

[Custom Property Attribute]

CONTACT

✉️ warmth.giver@gmail.com

 [GitHub](#)

Zion Lee

CONTENTS

목차

01

INTRODUCTION

제품 소개

02

KEY TECHNOLOGY

핵심 기술

03

FEATURES

기능 소개

INTRODUCTION

제품 소개

This asset provides 'CustomPropertyAttribute's that allow you to easily customize inspectors. It was created to eliminate the hassle of having to write editor scripts from scratch every time you customize an inspector. We built this using Unity's 'PropertyAttribute' and 'CustomPropertyDrawer' and improved the biggest issue with 'CustomPropertyDrawer' which doesn't allow multiple 'PropertyAttribute's to be applied to a single field to enable free inspector customization.

이 에셋은 인스펙터를 손쉽게 커스텀할 수 있는 'CustomPropertyAttribute'들을 제공합니다. 인스펙터를 커스텀 할 때마다 에디터 스크립트를 처음부터 작성해야 하는 번거로움 때문에 제작하게 되었습니다. 유니티에서 제공하는 'PropertyAttribute'와 'CustomPropertyDrawer'를 이용해 제작 했으며, 하나의 필드에 'PropertyAttribute'를 여러 개 적용할 수 없는 'CustomPropertyDrawer'의 가장 큰 문제점을 개선하여 자유로운 인스펙터 커스텀이 가능하도록 했습니다.

[CUSTOM PROPERTY ATTRIBUTE]

```
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74
```

```
..... [Space]  
..... [LineMargin = 0]  
..... [Text("<b>Debugging</b>", FontSize = 16)]  
..... [Margin]  
..... [ToggleIf("cannot(drawGizmo), false)]  
..... [SerializeField]  
..... [Alias("Default Color")]  
..... [UsingCustomProperty]  
..... [SerializeField]  
..... private Color defaultGizmoColor = new(1f, 0f, 0f, 0.25f);
```

KEY TECHNOLOGY

핵심 기술

CustomPropertyAttribute

After inheriting the class, you can override the function as needed to create a new 'CustomPropertyAttribute'.

- The 'Initialize' function initializes the attribute.
- The 'Preset' function presets settings such as whether to 'Draw' or not.
- The 'Draw' function draws property fields such as buttons, dividers, and spaces in the inspector.

해당 클래스를 상속한 뒤 용도에 맞게 함수를 재정의하여 새로운 'CustomPropertyAttribute'를 만들 수 있습니다.

- 'Initialize' 함수는 어트리뷰트를 초기화합니다.
- 'Preset' 함수는 'Draw' 여부 등의 사전 설정을 합니다.
- 'Draw' 함수는 인스펙터에 버튼, 구분선, 공백 등 프로퍼티 필드를 그립니다.

```
[AttributeUsage(AttributeTargets.Field, Inherited = true, AllowMultiple = true)]  
public abstract class CustomPropertyAttribute : PropertyAttribute  
{  
    ...  
    protected virtual void Initialize(Drawer drawer) {}  
    protected virtual void Preset(Drawer drawer) {}  
    protected virtual void Draw(Drawer drawer) {}  
    ...  
}
```

```
public sealed class ButtonAttribute : CustomPropertyAttribute  
{  
    ...  
    protected override void Initialize(Drawer drawer)  
    {  
        var type = drawer.TargetObject.GetType();  
        method = type.GetMethod(methodName);  
    }  
    protected override void Draw(Drawer drawer)  
    {  
        drawer.DrawButton(method, text, Height);  
    }  
}
```

KEY TECHNOLOGY

핵심 기술

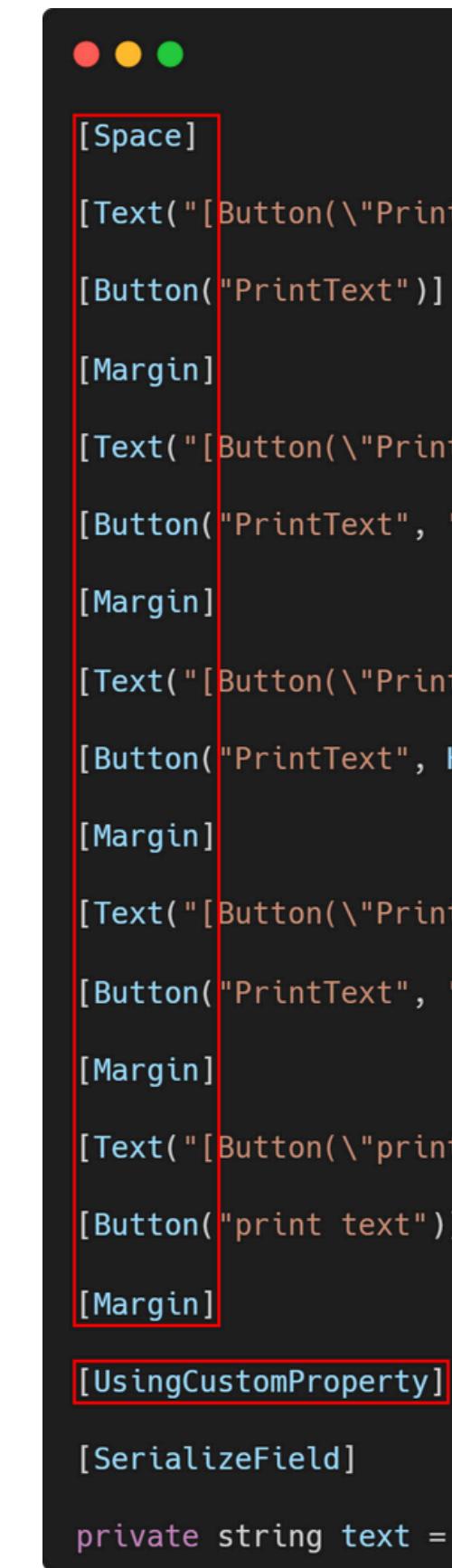
[UsingCustomProperty]

Collects all 'CustomPropertyAttribute's declared in the field and draws them all at once. This improves the issue where multiple 'PropertyAttribute's cannot be applied to a single field.

When the inspector is redrawn in 'CustomPropertyDrawer', the 'Initialize' function of the attributes collected by 'CustomPropertyAttribute' is called once. After that, whenever a change in the inspector is detected, the 'Preset' function and 'Draw' function are called.

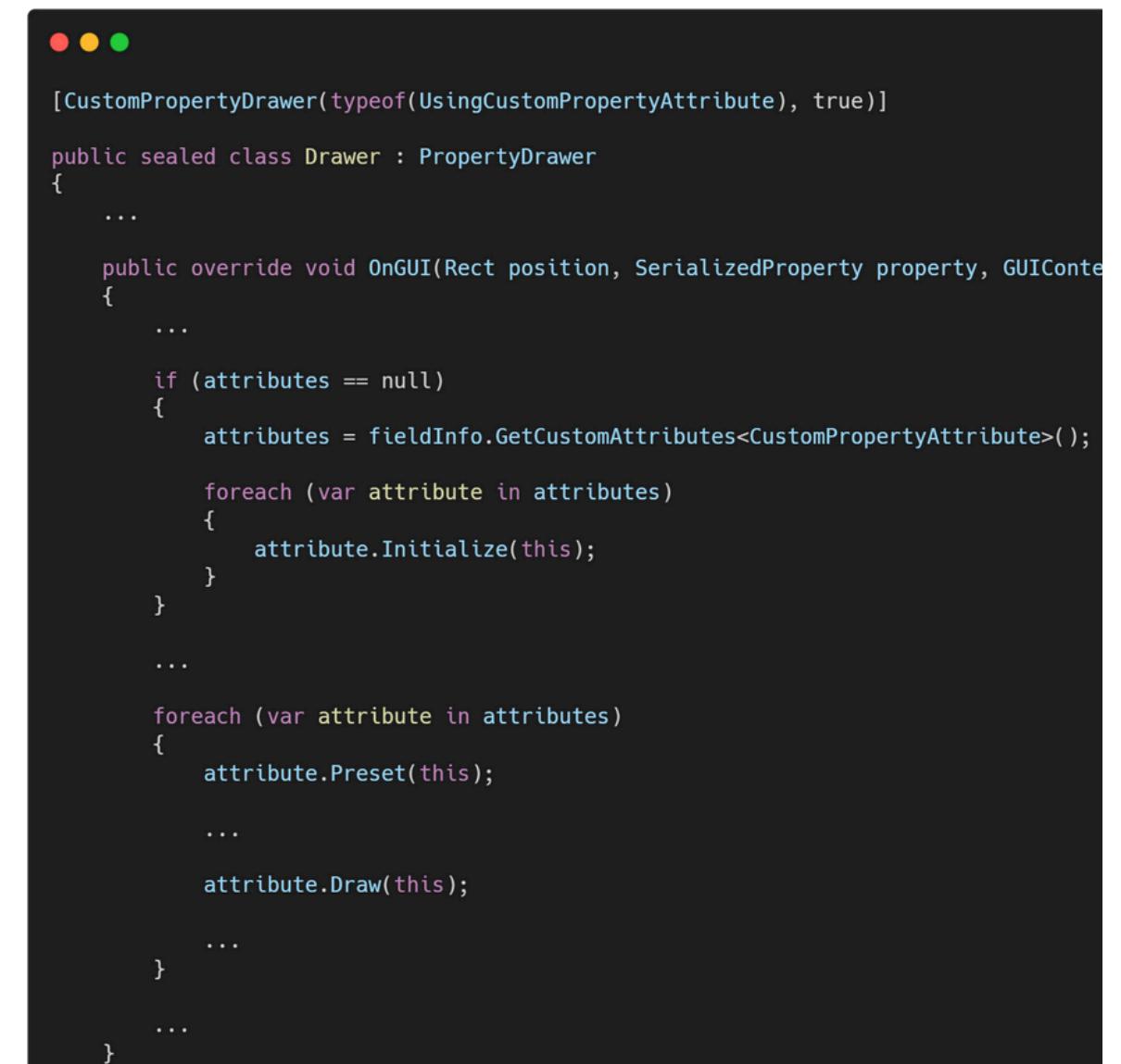
필드에 선언된 모든 'CustomPropertyAttribute'들을 모아서 한 번에 그립니다. 이것으로 하나의 필드에 'PropertyAttribute'를 여러 개 적용할 수 없는 문제를 개선합니다.

'CustomPropertyDrawer'에서 인스펙터가 새로 그려질 때 'CustomPropertyAttribute'가 모은 어트리뷰트들의 'Initialize' 함수를 한 번 호출합니다. 이후 인스펙터의 변화가 감지될 때마다 'Preset' 함수와 'Draw' 함수를 호출합니다.



The screenshot shows the Unity Inspector for a script. A specific field is selected, and its properties are displayed. The properties include '[Space]', '[Text("["], [Button(@"PrintT"), and '[Margin]'. These are highlighted with red boxes, indicating they are multiple 'PropertyAttribute' instances being managed by a single 'CustomPropertyAttribute'.

```
[Space]
[Text("[Button(@"PrintT
[Button("PrintText")]
[Margin]
[Text("[Button(@"PrintT
[Button("PrintText", "P
[Margin]
[Text("[Button(@"PrintT
[Button("PrintText", He
[Margin]
[Text("[Button(@"PrintT
[Button("PrintText", "P
[Margin]
[Text("[Button(@"print
[Button("print text")]
[Margin]
[UsingCustomProperty]
[SerializeField]
private string text = "
```



The screenshot shows the source code for the 'CustomPropertyDrawer' class. It includes logic for collecting attributes, initializing them once, and then calling 'Preset' and 'Draw' functions whenever the inspector changes.

```
[CustomPropertyDrawer(typeof(UsingCustomPropertyAttribute), true)]
public sealed class Drawer : PropertyDrawer
{
    ...
    public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
    {
        ...
        if (attributes == null)
        {
            attributes = fieldInfo.GetCustomAttributes<CustomPropertyAttribute>();
            foreach (var attribute in attributes)
            {
                attribute.Initialize(this);
            }
        }
        ...
        foreach (var attribute in attributes)
        {
            attribute.Preset(this);
            ...
            attribute.Draw(this);
        }
        ...
    }
}
```

FEATURES

기능 소개

[Alias]

Assign aliases to fields. Specify 'Null' as a parameter to draw an empty label. Specify 'Empty("")' to not draw a label.

