}
```

0 references

```
public void MyCheckButtonScript()
```

```
{
```

```
    if (bCheck_Parameter == true)
```

```
    {
```

```
        bCheck_Parameter = false;
```

```
    }else if (bCheck_Parameter == false) {
```

```
        bCheck_Parameter = true;
```

```
    }
```

```
    Debug.Log("My check button:" + bCheck_Parameter);
```

```
}
```

0 references

```
public void MyRadioButtonScript1()
```

```
{
```

```
    //opengl 선택
```

```
    bHardware = 1;
```

```
    Debug.Log("오픈지엘을 선택했습니다:" + bHardware);
```

```
}
```

0 references

```
public void MyRadioButtonScript2()
```

```
{
```

```
    //directx 선택
```

```
    bHardware = 2;
```

```
    Debug.Log("다이렉트X를 선택했습니다:" + bHardware);
```

```
}
```

0 references

```
public void MyInputField_Name()
```

```
{
```

```
    Debug.Log("이름을 입력했습니다.");
```

```
    if (Name_Input.text == "Kang")
```

```
    {
```

```
        Debug.Log("존재하는 ID 입니다");
```

```
    }
```

```
}
```

0 references

```
public void MyInputField_Password()
```

```
{
```

```
    Debug.Log("암호를 입력했습니다.");
```

```
    if (Password_Input.text == "1111")
```

```
    {
```

```
        Debug.Log("암호가 일치합니다.");
```

```
    }
```

```
}
```

0 references

```
public void MySlider(){
```

```
    Debug.Log("슬라이더값:" + Slider_Value.value);
```

```
}
```

```
}
```

UI Asset 다운로드

애셋스토어에서 Flat UI 받기

Sale Ends Soon • See the deals

unityAssetStore

