

Tạo setting game trên inspector.

Chào các bạn,

Hôm nay chúng ta sẽ cùng tạo 1 setting/config cho game trên inspector nhé!

Vậy tại sao lại là inspector? và khi nào thì chúng ta nên tạo mấy cái kiểu thế này?



Câu trả lời đó là khi chúng ta cần 1 cái gì đó để thao tác trong editor, trực quan nhất, không cần sửa code để cập nhật dữ liệu.... Và hẳn các bạn cũng từng làm việc với những value trên inspector rồi, ví dụ khi ta khai báo 1 biến dạng publish, chúng ta sẽ thấy nó hiển thị trên inspector và có thể edit được nó khi đang play.

Nếu chỉ cần khai báo publish là được thì sao phải có bài này? Đó là vì publish là không đủ cho nhiều nhu cầu, ví dụ chúng ta muốn hiển thị 1 nút ẩn trên inspector thì publish sẽ không làm được.

Ok, vậy vào việc nhé!

Mục tiêu của ta sẽ là tạo ra 1 dropdown, một vài trường dữ liệu, và 1 nút ẩn.

Bước 1: Tạo class.

Chúng ta cần 2 class, 1 cái là định nghĩa các phương thức và thuật toán, 1 class để hiển thị lên inspector, các bạn đặt tên thêm hậu tố Editor và implements từ Class Editor.

Ví dụ mình đặt tên 2 class là Settings và SettingsEditor.

Bước 2: Khai báo biến và phương thức.

Chúng ta tạo ra 1 biến dạng enum để làm dropdown, 1 biến dạng float có RangeAttribute, 1 biến bool làm checkbox, và 1 biến GameObject.

Phương thức chúng ta tạo 1 phương thức DisplayValue để show những value chúng ta vừa sửa trên Inspector.

```
SettingsEditor.cs Settings.cs X
Assembly-CSharp
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 @ Unity Script | 0 references
6 public class Settings : MonoBehaviour
7 {
8     1 reference
9     public enum Dropdown { WhiteMode, DarkMode}
10    public Dropdown mode;
11
12    [Range(0, 1)]
13    public float alphaBackground;
14
15    public bool isDebug;
16
17    public GameObject objectBoss;
18
19    0 references
20    public void DisplayValue()
21    {
22        Debug.Log($"Mode : {mode.ToString()}");
23        Debug.Log($"Alpha Background : {alphaBackground}");
24        Debug.Log($"Is Debug : {isDebug}");
25        Debug.Log($"Boss Name : {objectBoss.name}");
26    }
27 }
```

Bước 3: Định nghĩa SettingsEditor.

Khai báo lại các biến với thuộc tính là SerializedProperty.

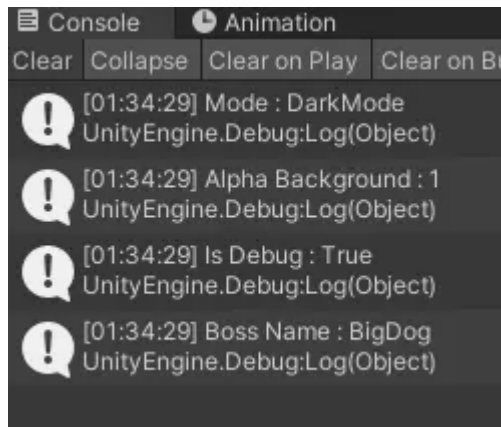
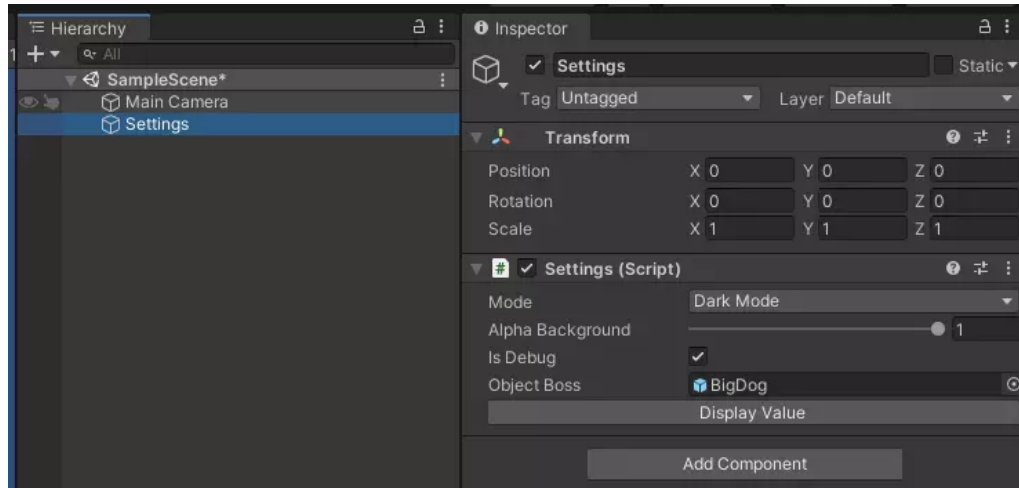
Trong phương thức OnEnable chúng ta sẽ tìm các property của các biến và gán cho những biến vừa tạo.

Tiếp theo chúng ta override phương thức OnInspectorGUI để hiển thị tất cả những biến và nút ở trên.

```
SettingsEditor.cs X
Assembly-CSharp SettingsEditor OnInspectorGUI()
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEditor;
5
6 [CustomEditor(typeof(Settings))]
7 @ Unity Script | 0 references
8 public class SettingsEditor : Editor
9 {
10    public SerializedProperty
11    mode,
12    alphaBackground,
13    isDebug,
14    objectBoss;
15
16    @ Unity Message | 0 references
17    private void OnEnable()
18    {
19        mode = serializedObject.FindProperty("mode");
20        alphaBackground = serializedObject.FindProperty("alphaBackground");
21        isDebug = serializedObject.FindProperty("isDebug");
22        objectBoss = serializedObject.FindProperty("objectBoss");
23    }
24
25    0 references
26    public override void OnInspectorGUI()
27    {
28        Settings mySettings = (Settings)target;
29
30        serializedObject.Update();
31        EditorGUILayout.PropertyField(mode);
32        EditorGUILayout.PropertyField(alphaBackground);
33        EditorGUILayout.PropertyField(isDebug);
34        EditorGUILayout.PropertyField(objectBoss);
35
36        if(GUILayout.Button("Display Value"))
37        {
38            mySettings.DisplayValue();
39        }
40
41        serializedObject.ApplyModifiedProperties();
42    }
43 }
```

Bước 4: Test.

Các bạn bật Editor Unity lên, tạo 1 GameObject bất kì, sau đó kéo Script Settings vào. Tiếp theo các bạn modify các giá trị, cũng như kéo 1 prefab boss vào. Cuối cùng các bạn ấn vào button Display Value để xem log trên Console có đúng value các bạn vừa sửa không nhé.



Như vậy là chúng ta vừa tạo ra 1 setting có thể sửa trực tiếp trên inspector và nếu muốn các bạn có thể sửa nút Display Value thành Save Settings để lưu những thông số cài đặt đó xuống mà không phải play/run game mới có thể cập nhật được!

Chúc các bạn thành công nhé! 😊