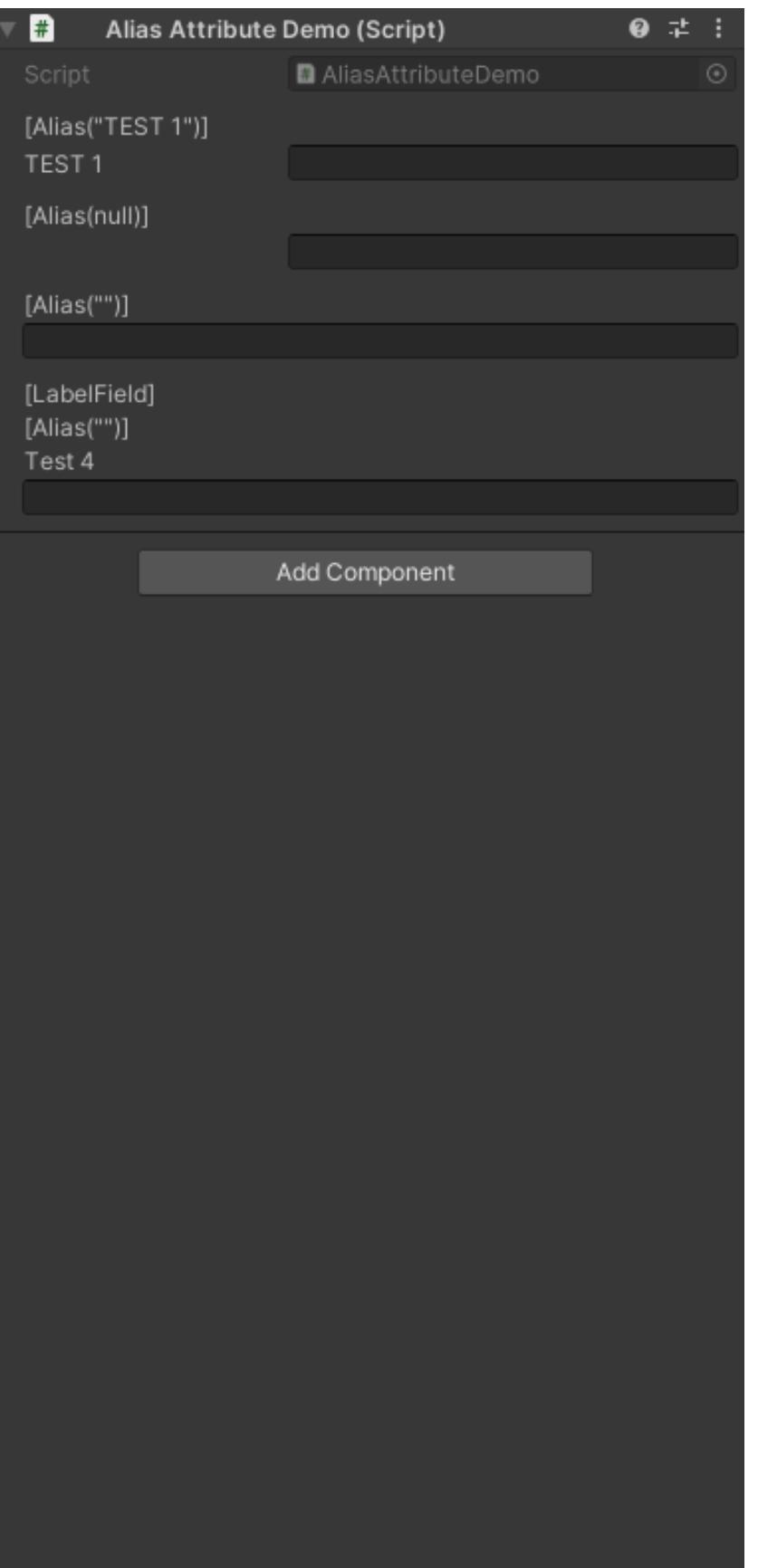
필드에 별칭을 부여합니다. 매개변수로 'Null'을 지정하면 빈 라벨을 그립니다. 'Empty("")'를 지정하면 라벨을 그리지 않습니다.

```
[Space]
[Text("[Alias(\"TEST 1\")]")]
[Alias("TEST 1")]
[UsingCustomProperty]
[SerializeField]
private string test1 = "";

[Space]
[Text("[Alias(null)]")]
[Alias(null)]
[UsingCustomProperty]
[SerializeField]
private string test2 = "";

[Space]
[Text("[Alias(\"\\\")")]
[Alias("")]
[UsingCustomProperty]
[SerializeField]
private string test3 = "";

[Space]
[Text("[LabelField]")]
[Text("[Alias(\"\\\")")]
[LabelField]
[Alias("")]
[PropertyField]
[UsingCustomProperty]
[SerializeField]
private string test4 = "";
```



FEATURES

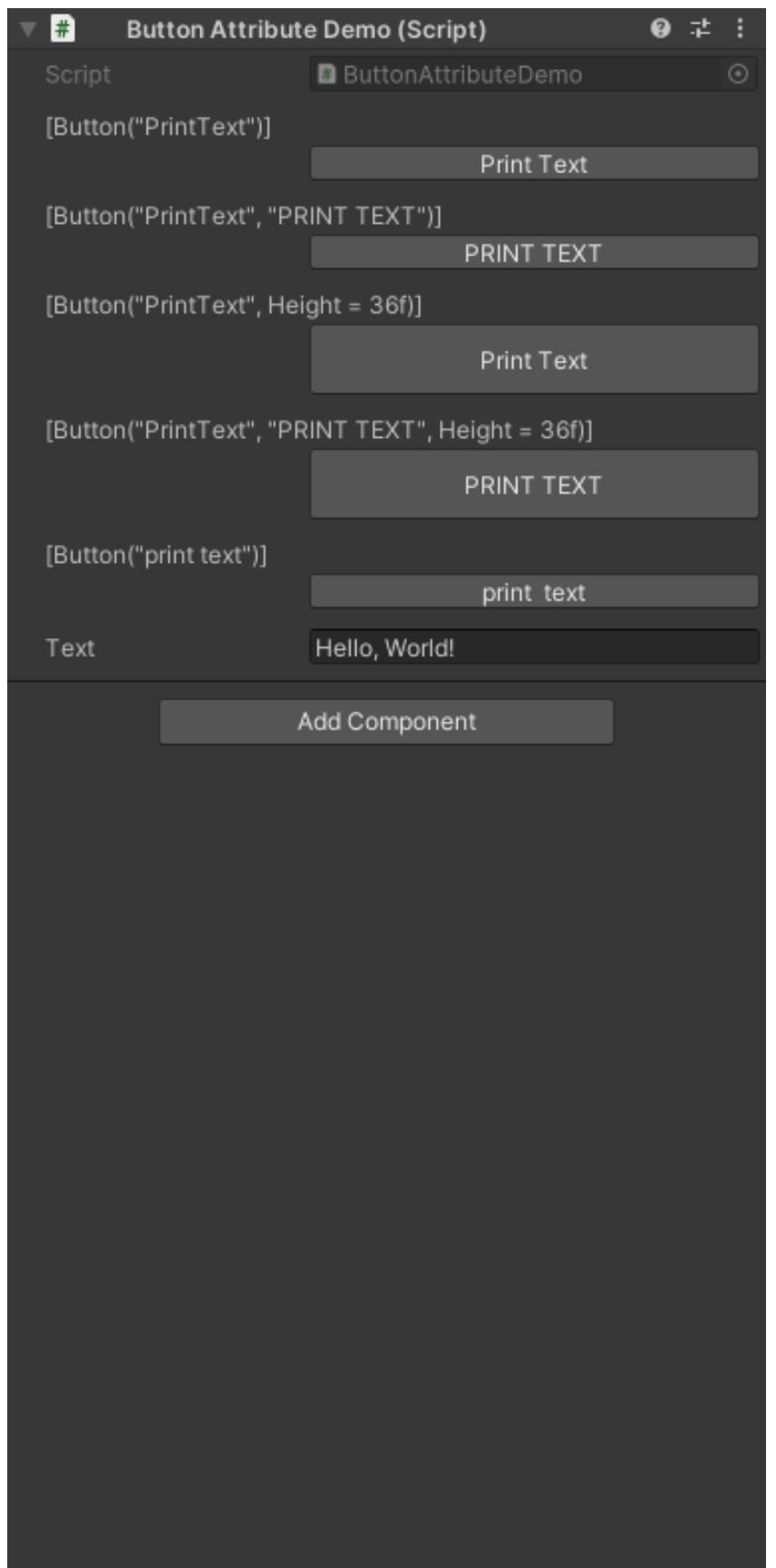
기능 소개

[Button]

Displays the function as a button in the inspector. You can specify the text and height of the button as parameters.

함수를 인스펙터에 버튼으로 표시합니다. 매개변수로 버튼의 텍스트와 높이를 지정할 수 있습니다.

```
[Space]  
[Text("[Button(\"PrintText\"])]")]  
[Button("PrintText")]  
[Margin]  
[Text("[Button(\"PrintText\", \"PRINT TEXT\")]")]  
[Button("PrintText", "PRINT TEXT")]  
[Margin]  
[Text("[Button(\"PrintText\", Height = 36f\")]")]  
[Button("PrintText", Height = 36f)]  
[Margin]  
[Text("[Button(\"PrintText\", \"PRINT TEXT\", Height = 36f\")]")]  
[Button("PrintText", "PRINT TEXT", Height = 36f)]  
[Margin]  
[Text("[Button(\"print text\")]")]  
[Button("print text")]  
[Margin]  
[UsingCustomProperty]  
[SerializeField]  
private string text = "Hello, World!";  
public void PrintText()  
{  
    FixedDebug.Log(text);  
}
```



FEATURES

기능 소개

[ReadOnly]

Sets the fields and attributes declared under the specified attribute as read-only.

해당 Attribute 하위에 선언된 필드 및 Attribute들을 읽기 전용으로 설정합니다.