Search for assets

SK

Home > Tools > GUI > Flat UI

You purchased this item on May 12, 2019.

Flat UI!

ks News About Us Contact Us Exit

Button

Button ☒ Checkbox ☐ Radio

Subscribe

Enter text...

Enter Email

Subscribe

My Unity Dream

MYUNITYDREAM

Flat UI

FREE

★★★★☆

Import

Popular Tags

Add a new tag right now?

Add tags

Flat UI Is For You.
you can use Flat UI For Your Unity Projects.
This UI Is New UI System in Unity 3d 4.6.

Instructions:
1- Import Flat UI In Unity3d
2- Go To Prefab Directory And Drag Your Choose Into Scene View
Enjoy...
By Thanks
My Unity Dream

Package contents

255.5 KB

Total file size255.5 KB

Number of files16

Package Preview

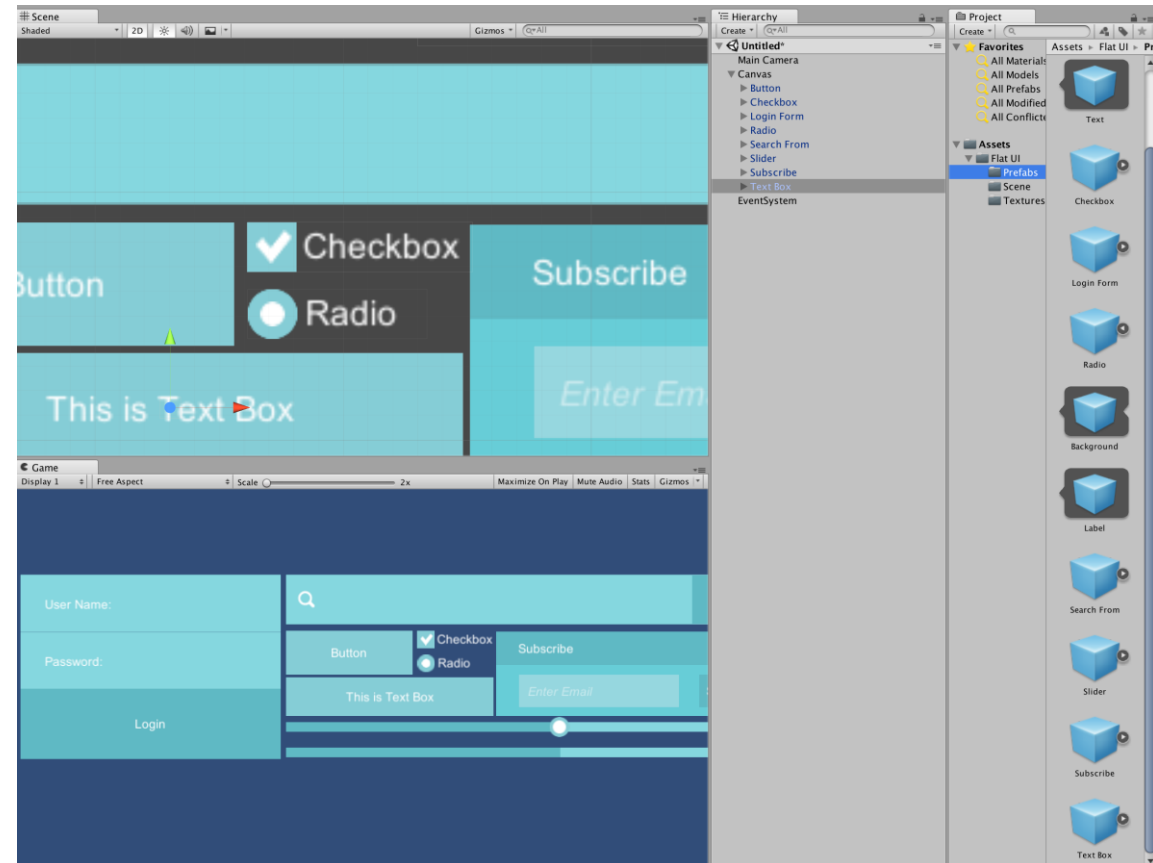
Share

Add to List

Report this asset

UI Asset 다운로드

Canvas를 만들고 사전에 만들어져 있는 Prefab를 Canvas에 드래그앤 드롭
(그냥 드래그앤 드롭하면 사용 불가)



Button 스크립트짜기

<방법1, 개별 스크립트 관리>

- Button_Script 만들고 임의의 public 함수 만들기 (Public을 안 붙이면 외부에서 접근 불가)

- Button_Script 컴포넌트를 Button 객체에 컴포넌트로 붙이기

Button 객체에서 On_Click → runtime_only 밑에 자기 자신을 드래그해서 붙이고,

No Function을 눌러보면 방금 붙인 스크립트명 (Button_Script)가 나오고, 거기에 등록되어 있는 public 함수를 선택할 수 있음

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

public class Button_Script : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

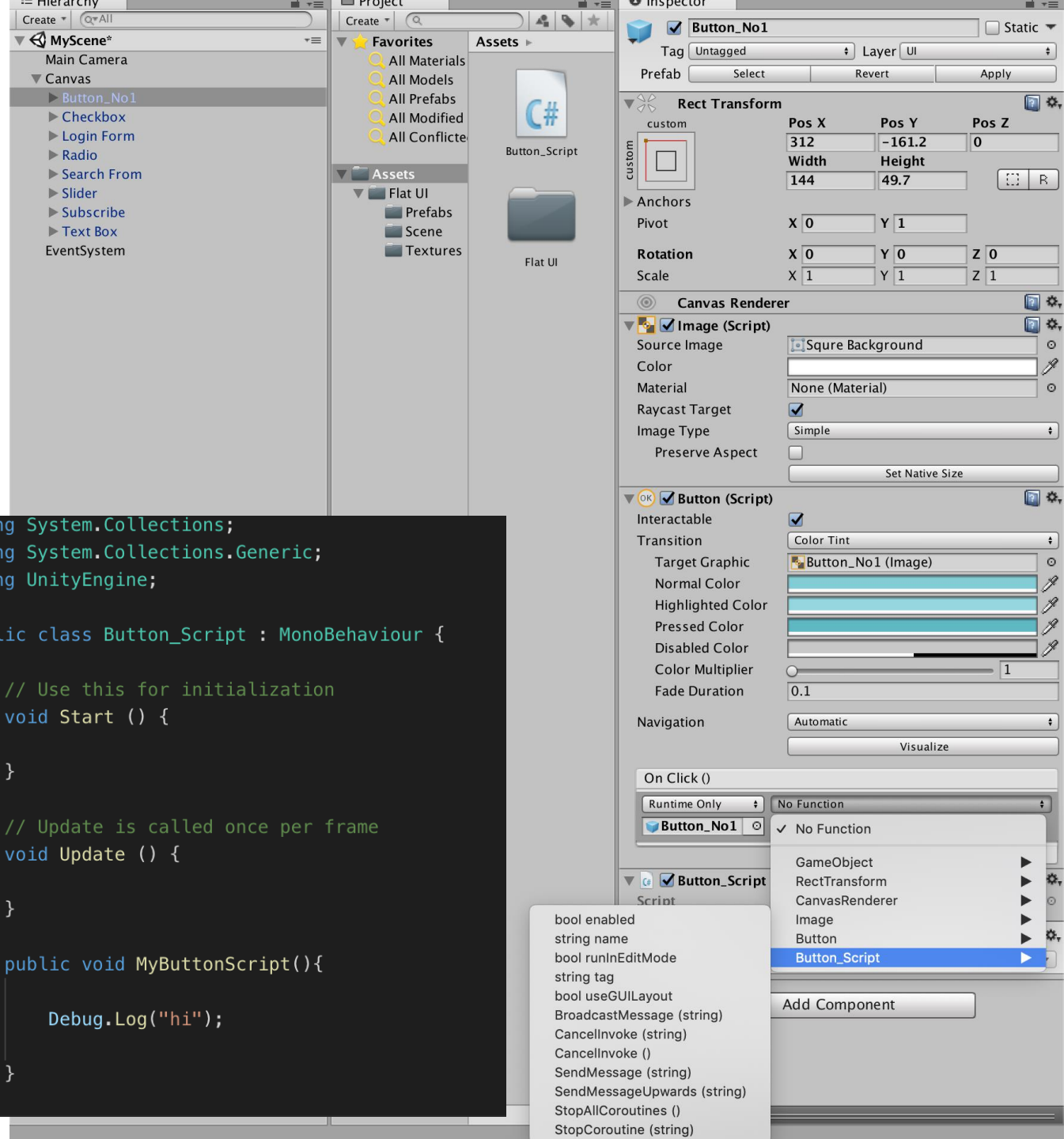
    }

    public void MyButtonScript(){

        Debug.Log("hi");

    }

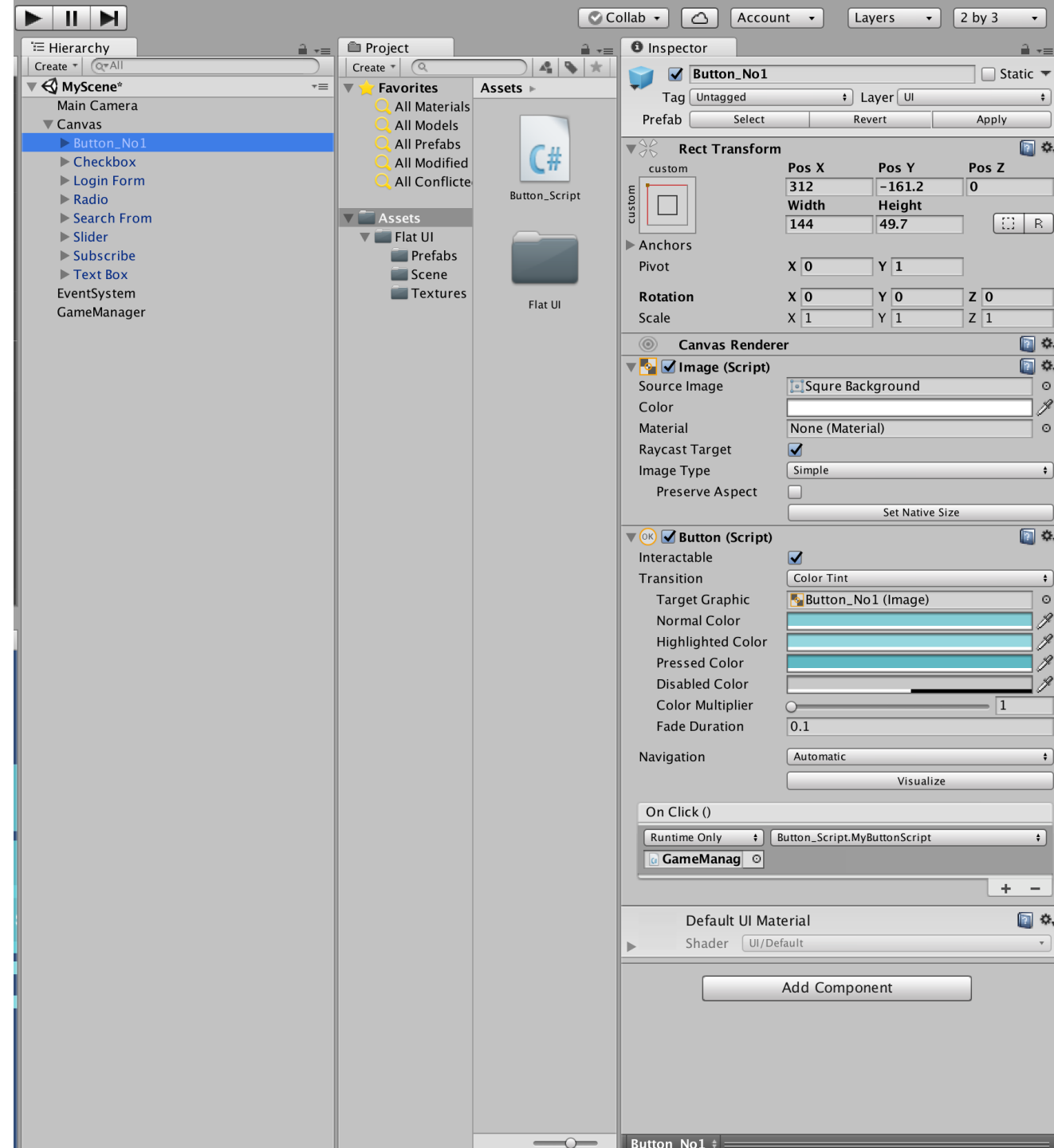
}
```



