

ENIV INSPECTOR KIT

Eniv Inspector Kit is an easy-to-use custom inspector kit for Unity. It consists of many attributes to customize and enhance the Unity inspector interface.

→ Image Attribute

```
using UnityEngine;
using EnivStudios.EnivInspector;

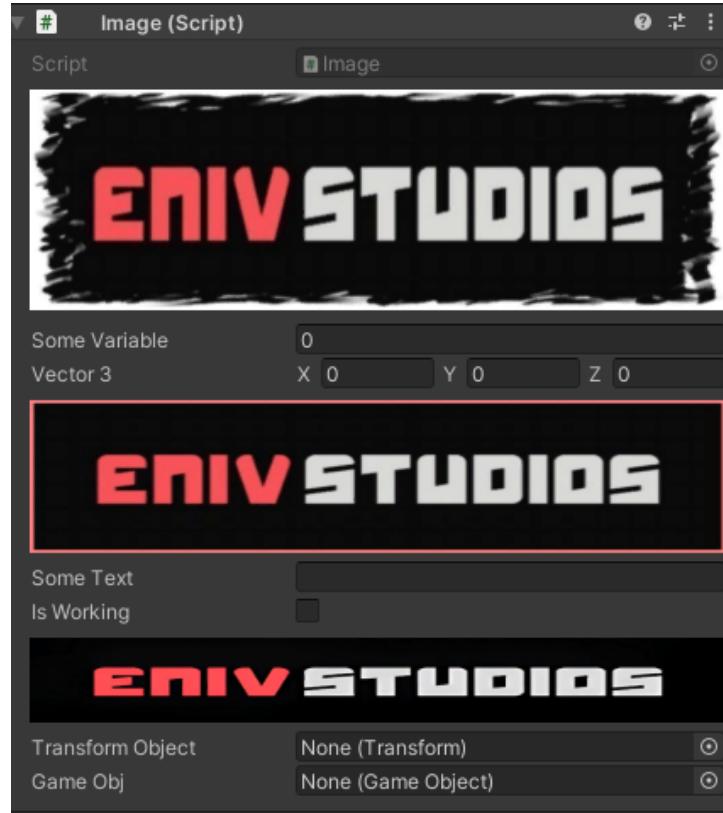
public class Image : MonoBehaviour
{
    [Image(imagePath: "Assets/StylizeC#ScriptPro/Images/Logo_1.png", imageHeight: 130f, spaceAbove: 5f,
    spaceBelow: 8f)]
    [SerializeField] private int someVariable;
    [SerializeField] private Vector3 vector3;

    [Image(imagePath: "Assets/StylizeC#ScriptPro/Images/Logo_2.png", imageHeight: 90f, spaceAbove: 5f,
    spaceBelow: 5f)]
    [SerializeField] private string someText;
    [SerializeField] private bool isWorking;

    [Image(imagePath: "Assets/StylizeC#ScriptPro/Images/Logo_3.png", imageHeight: 50f, spaceAbove: 5f,
    spaceBelow: 5f)]
    [SerializeField] private Transform transformObject;
    [SerializeField] private GameObject gameObj;
}
```

This attribute allows you to put different images with different sizes in the inspector.

- **imagePath:** is used to assign location of image.
- **imageHeight:** is used to adjust height of image.
- **spaceAbove** and **spaceBelow:** refers to spacing above and below the image.