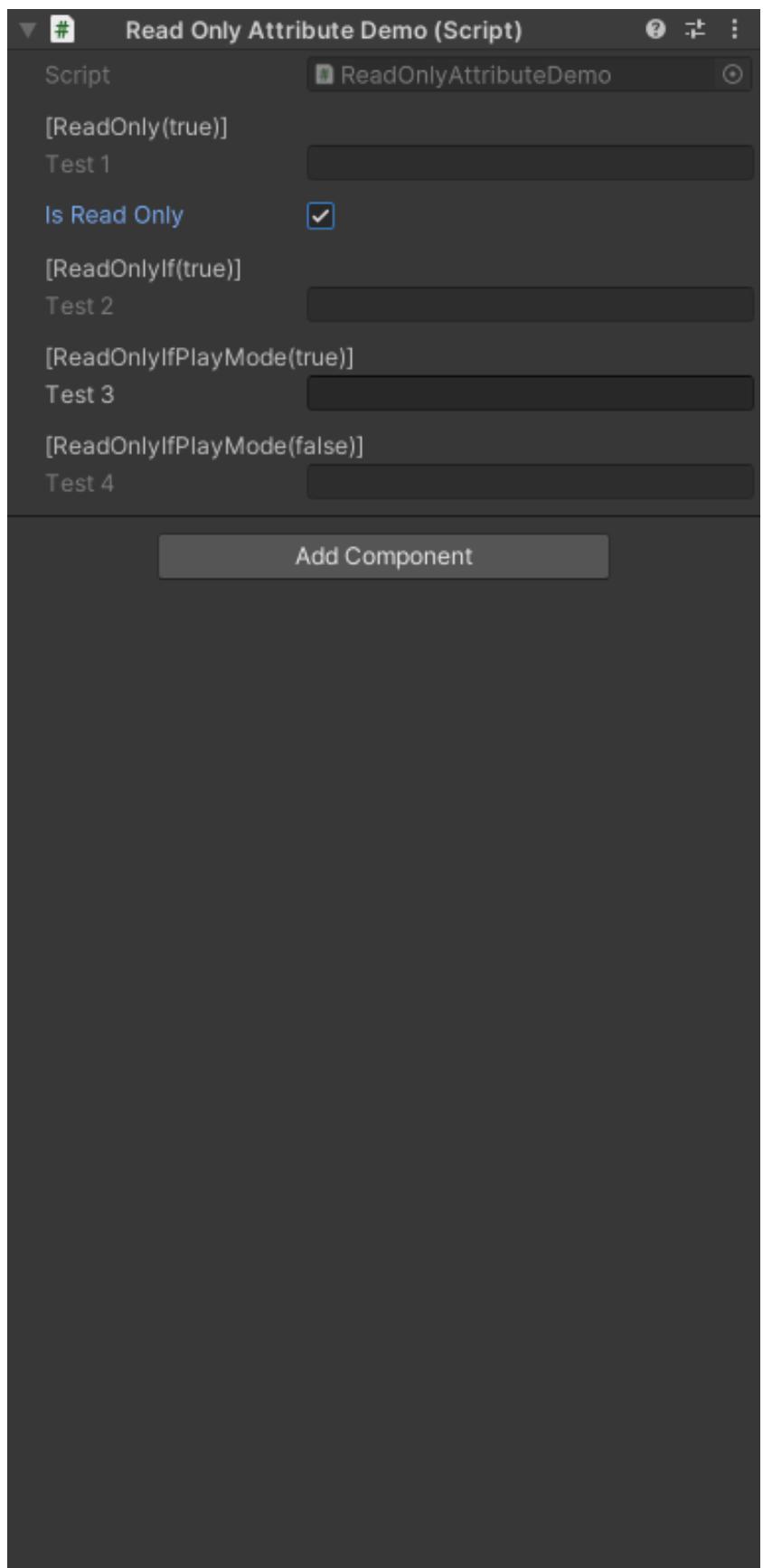
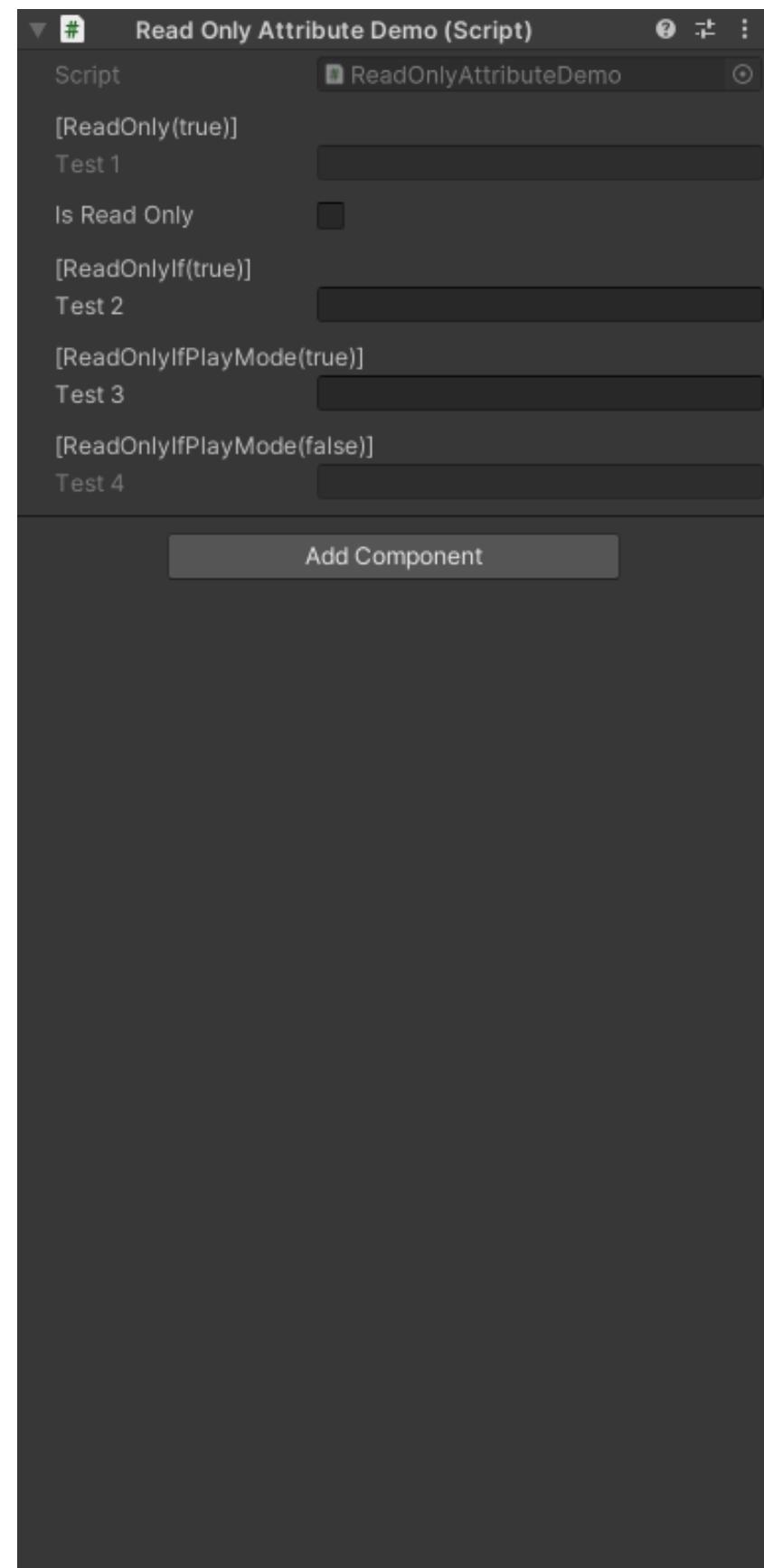
[ReadOnlyIf]

[ReadOnlyIfEditMode]

Set as read-only only under specific conditions, such as when the value of a specific field is compared or when the application is playing.

특정 필드의 값 비교, 어플리케이션이 실행중인지 아닌지 등 특정 조건에서만 읽기 전용으로 설정합니다.

```
[Space]  
[Text("[ReadOnly(true)]")]  
[ReadOnly(true)]  
[UsingCustomProperty]  
[SerializeField]  
private string test1 = "";  
  
[Space]  
[SerializeField]  
private bool isReadOnly = false;  
  
[Space]  
[Text("[ReadOnlyIf(true)]")]  
[ReadOnlyIf("isReadOnly", true)]  
[UsingCustomProperty]  
[SerializeField]  
private string test2 = "";  
  
[Space]  
[Text("[ReadOnlyIfEditMode(true)]")]  
[ReadOnlyIfEditMode(true)]  
[UsingCustomProperty]  
[SerializeField]  
private string test3 = "";  
  
[Space]  
[Text("[ReadOnlyIfEditMode(false)]")]  
[ReadOnlyIfEditMode(false)]  
[UsingCustomProperty]  
[SerializeField]  
private string test4 = "";
```



FEATURES

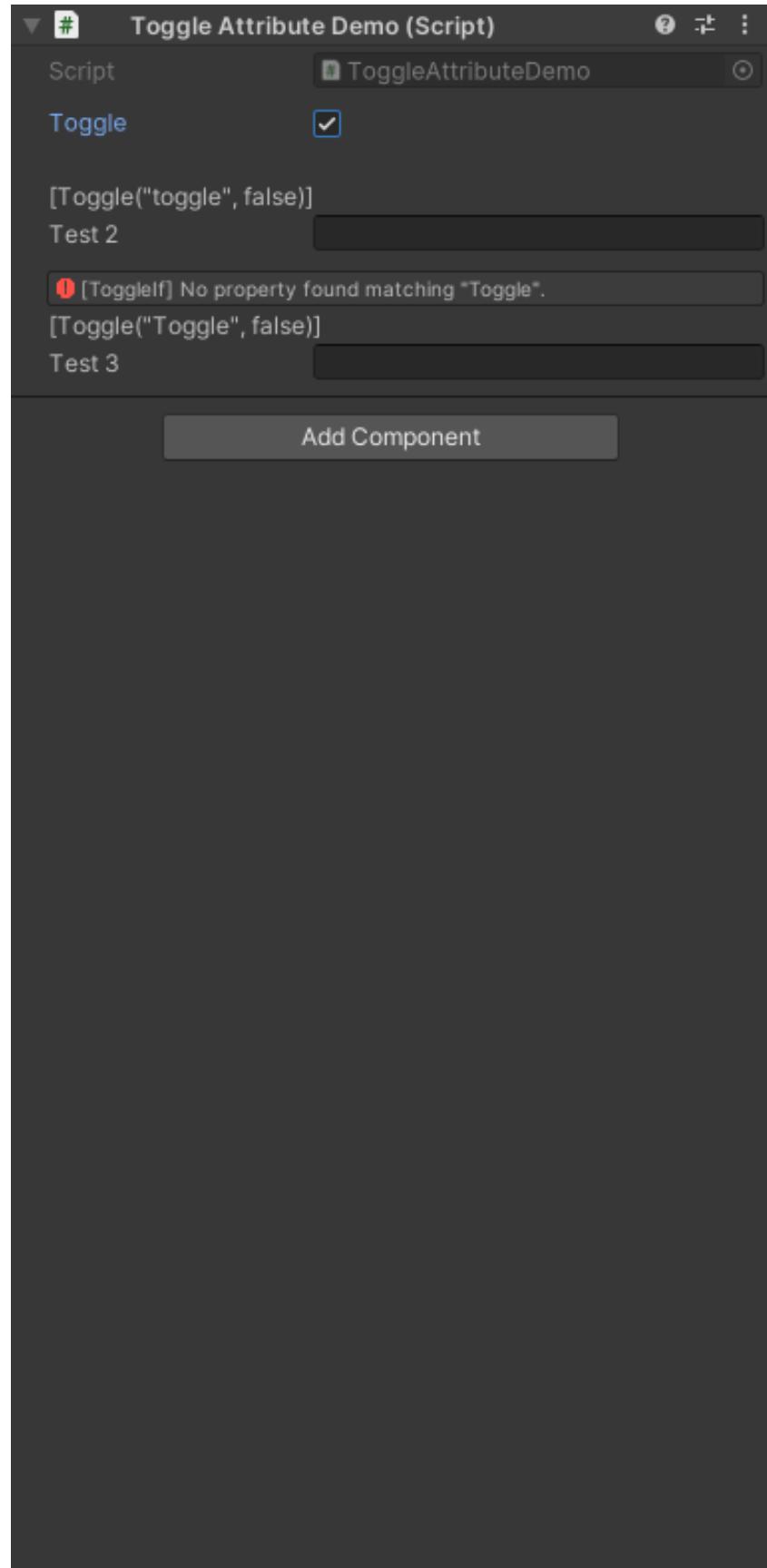
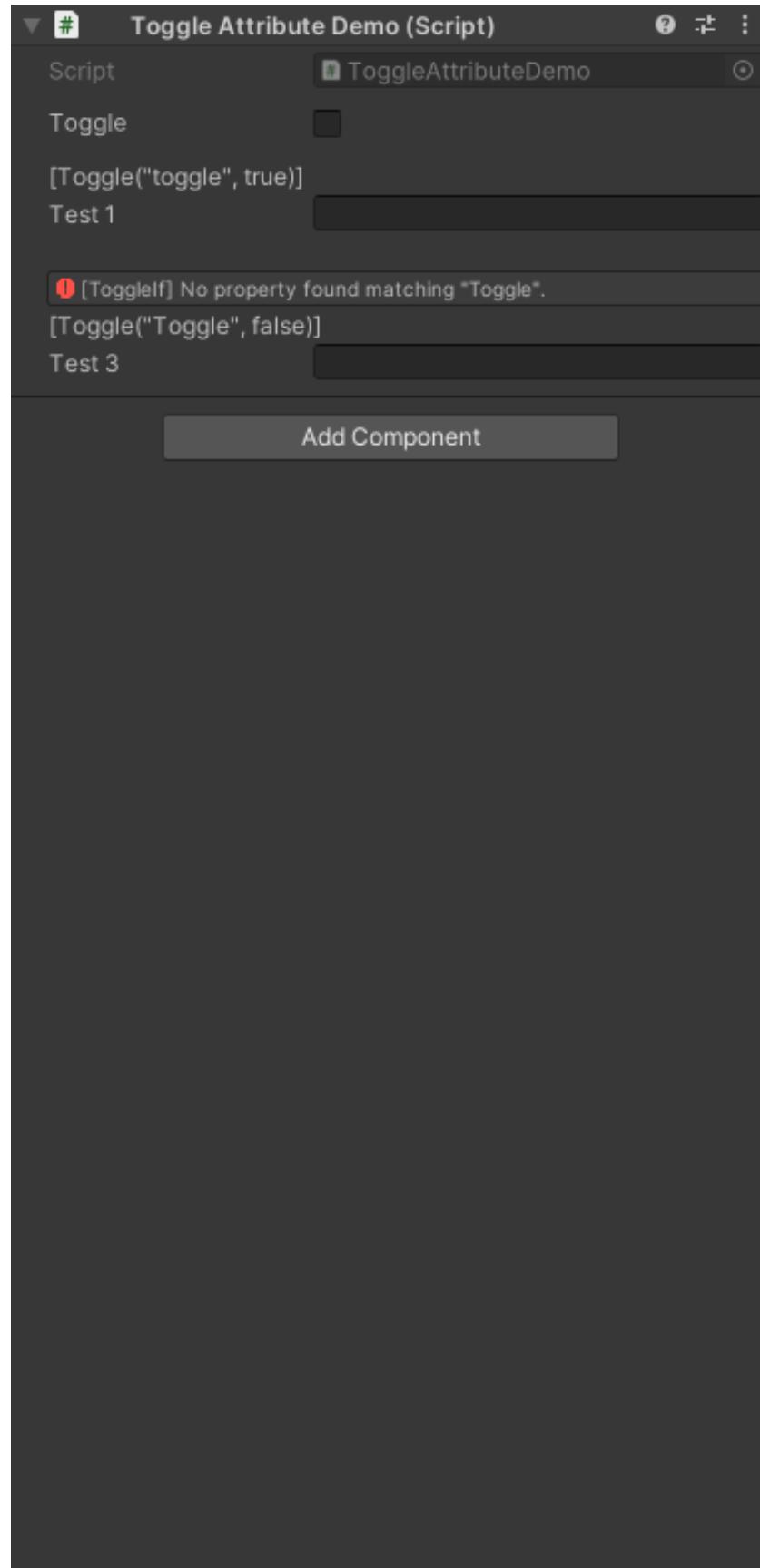
기능 소개