Button 스크립트짜기

<방법2, 중앙 스크립트 관리>

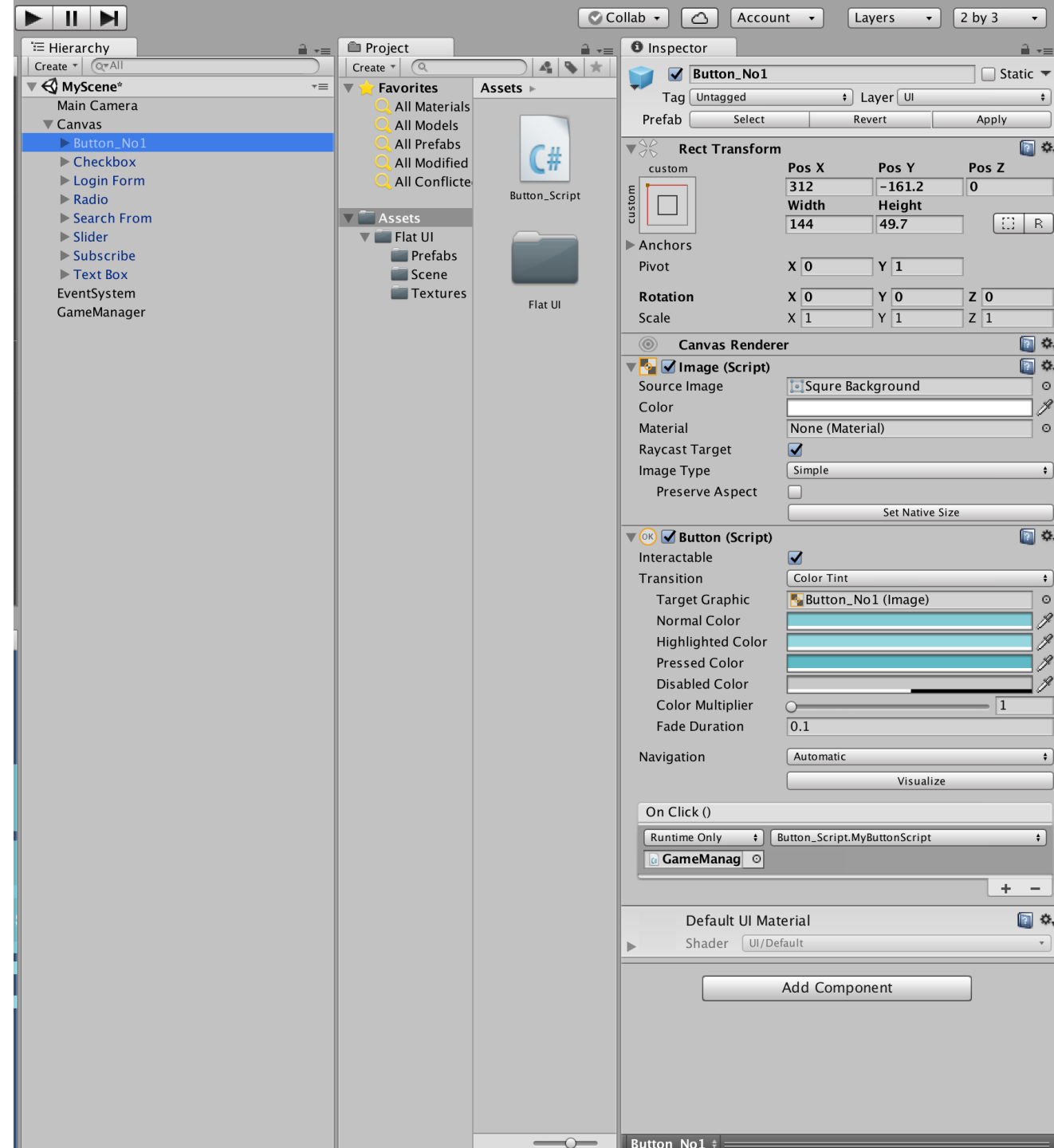
- GameManager를 만들고 그 곳에 스크립트를 부착한다.
- Button 객체에서 위의 함수를 등록해서 사용