→ UpperInfoBox Attribute

```

using UnityEngine;
using EnivStudios.EnivInspector;

public class UpperInfoBox : MonoBehaviour
{
    [UpperInfoBox("This variable calculates cat speed", MessageType.Info, spaceAbove: 2f, spaceBelow: 2f)]
    [SerializeField] private int catRunSpeed;

    [UpperInfoBox("Make sure dog speed doesn't increase too much!", MessageType.Warning, spaceAbove: 2f, spaceBelow: 2f)]
    [SerializeField] private int dogRunSpeed;

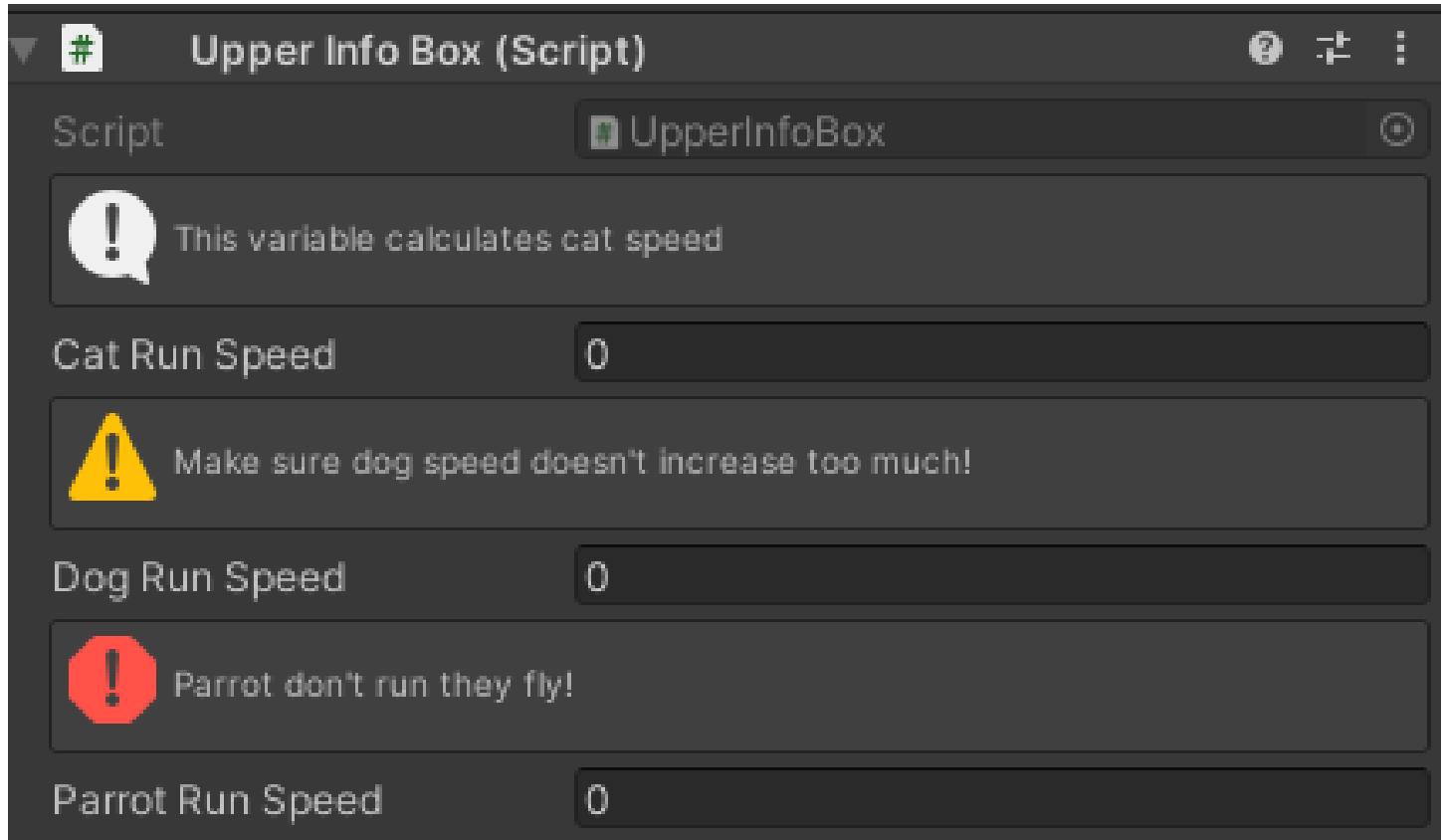
    [UpperInfoBox("Parrot don't run they fly!", MessageType.Error, spaceAbove: 2f, spaceBelow: 2f)]
    [SerializeField] private int parrotRunSpeed;
}

```

The `UpperInfoBox` attribute enables the display of warning, info, or error messages above property fields in the Unity Inspector.

- **message:** The message to display.
- **MessageType:** Specifies the message's appearance type (warning, info, or error).

- **spaceAbove**: Adjusts space above the message.
- **spaceBelow**: Adjusts space below the message.