[Toggle]

Disables the fields and attributes declared under the specified attribute. It is also possible to disable them only under specific conditions by comparing the status of specific fields.

해당 Attribute 하위에 선언된 필드 및 Attribute들을 비활성화시킵니다. 특정 필드의 상태 비교로 특정 조건에서만 비활성화하는 것도 가능합니다.

```
[Space]  
[SerializeField]  
  
private bool toggle = false;  
  
[Space]  
  
[ToggleIf("toggle", true)]  
[Text("[Toggle(\"toggle\", true)])"]  
[UsingCustomProperty]  
  
[SerializeField]  
  
private string test1 = "";  
  
[Space]  
  
[ToggleIf("toggle", false)]  
[Text("[Toggle(\"toggle\", false)])"]  
[UsingCustomProperty]  
  
[SerializeField]  
  
private string test2 = "";  
  
[Space]  
  
[ToggleIf("Toggle", false)]  
[Text("[Toggle(\"Toggle\", false)])"]  
[UsingCustomProperty]  
  
[SerializeField]  
  
private string test3 = "";
```



FEATURES

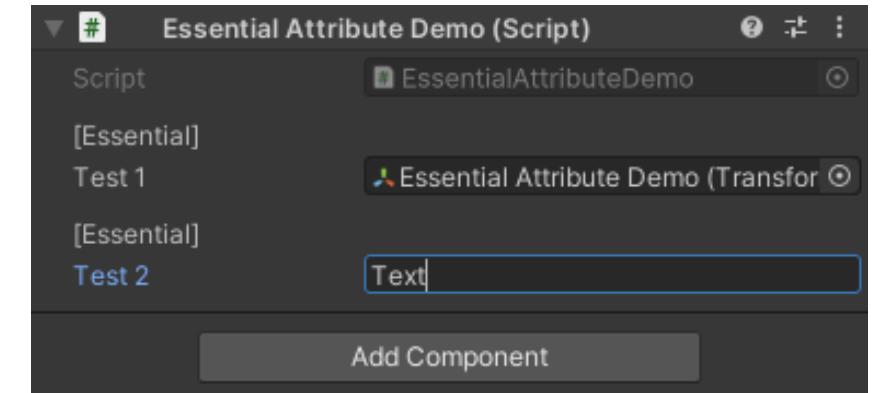
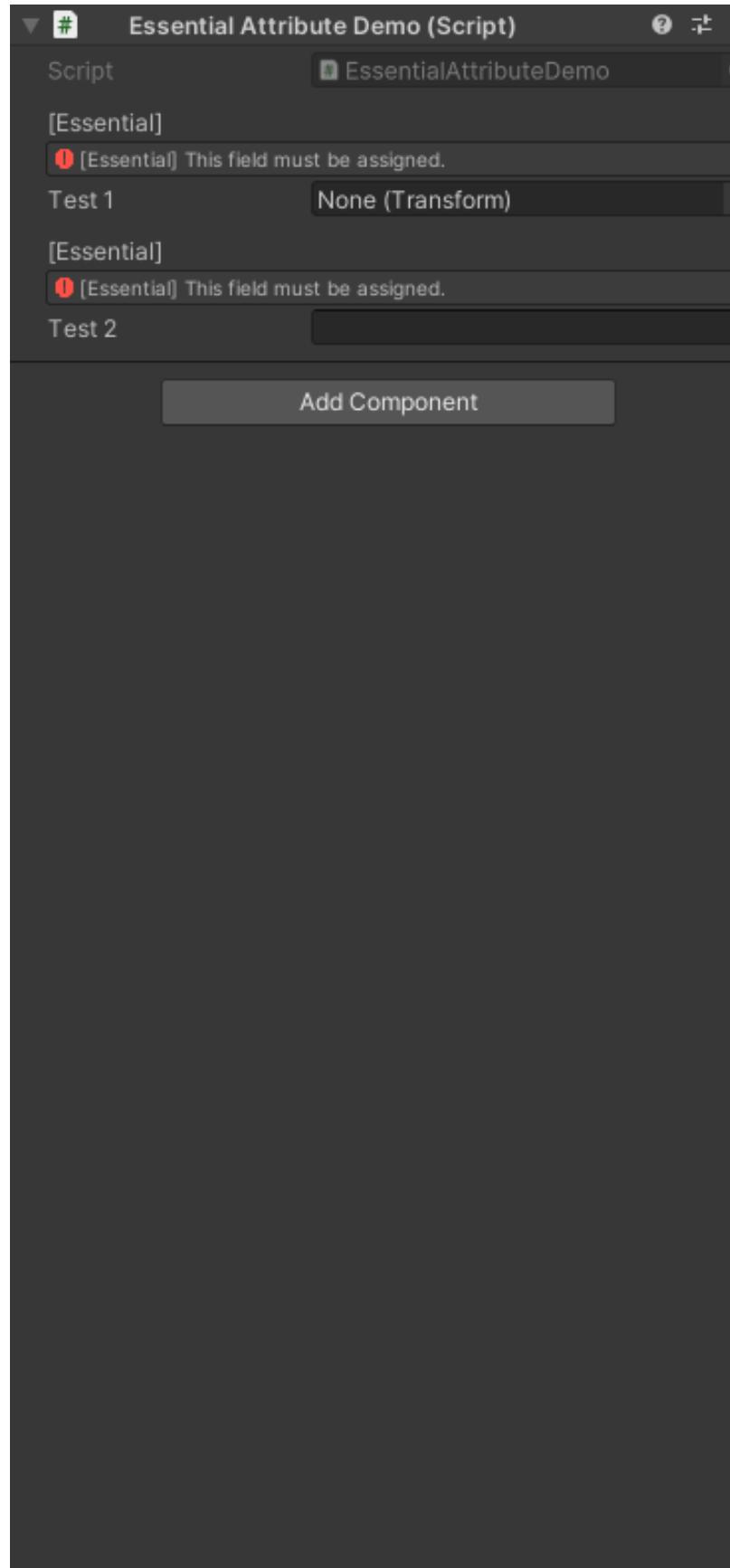
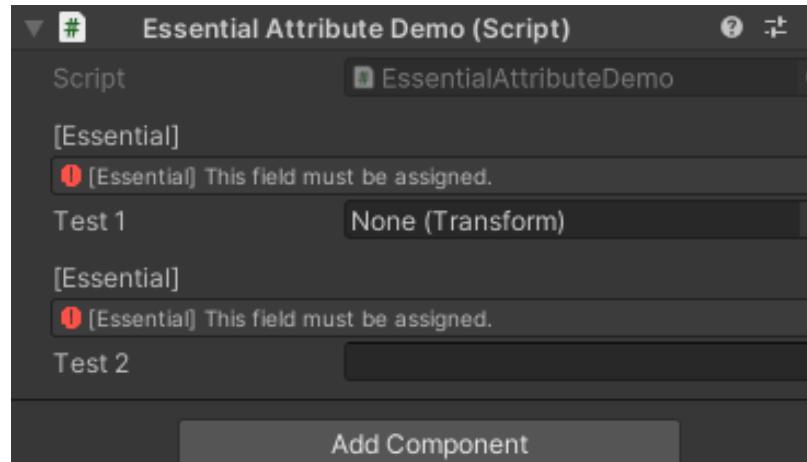
기능 소개

[Essential]

Displays a warning message when the value of an 'Object' type field is 'Null'. For 'string' type fields, displays a warning message when the value is 'Empty("")'.

'Object' 타입 필드의 값이 'Null' 일 때 경고 메시지를 표시합니다.
'string' 타입 필드의 경우 값이 'Empty("")' 일 때 경고 메시지를 표시합니다.

```
Space  
[Text(" [Essential] ")]  
[Essential]  
[UsingCustomProperty]  
[SerializeField]  
private Transform test1 = null;  
  
Space  
[Text(" [Essential] ")]  
[Essential]  
[UsingCustomProperty]  
[SerializeField]  
private string test2 = "";
```



FEATURES

기능 소개

[PropertyField]

Draw the field before other attributes.