Button 스크립트짜기

<방법2, 중앙 스크립트 관리>

- GameManager를 만들고 그 곳에 스크립트를 부착한다.
- Button 객체에서 위의 함수를 등록해서 사용



Checkbox 스크립트짜기

- GameManager에 부착된 Button_Script 를 업데이트
- Checkbox에 업데이트된 함수 등록

```
public static bool bCheck_Parameter = true;

// Use this for initialization
void Start () {

}

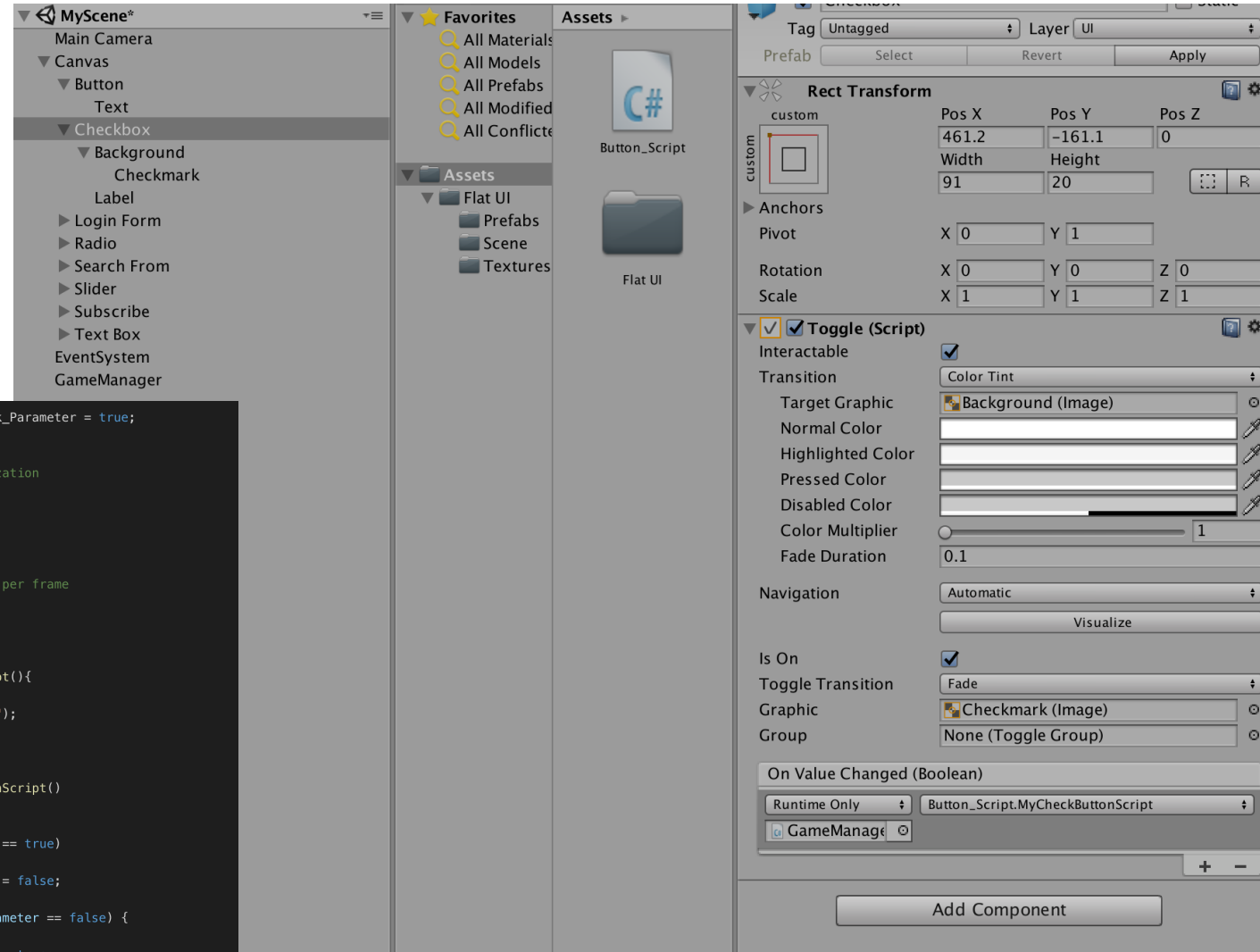
// Update is called once per frame
void Update () {

}

public void MyButtonScript(){
    Debug.Log("My Button");
}

public void MyCheckButtonScript()
{
    if (bCheck_Parameter == true)
    {
        bCheck_Parameter = false;
    }else if (bCheck_Parameter == false) {
        bCheck_Parameter = true;
    }

    Debug.Log("My check button:" + bCheck_Parameter);
}
```