→ LowerInfoBox Attribute

```
using UnityEngine;
using EnvStudios.EnvInspector;

public class LowerInfoBox : MonoBehaviour
{
    [LowerInfoBox("This variable calculates cat speed", MessageType.Info, spaceAbove: 2f, spaceBelow: 2f)]
    [SerializeField] private int catRunSpeed;

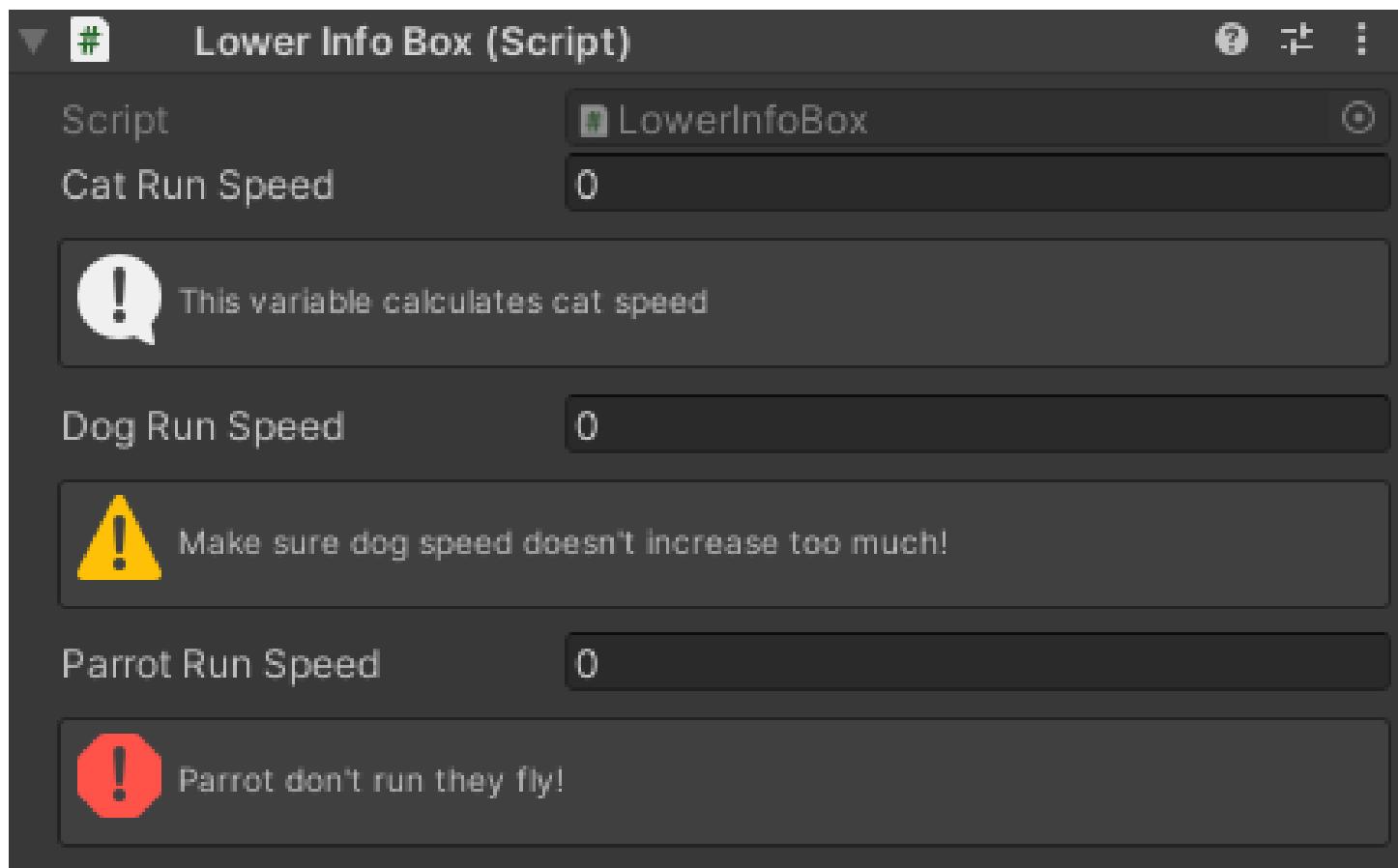
    [LowerInfoBox("Make sure dog speed doesn't increase too much!", MessageType.Warning, spaceAbove: 2f,
    spaceBelow: 2f)]
    [SerializeField] private int dogRunSpeed;

    [LowerInfoBox("Parrot don't run they fly!", MessageType.Error, spaceAbove: 2f, spaceBelow: 2f)]
    [SerializeField] private int parrotRunSpeed;
}
```

The LowerInfoBox attribute enables the display of warning, info, or error messages below property fields in the Unity Inspector.

- **message**: The message to display.