필드를 다른 Attribute보다 먼저 그립니다.

[LabelField]

Draw only the labels in the field.

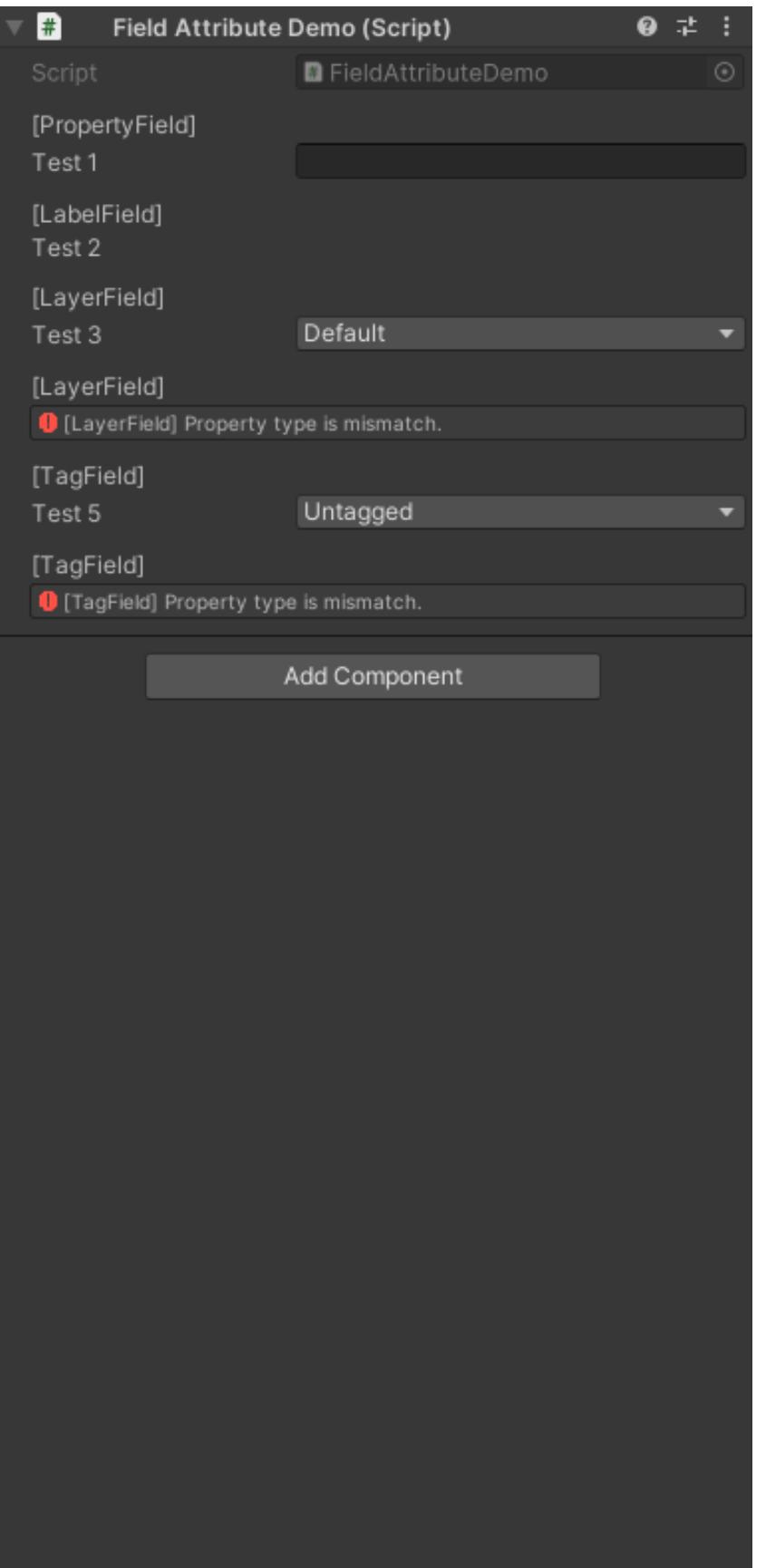
필드의 라벨만 따로 그립니다.

[LayerField], [TagField]

Displays 'int' and 'string' type fields as 'Layer' and 'Tag' drop-down menus.

'int' 및 'string' 타입 필드를 'Layer' 및 'Tag' 드롭다운 메뉴로 표시합니다.

```
[Space]
[Text("[PropertyField]")]
[PropertyField]
[UsingCustomProperty]
[SerializeField]
private string test1 = "";
[Space]
[Text("[LabelField]")]
[LabelField]
[UsingCustomProperty]
[SerializeField]
private string test2 = "";
[Space]
[Text("[LayerField]")]
[LayerField]
[UsingCustomProperty]
[SerializeField]
private int test3 = 0;
[Space]
[Text("[LayerField]")]
[LayerField]
[UsingCustomProperty]
[SerializeField]
private string test4 = "";
[Space]
[Text("[TagField]")]
[TagField]
[UsingCustomProperty]
[SerializeField]
private string test5 = "Untagged";
[Space]
[Text("[TagField]")]
[TagField]
[UsingCustomProperty]
[SerializeField]
private int test6 = 0;
```



FEATURES

기능 소개

[GetComponent]

[GetComponentInParent]

[GetComponentInParentOnly]

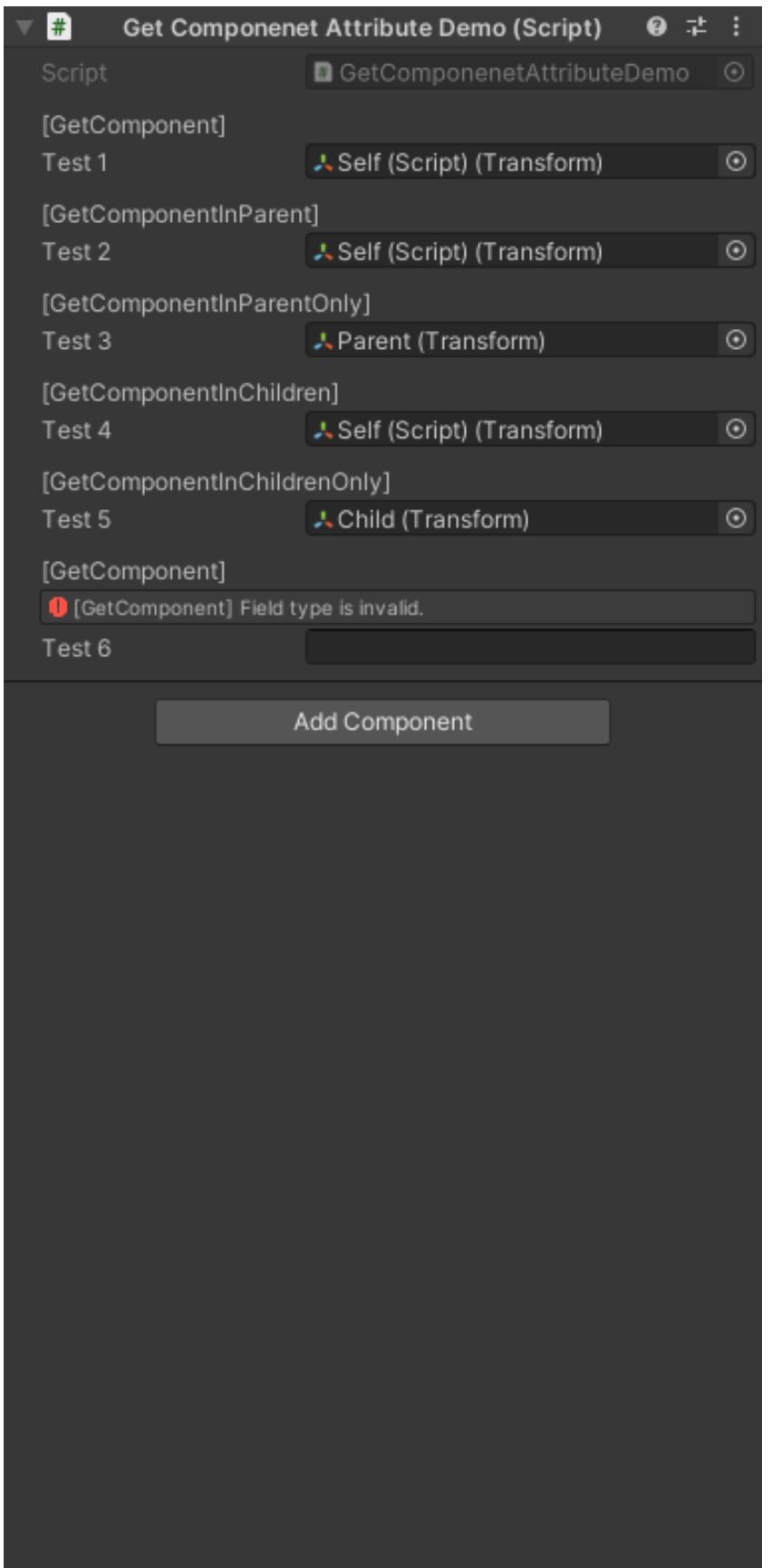
[GetComponentInChildren]

[GetComponentInChildrenOnly]

Finds and assigns components of the corresponding type to the 'Component' type field within each scope.

'Component' 타입 필드에 해당 타입의 컴포넌트를 각 범위 내에서 찾아서 할당합니다.

```
Space
[Text("[GetComponent]")]
 GetComponent
[UsingCustomProperty]
[SerializeField]
private Transform test1 = null;
Space
[Text("[GetComponentInParent]")]
 GetComponentInParent
[UsingCustomProperty]
[SerializeField]
private Transform test2 = null;
Space
[Text("[GetComponentInParentOnly]")]
 GetComponentInParentOnly
[UsingCustomProperty]
[SerializeField]
private Transform test3 = null;
Space
[Text("[GetComponentInChildren]")]
 GetComponentInChildren
[UsingCustomProperty]
[SerializeField]
private Transform test4 = null;
Space
[Text("[GetComponentInChildrenOnly]")]
 GetComponentInChildrenOnly
[UsingCustomProperty]
[SerializeField]
private Transform test5 = null;
Space
[Text("[GetComponent]")]
 GetComponent
[UsingCustomProperty]
[SerializeField]
private string test6 = "";
```



FEATURES

기능 소개

[MessageBox]

[InfoBox]