Checkbox 스크립트짜기

- GameManager에 부착된 Button_Script 를 업데이트
- Checkbox에 업데이트된 함수 등록

```
public static bool bCheck_Parameter = true;

// Use this for initialization
void Start () {

}

// Update is called once per frame
void Update () {

}

public void MyButtonScript(){

    Debug.Log("My Button");

}

public void MyCheckButtonScript()
{

    if (bCheck_Parameter == true)
    {
        bCheck_Parameter = false;
    }else if (bCheck_Parameter == false) {

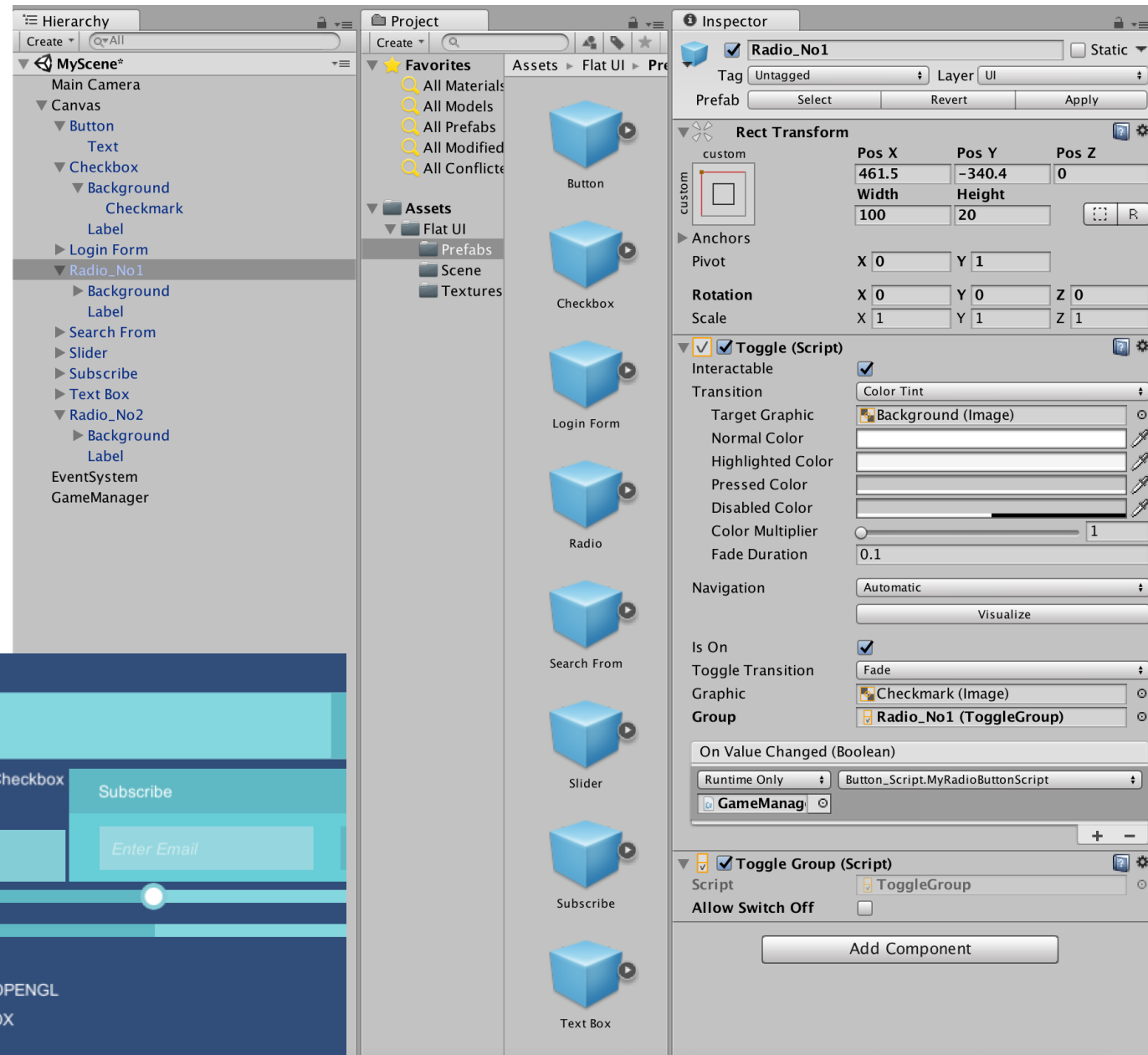
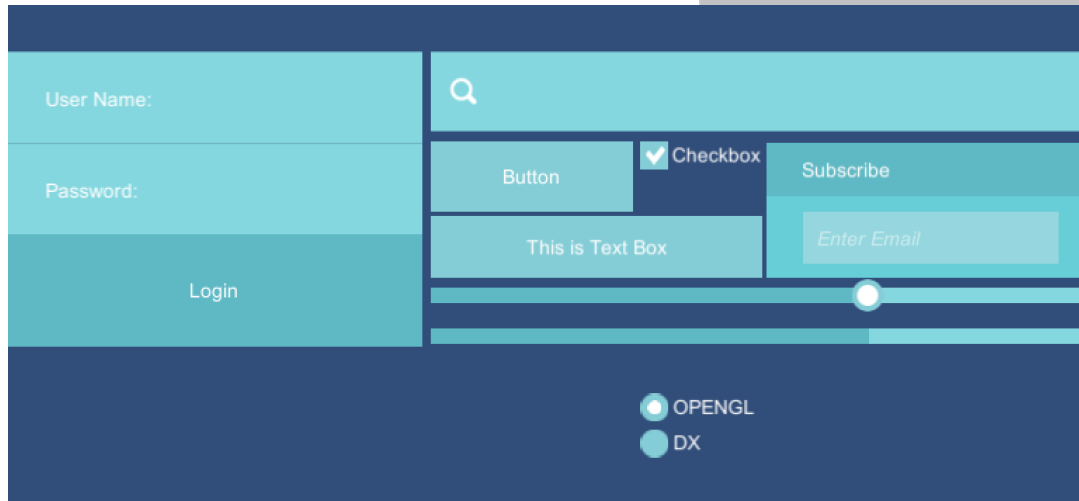
        bCheck_Parameter = true;
    }

    Debug.Log("My check button:" + bCheck_Parameter);

}
```

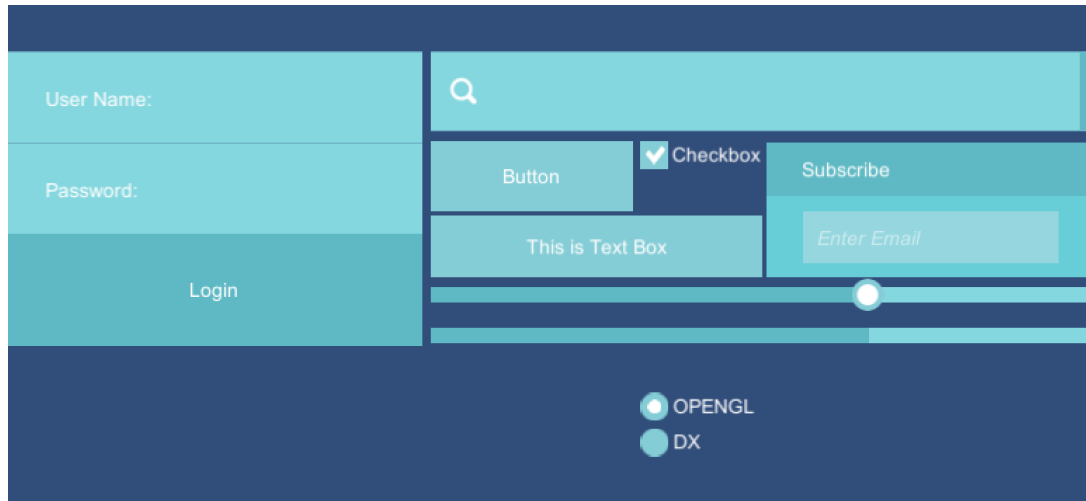
radio 버튼 그룹 만들기

- radio 버튼을 2개 만들고,
- 이 중 하나의 Radio 버튼 토글 그룹 컴포넌트를 붙인다.
- 그리고 해당 토글 그룹을 2개의 radio 버튼에서 선택한다.
- <https://undercode.tistory.com/13>
- 둘 중 하나만 is on 시킨다



radio 버튼 스크립트짜기

- 하드웨어 전역 변수를 만들어서 각각의 라디오버튼을 누르면 해당 값으로 셋팅하는 코드



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

public class Button_Script : MonoBehaviour {

    public static bool bCheck_Parameter = true;
    public static int bHardware = 0;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public void MyButtonScript(){

        Debug.Log("My Button");

    }

    public void MyCheckButtonScript()
    {

        if (bCheck_Parameter == true)
        {
            bCheck_Parameter = false;
        }else if (bCheck_Parameter == false) {

            bCheck_Parameter = true;
        }

        Debug.Log("My check button:" + bCheck_Parameter);

    }

    public void MyRadioButtonScript1()
    {

        //opengl 선택
        bHardware = 1;
        Debug.Log("오픈지엘을 선택했습니다:" + bHardware);

    }

    public void MyRadioButtonScript2()
    {

        //directx 선택
        bHardware = 2;
        Debug.Log("다이렉트x를 선택했습니다:" + bHardware);