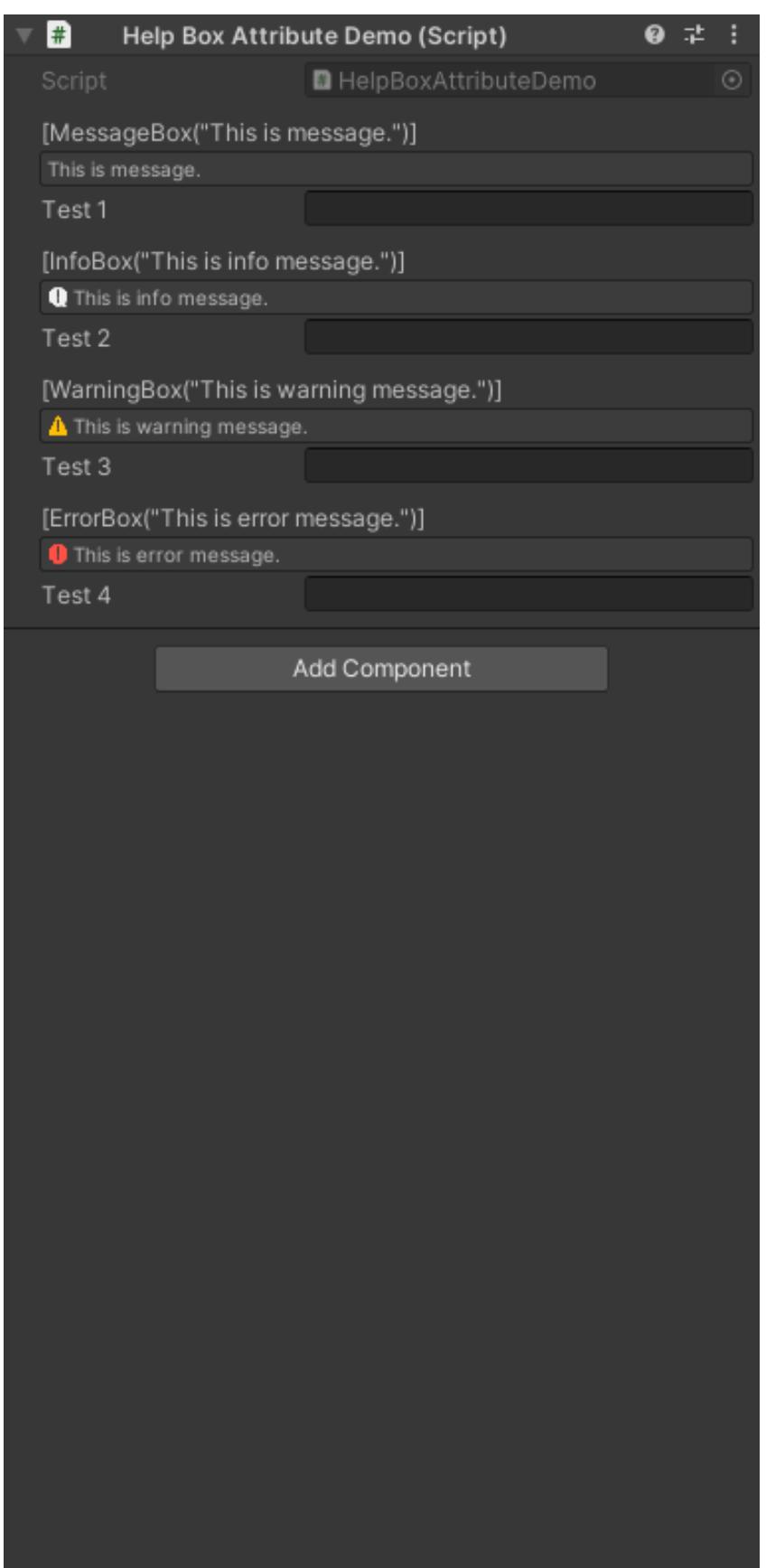
[WarningBox]

[ErrorBox]

Draw a message box with the corresponding icon.

각자 해당하는 아이콘과 함께 메시지 박스를 그립니다.

```
● ● ●  
[Space]  
[Text(" [MessageBox(\"This is message.\") ] ")]  
[MessageBox("This is message.")]  
[UsingCustomProperty]  
[SerializeField]  
private string test1 = "";  
[Space]  
[Text(" [InfoBox(\"This is info message.\") ] ")]  
[InfoBox("This is info message.")]  
[UsingCustomProperty]  
[SerializeField]  
private string test2 = "";  
[Space]  
[Text(" [WarningBox(\"This is warning message.\") ] ")]  
[WarningBox("This is warning message.")]  
[UsingCustomProperty]  
[SerializeField]  
private string test3 = "";  
[Space]  
[Text(" [ErrorBox(\"This is error message.\") ] ")]  
[ErrorBox("This is error message.")]  
[UsingCustomProperty]  
[SerializeField]  
private string test4 = "";
```



FEATURES

기능 소개

[AddIndent]

Indents the field by the specified value.

필드를 지정한 값만큼 들여 씁니다.

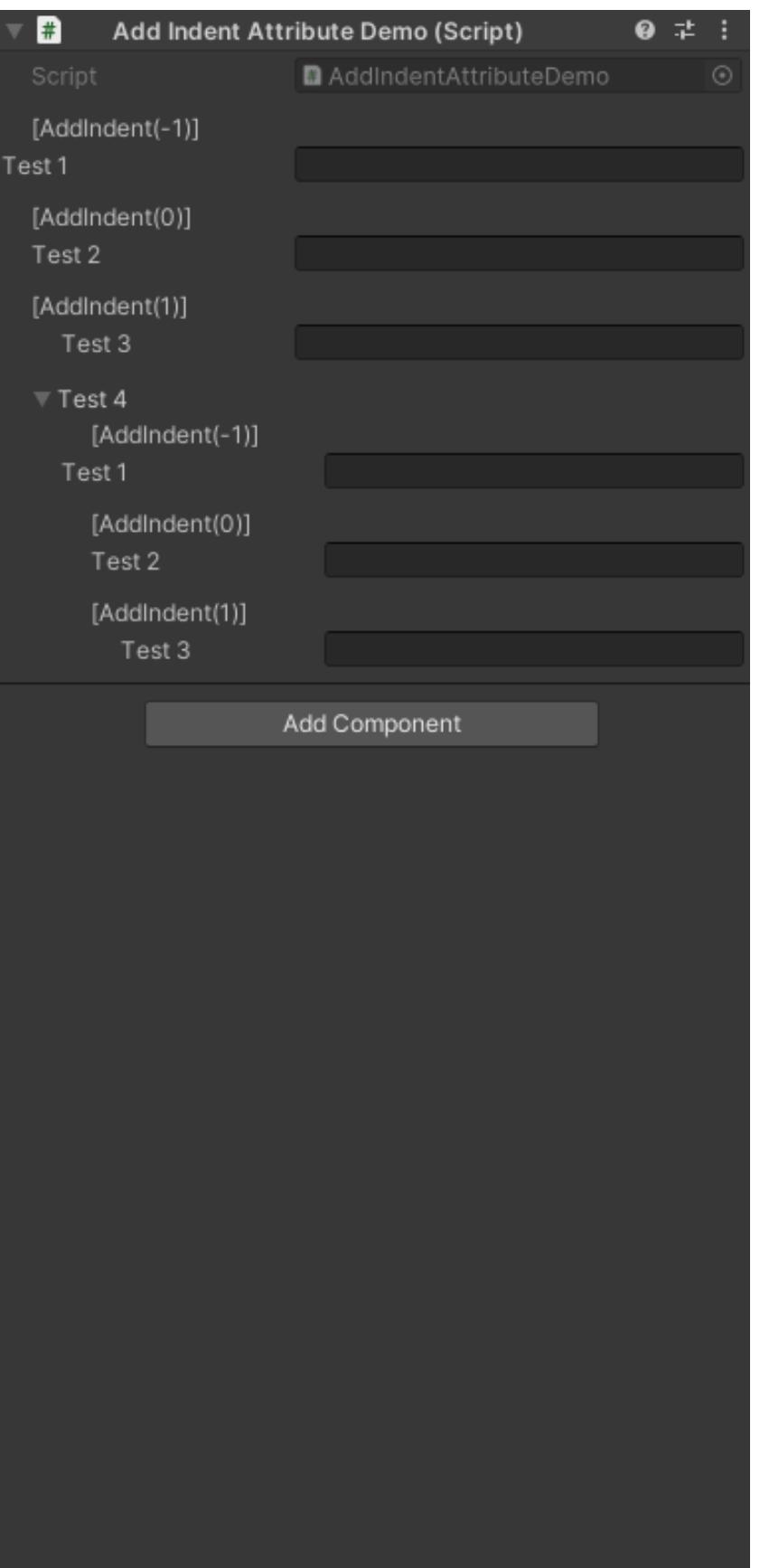
```
[Space]
[Text(" [AddIndent(-1)]")]
[AddIndent(-1)]
[UsingCustomProperty]
[SerializeField]
private string test1 = "";

[Space]
[Text(" [AddIndent(0)]")]
[AddIndent(0)]
[UsingCustomProperty]
[SerializeField]
private string test2 = "";

[Space]
[Text(" [AddIndent(1)]")]
[AddIndent(1)]
[UsingCustomProperty]
[SerializeField]
private string test3 = "";

[Space]
[UsingCustomProperty]
[SerializeField]
private Test test4 = null;

[Serializable]
public class Test
{
    "
}
```



FEATURES

기능 소개

[SetIndent]

Specifies the amount of indentation in the field.

The difference between 'AddIndent' and 'SetIndent' is that 'AddIndent' adds indentation to the original position, while 'SetIndent' specifies the indentation position.

필드의 들여 쓴 정도를 지정합니다.

'AddIndent'와의 차이점은 'AddIndent'는 본래 위치에서 들여 쓰기를 더하는 것이고 'SetIndent'는 들여 쓴 위치를 지정한다는 것입니다.

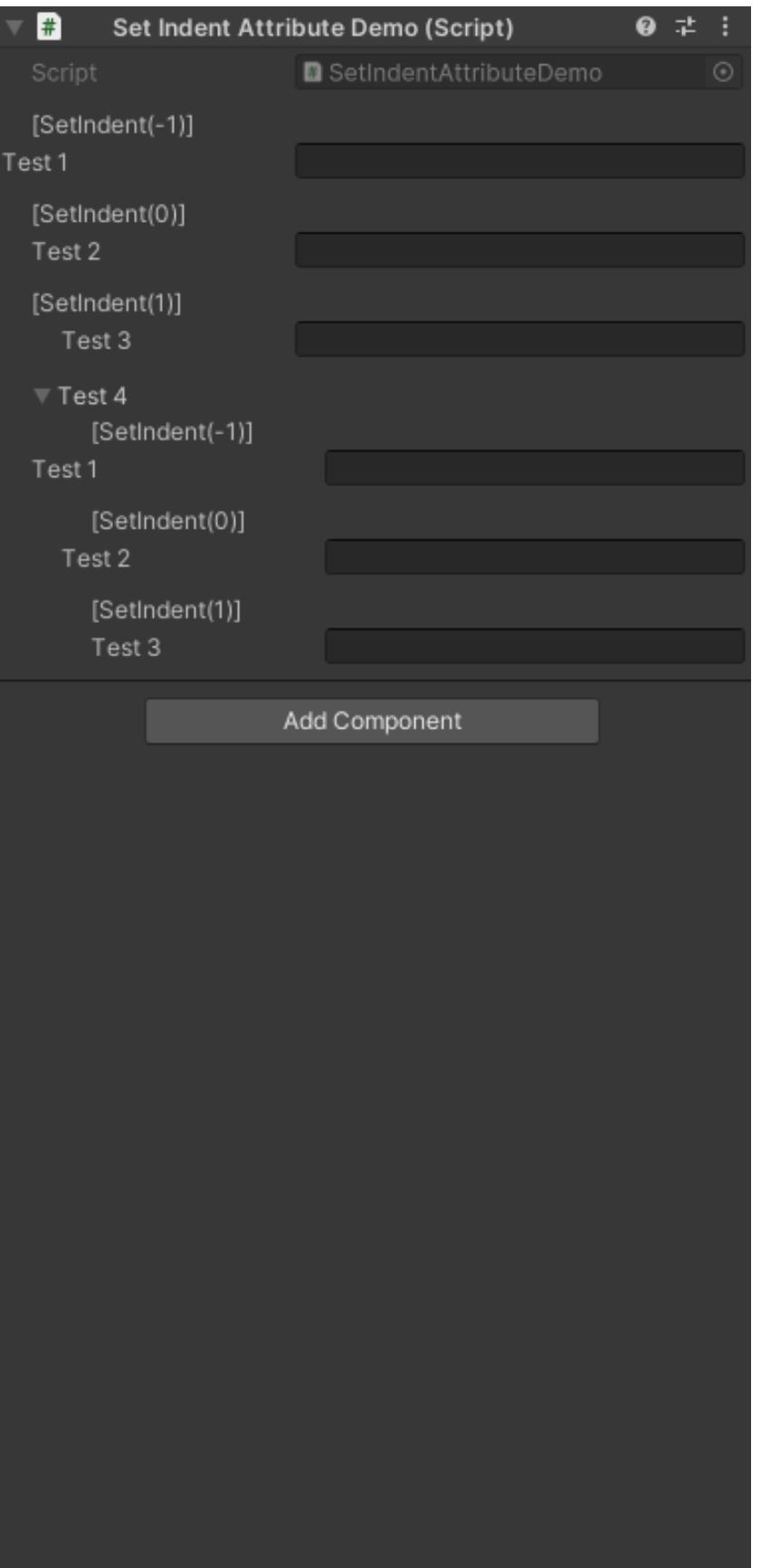
```
[Space]
[Text("[SetIndent(-1)]")]
[SetIndent(-1)]
[UsingCustomProperty]
[SerializeField]
private string test1 = "";

[Space]
[Text("[SetIndent(0)]")]
[SetIndent(0)]
[UsingCustomProperty]
[SerializeField]
private string test2 = "";

[Space]
[Text("[SetIndent(1)]")]
[SetIndent(1)]
[UsingCustomProperty]
[SerializeField]
private string test3 = "";

[Space]
[UsingCustomProperty]
[SerializeField]
private Test test4 = null;

[Serializable]
public class Test
{
    "
}
```