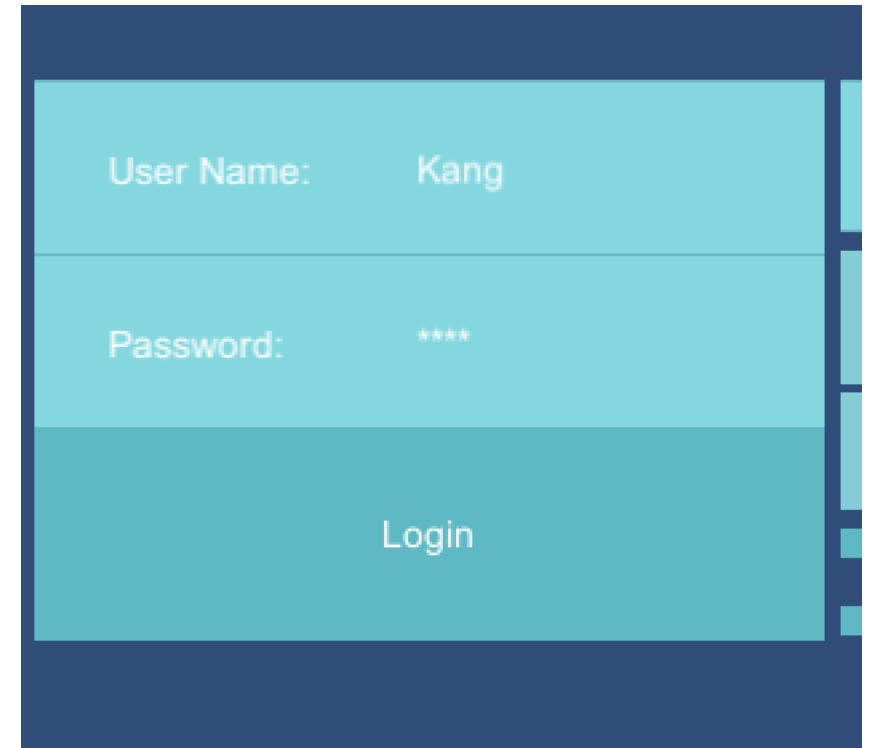
    }

}
```

Input Field

- 사용자로부터 직접 값을 얻어오는 UI
- 로그인 시 주로 사용
- 형변환에 주의
- 참조 영상

<https://youtu.be/ZOTZOvRgFmY>



User Name: Kang

Password: ****

Login

Input Field

- using UnityEngine.UI 추가
- 입력값을 받아올 2개의 public 필드값 정의

[SerializeField] private
InputField Name_Input;

[SerializeField] private
InputField Password_Input;

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

0 references
public class Button_Script : MonoBehaviour {

    5 references
    public static bool bCheck_Parameter = true;

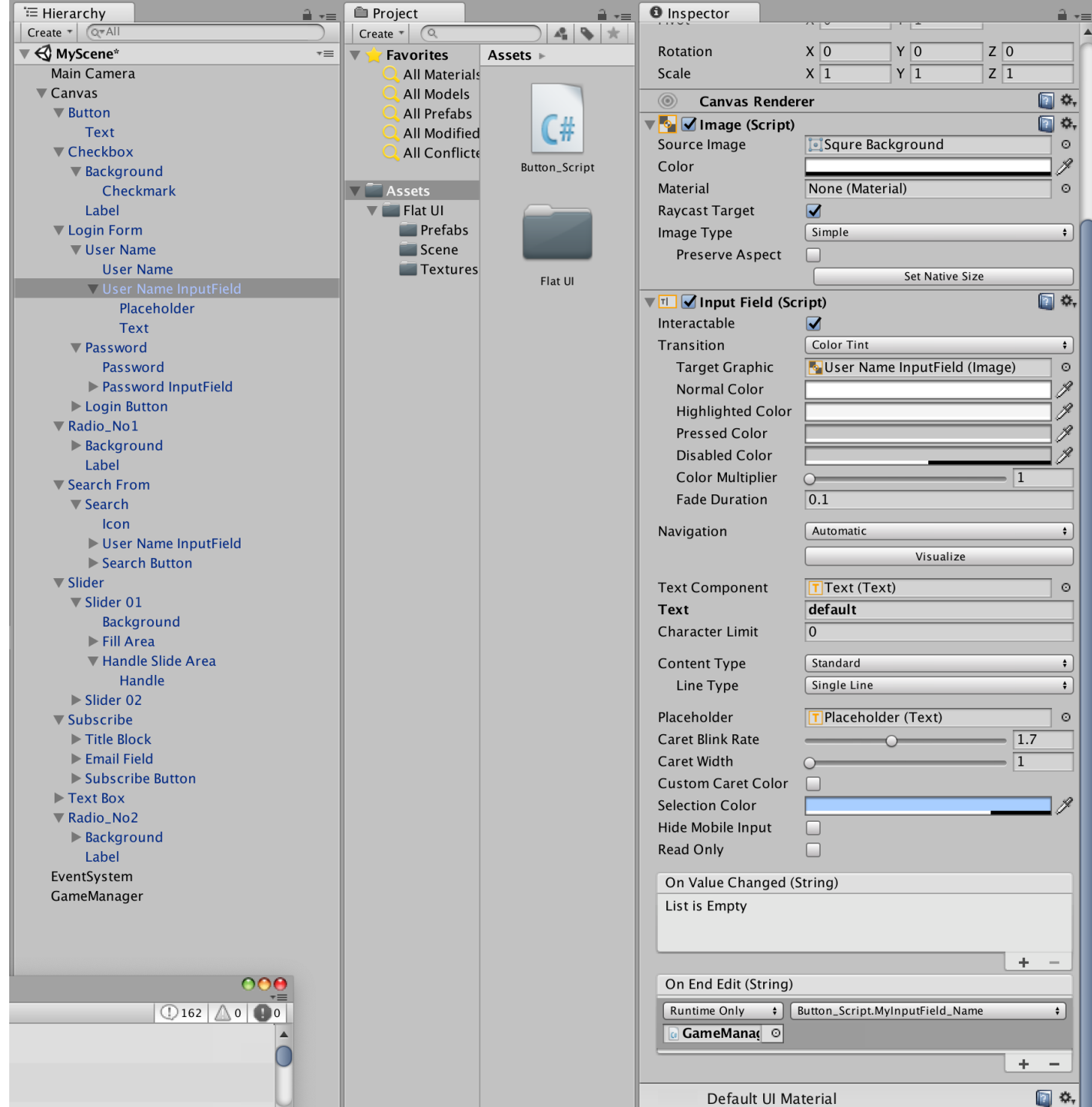
    4 references
    public static int bHardware = 0;

    1 reference
    [SerializeField] private InputField Name_Input;

    0 references
    [SerializeField] private InputField Password_Input;
```

Input Field

- 이벤트 추가



Input Field

- 각각의 인풋 필드에서 입력이 끝날 때마다 정확히 입력한 것인지 체크하는 코드 추가

0 references

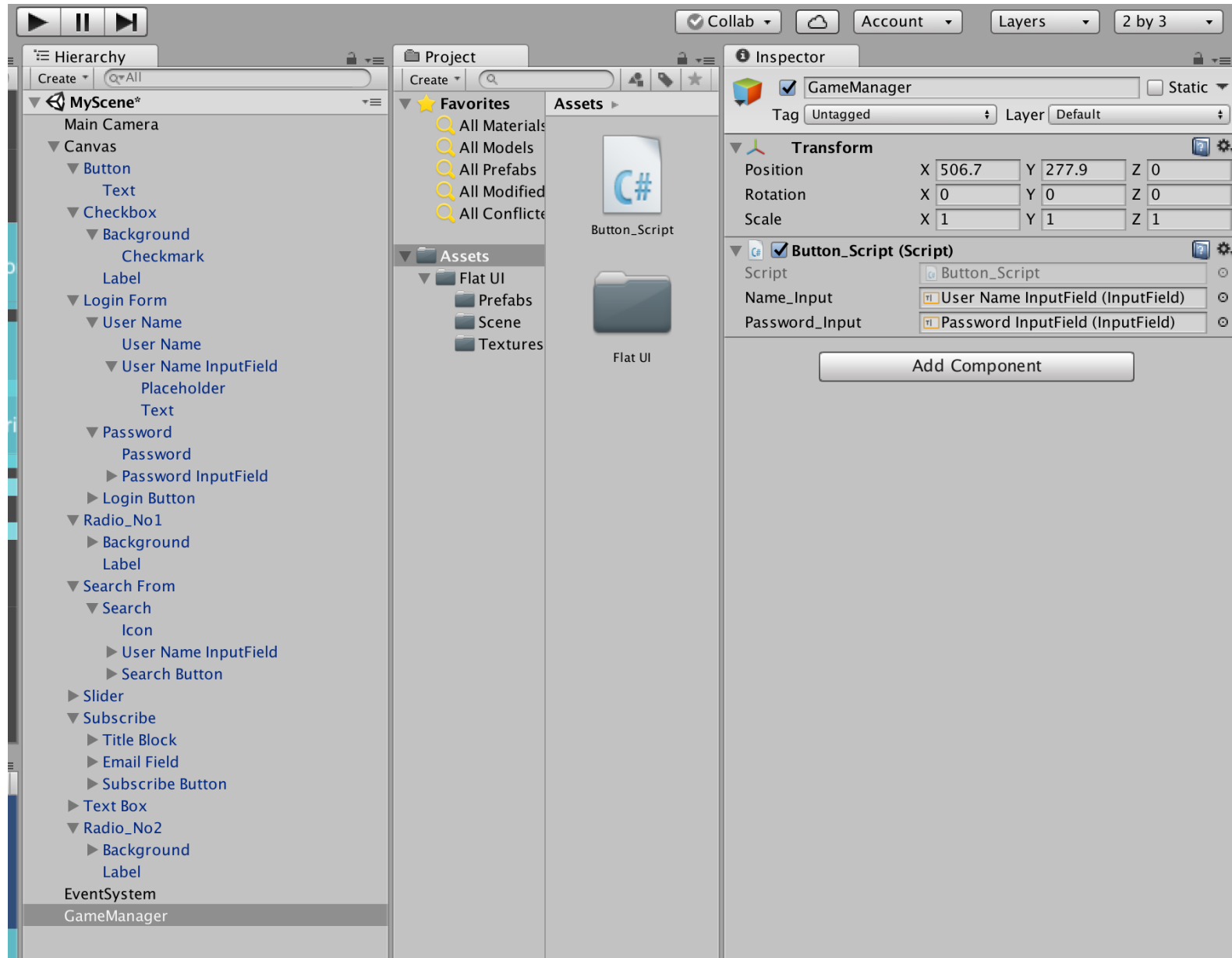
```
public void MyInputField_Name()
{
    Debug.Log("이름을 입력했습니다.");
    if (Name_Input.text == "Kang")
    {
        Debug.Log("존재하는 ID 입니다");
    }
}
```

0 references

```
public void MyInputField_Password()
{
    Debug.Log("암호를 입력했습니다.");
    if (Password_Input.text == "1111")
    {
        Debug.Log("암호가 일치합니다.");
    }
}
```

Input Field

- public으로 선언한 2개의 인풋필드를 UI 인풋필드와 연결하기



Slider

- 연속된 값중 하나를 선택
- 참조 영상

<https://keidy.tistory.com/318?category=775337>

Slider

- 입력값을 받아올 1개의 public 필드값 정의

[SerializeField] private
Slider Slider_Value;

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

0 references
public class Button_Script : MonoBehaviour {

    5 references
    public static bool bCheck_Parameter = true;

    4 references
    public static int bHardware = 0;

    1 reference
    [SerializeField] private InputField Name_Input;

    1 reference
    [SerializeField] private InputField Password_Input;