FEATURES

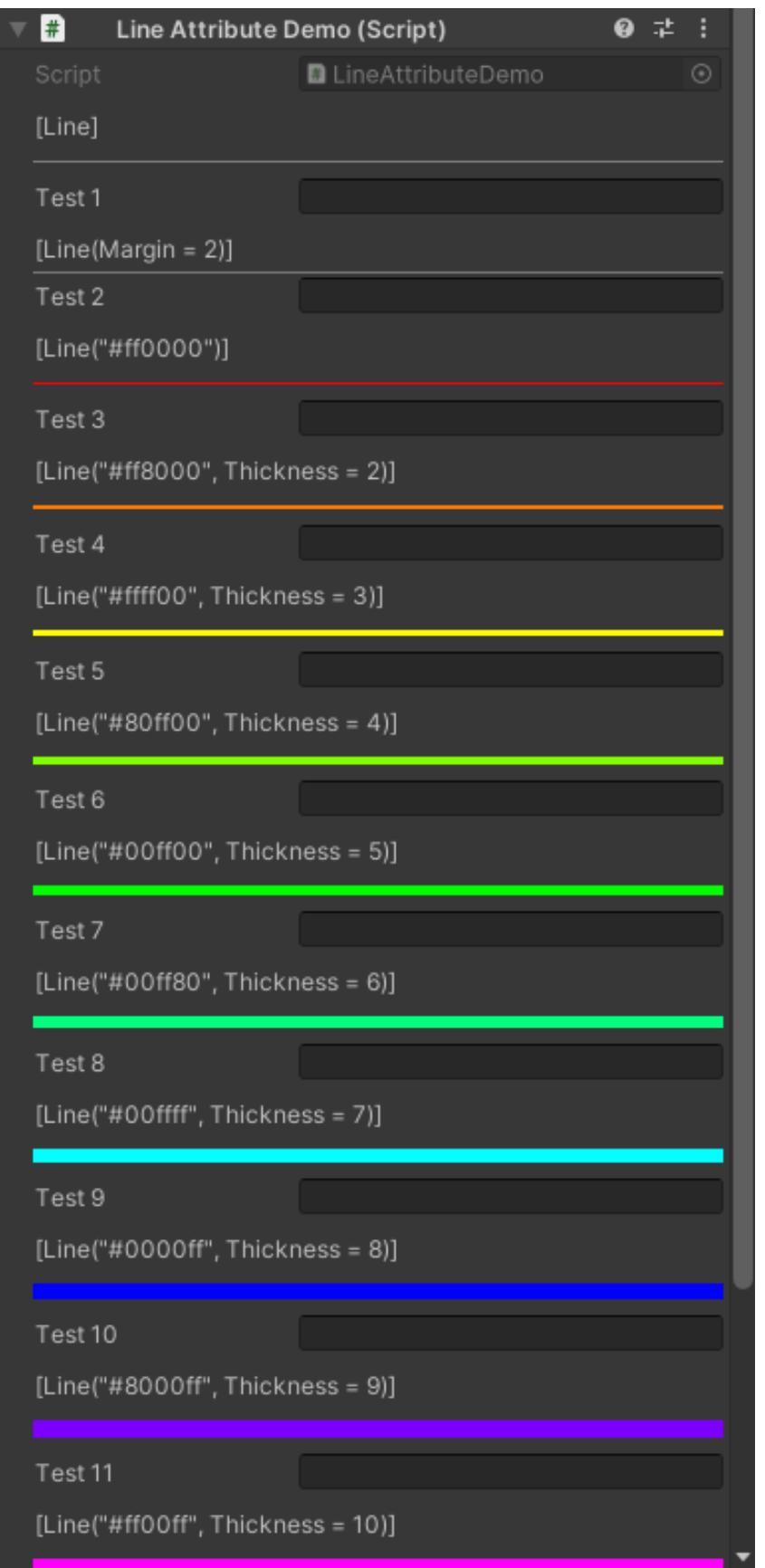
기능 소개

[Line]

Draw a dividing line. You can specify the color and thickness of the line with parameters.

구분선을 그립니다. 매개변수로 선의 색상과 굵기를 지정할 수 있습니다.

```
[Space]
[Text("[Line]")]
[Line]
[UsingCustomProperty]
[SerializeField]
private string test1 = "";
[Space]
[Text("[Line(Margin = 2)]")]
[Line(Margin = 2)]
[UsingCustomProperty]
[SerializeField]
private string test2 = "";
[Space]
[Text("[Line(\"#ff0000\")]")]
[Line("#ff0000")]
[UsingCustomProperty]
[SerializeField]
private string test3 = "";
[Space]
[Text("[Line(\"#ff8000\", Thickness = 2)]")]
[Line("#ff8000", Thickness = 2)]
[UsingCustomProperty]
[SerializeField]
private string test4 = "";
[Space]
[Text("[Line(\"#ffff00\", Thickness = 3)]")]
[Line("#ffff00", Thickness = 3)]
[UsingCustomProperty]
[SerializeField]
private string test5 = "";
[Space]
[Text("[Line(\"#00ff00\", Thickness = 4)]")]
[Line("#00ff00", Thickness = 4)]
[UsingCustomProperty]
[SerializeField]
private string test6 = "";
[Space]
[Text("[Line(\"#00ff80\", Thickness = 5)]")]
[Line("#00ff80", Thickness = 5)]
[UsingCustomProperty]
[SerializeField]
private string test7 = "";
[Space]
[Text("[Line(\"#00ffff\", Thickness = 6)]")]
[Line("#00ffff", Thickness = 6)]
[UsingCustomProperty]
[SerializeField]
private string test8 = "";
[Space]
[Text("[Line(\"#0000ff\", Thickness = 7)]")]
[Line("#0000ff", Thickness = 7)]
[UsingCustomProperty]
[SerializeField]
private string test9 = "";
[Space]
[Text("[Line(\"#8000ff\", Thickness = 8)]")]
[Line("#8000ff", Thickness = 8)]
[UsingCustomProperty]
[SerializeField]
private string test10 = "";
[Space]
[Text("[Line(\"#ff00ff\", Thickness = 9)]")]
[Line("#ff00ff", Thickness = 9)]
[UsingCustomProperty]
[SerializeField]
private string test11 = "";
[Space]
[Text("[Line(\"#ff00ff\", Thickness = 10)]")]
[Line("#ff00ff", Thickness = 10)]
[...]
```



FEATURES

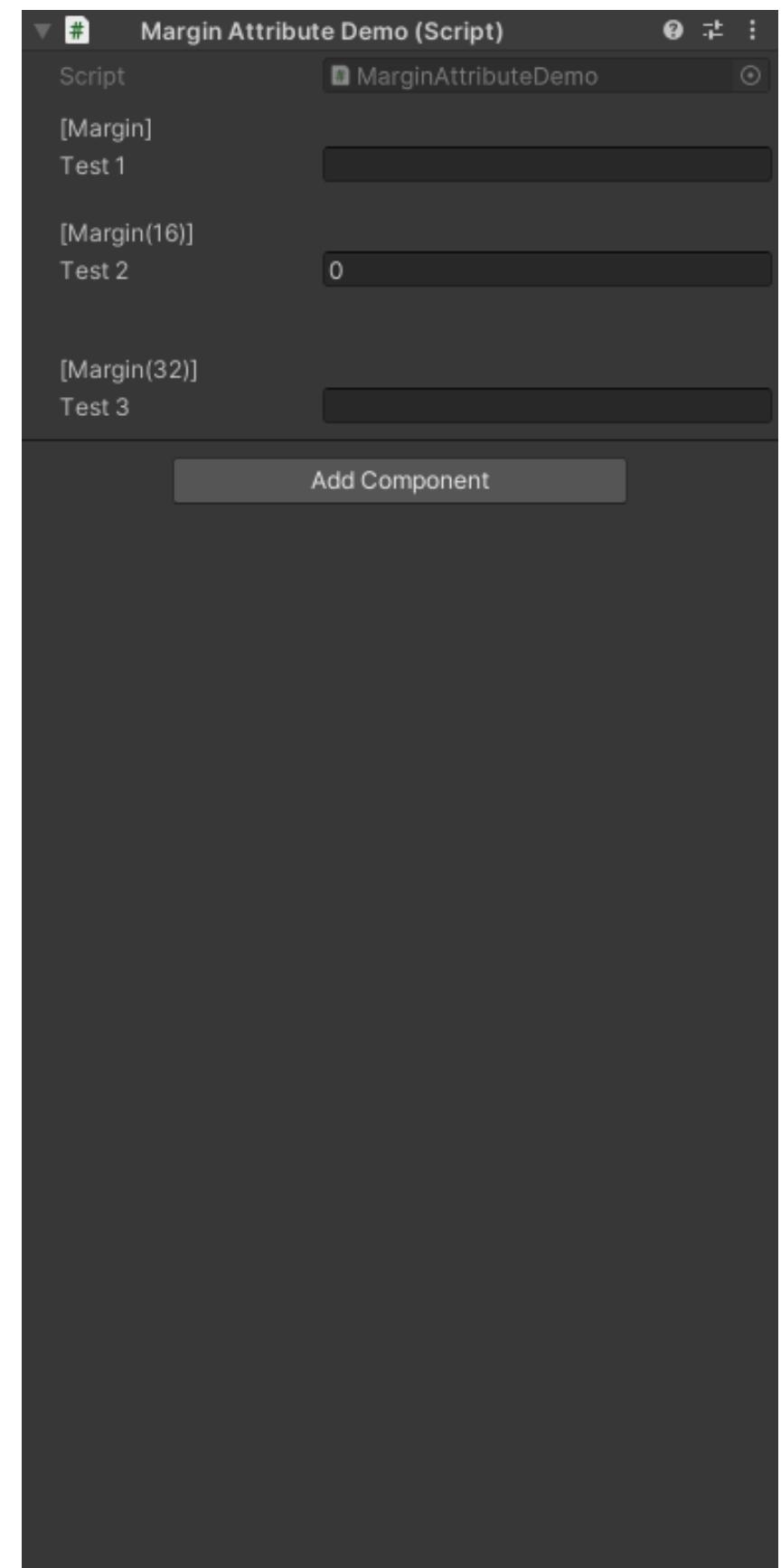
기능 소개

[Margin]

Draws a margin. You can specify the height of the margin as a parameter.

여백을 그립니다. 매개변수로 여백의 높이를 지정할 수 있습니다.

```
[Margin]  
[Text(" [Margin] ")]  
[UsingCustomProperty]  
[SerializeField]  
private string test1 = "";  
  
[Margin(16)]  
[Text(" [Margin(16)] ")]  
[UsingCustomProperty]  
[SerializeField]  
private int test2 = 0;  
  
[Margin(32)]  
[Text(" [Margin(32)] ")]  
[UsingCustomProperty]  
[SerializeField]  
private string test3 = "";
```



FEATURES

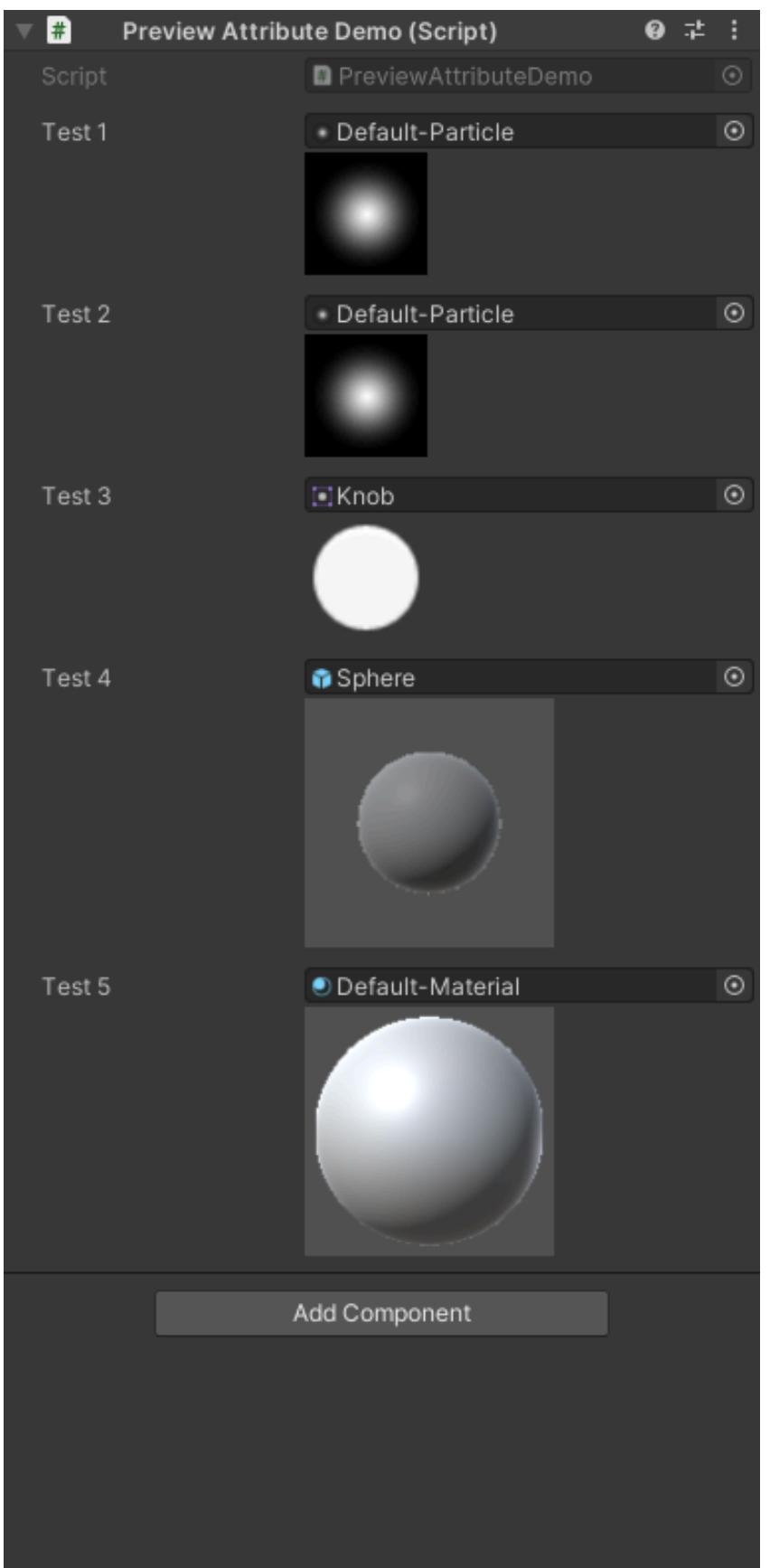
기능 소개

[Preview]

Draws a preview for 'Object' type fields containing images such as Texture, Sprite, GameObject, and Material.

Texture, Sprite, GameObject, Material 등 이미지가 존재하는 'Object' 타입 필드의 미리 보기 를 그립니다.

```
[Space]  
[PropertyField]  
[Preview]  
[UsingCustomProperty]  
[SerializeField]  
private Texture test1 = null;  
[Space]  
[PropertyField]  
[Preview]  
[UsingCustomProperty]  
[SerializeField]  
private Texture2D test2 = null;  
[Space]  
[PropertyField]  
[Preview]  
[UsingCustomProperty]  
[SerializeField]  
private Sprite test3 = null;  
[Space]  
[PropertyField]  
[Preview]  
[UsingCustomProperty]  
[SerializeField]  
private GameObject test4 = null;  
[Space]  
[PropertyField]  
[Preview]  
[UsingCustomProperty]  
[SerializeField]  
private Material test5 = null;
```



FEATURES

기능 소개

[Text]

Draws text. You can specify the color, alignment position, style, and font size of the text as parameters.

텍스트를 그립니다. 매개변수로 텍스트의 색상과 정렬 위치, 스타일, 폰트 크기를 지정할 수 있습니다.

```
[Space]
[Text("Text(\"Hello World!\")")]
[Text("Hello World!")]
[UsingCustomProperty]
[SerializeField]
private string test1 = "";

[Space]
[Text("Text(\"<color=#FF0000>Hello World!</color>\")", RichText = false)]
[Text("<color=#FF0000>Hello World!</color>")]

[UsingCustomProperty]
[SerializeField]
private string test2 = "";
...

[UsingCustomProperty]
[SerializeField]
private string test4 = "";

[Space]
[Text("Text(\"Hello World!\", TextAnchor.UpperLeft, Height = 36f)")]
[Text("Hello World!", TextAnchor.UpperLeft, Height = 36f)]
[UsingCustomProperty]
[SerializeField]
private string test5 = "";
...

[Space]
[Text("Text(\"<b>Hello World!</b>\")", RichText = false)]
[Text("<b>Hello World!</b>")]

[UsingCustomProperty]
[SerializeField]
private string test8 = "";
...

[Space]
[Text("Text(\"Hello World!\", FontSize = 24)")]
[Text("Hello World!", FontSize = 24)]
[UsingCustomProperty]
[SerializeField]
private string test11 = "";
...
```

Text Attribute Demo (Script)

Script TextAttributeDemo

Test 1 [Text("Hello World!")] Hello World!

Test 2 [Text("<color=#FF0000>Hello World!</color>")] Hello World!

Test 3 [Text("<color=#00FF00>Hello World!</color>")] Hello World!

Test 4 [Text("<color=#0000FF>Hello World!</color>")] Hello World!

Test 5 [Text("Hello World!", TextAnchor.UpperLeft, Height = 36f)] Hello World!

Test 6 [Text("Hello World!", TextAnchor.MiddleCenter, Height = 36f)] Hello World!

Test 7 [Text("Hello World!", TextAnchor.LowerRight, Height = 36f)] Hello World!

Test 8 [Text("Hello World!")] Hello World!

Test 9 [Text("<i>Hello World!</i>")] Hello World!

Test 10 [Text("<i>Hello World!</i>")] Hello World!

Test 11 [Text("Hello World!", FontSize = 24)] Hello World!

FEATURES

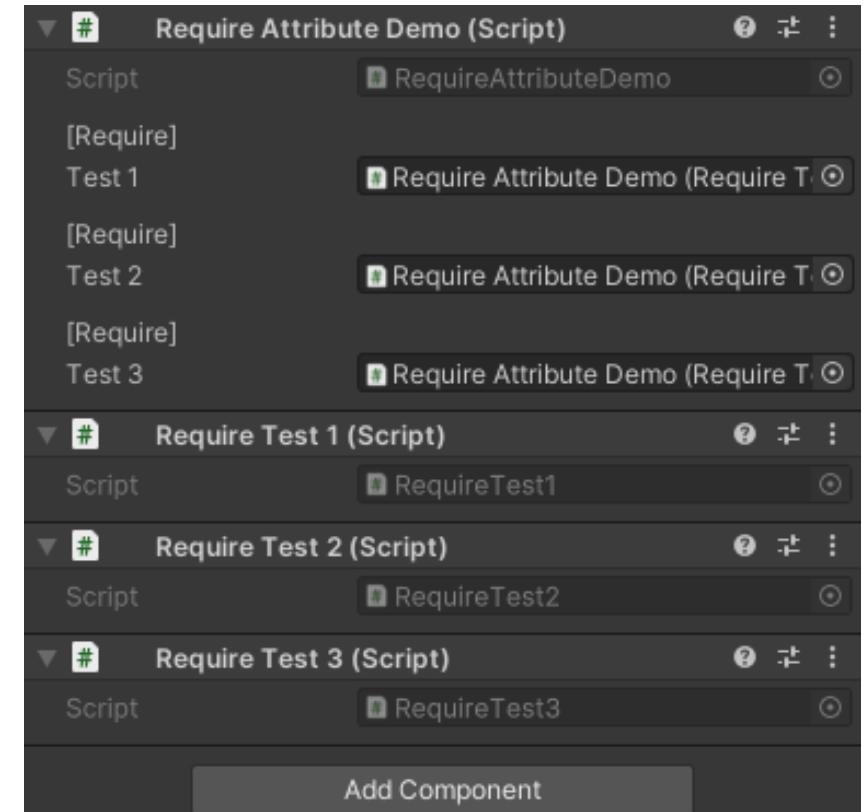
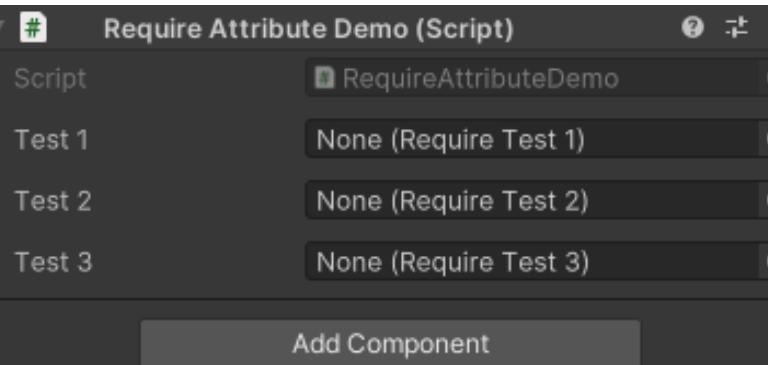
기능 소개

[Require]

Automatically adds and assigns the required components.

필수적인 Component를 자동 추가 및 할당합니다.

```
[Space]  
[Text( "[Require]" )]  
  
[Require]  
  
[UsingCustomProperty]  
  
[SerializeField]  
  
private RequireTest1 test1 = null;  
  
[Space]  
[Text( "[Require]" )]  
  
[Require]  
  
[UsingCustomProperty]  
  
[SerializeField]  
  
private RequireTest2 test2 = null;  
  
[Space]  
[Text( "[Require]" )]  
  
[Require]  
  
[UsingCustomProperty]  
  
[SerializeField]  
  
private RequireTest3 test3 = null;
```



THANK YOU;

CONTACT

 warmth.giver@gmail.com

 [GitHub](#)

Zion Lee