    1 reference
    [SerializeField] private Slider Slider_Value;
```

Slider

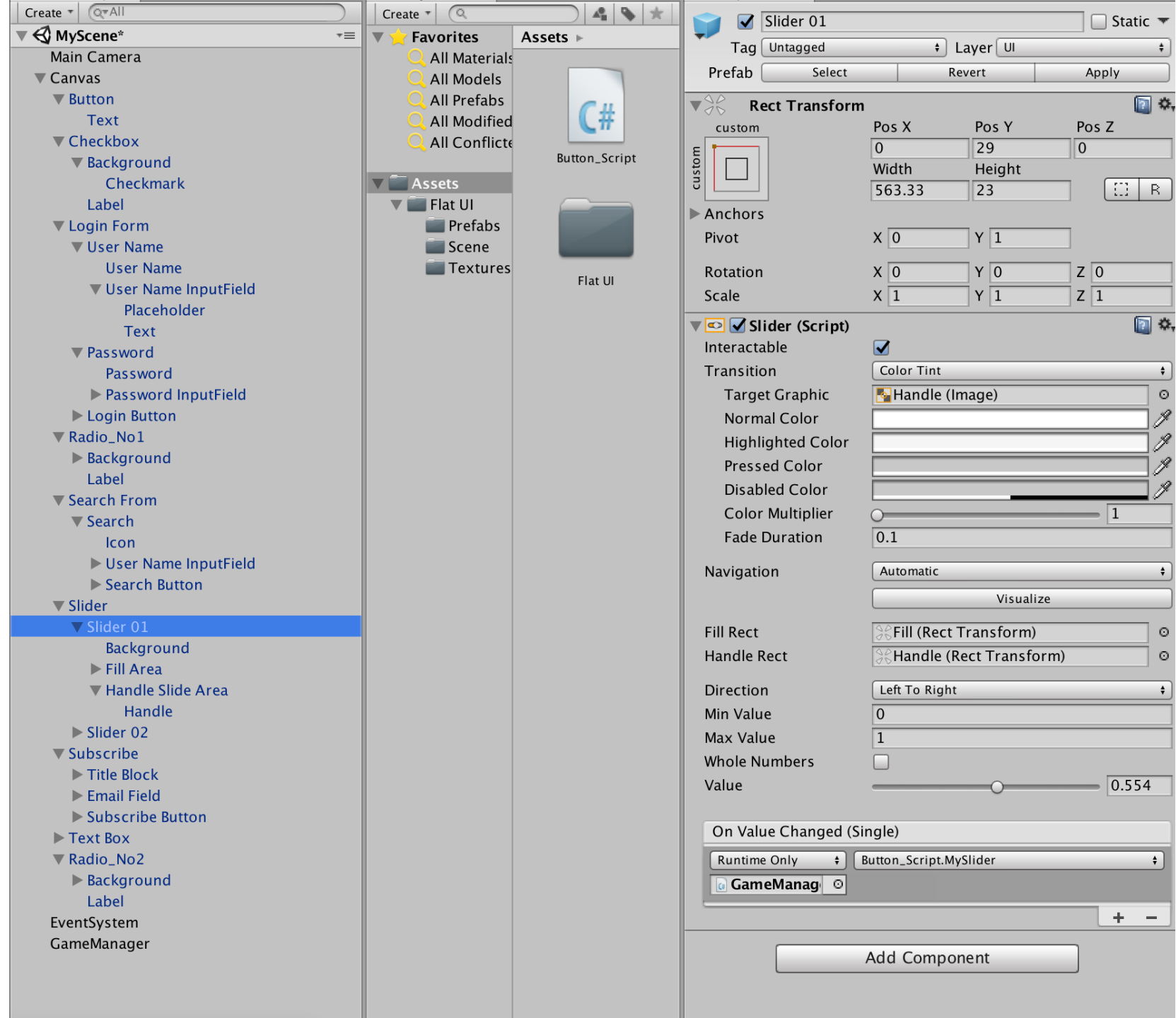
- 슬라이더 값 변환 시 수행할 코드 추가

0 references

```
public void MySlider(){  
    Debug.Log("슬라이더값:" + Slider_Value.value);  
}
```

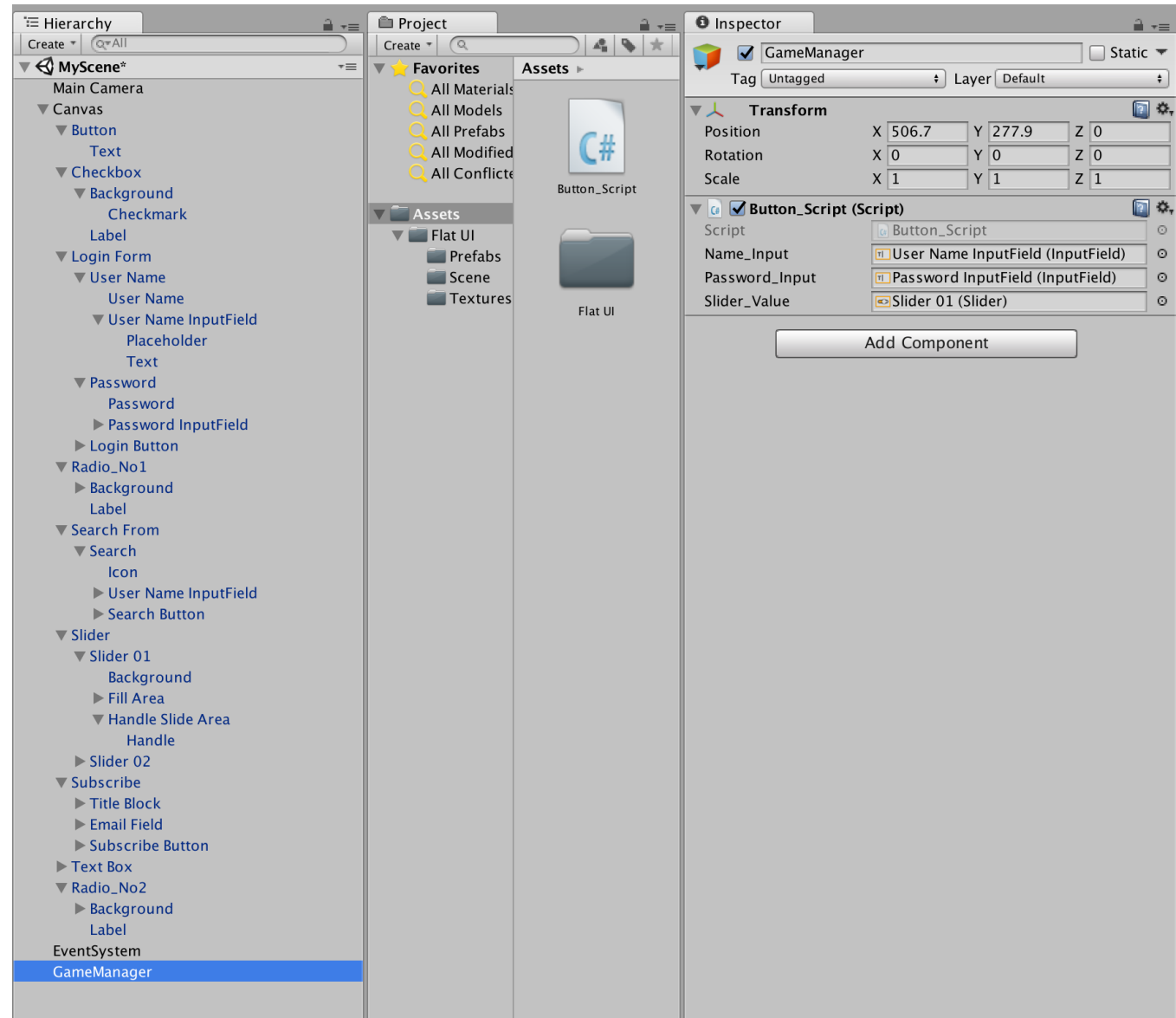
Slider

- 이벤트 추가



Slider

- GameManager 에서 Public으로 노출된 slider 배정



퀴즈

- 자신이 만들려는 게임의 메인 UI 화면을 만들어보세요.