- **MessageType**: Specifies the message's appearance type (warning, info, or error).
- **spaceAbove**: Adjusts space above the message.
- **spaceBelow**: Adjusts space below the message.



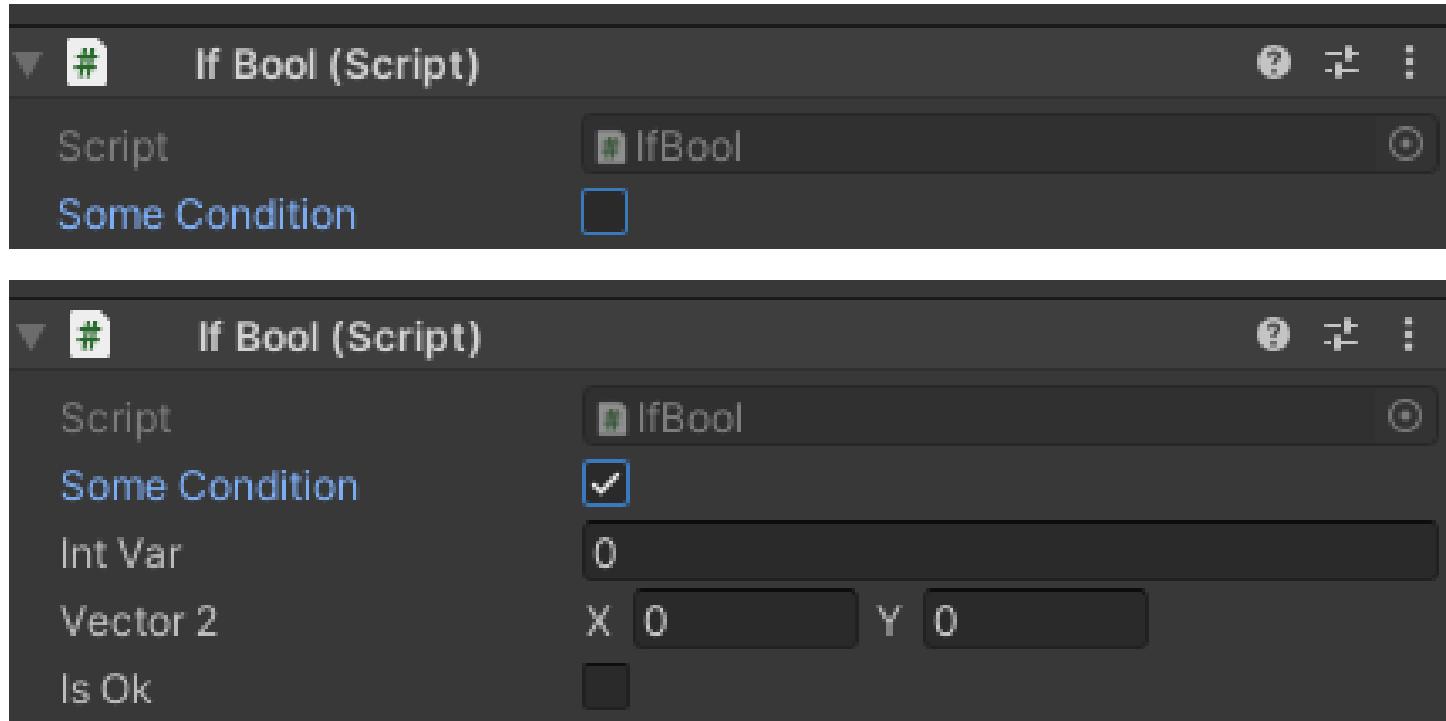
→ IfBool Attribute

```
using UnityEngine;
using EnivStudios.EnivInspector;
@ Unity Script (1 asset reference) | 0 references
public class IfBool : MonoBehaviour
{
    [SerializeField] private bool someCondition;

    [IfBool("someCondition", showProperty: true)][SerializeField] private int intVar;
    [IfBool("someCondition", showProperty: true)][SerializeField] private Vector2 vector2;
    [IfBool("someCondition", showProperty: true)][SerializeField] private bool isOk;
}
```

This attribute allows for dynamic control over the visibility of properties within the Unity Inspector based on the value of a boolean condition.

- **boolFieldName**: The name of the boolean field that determines the visibility of properties in the Unity Inspector.
- **showProperty**: A boolean value used to toggle the visibility of properties in the Unity Inspector.



→ ToggleLeft Attribute

```

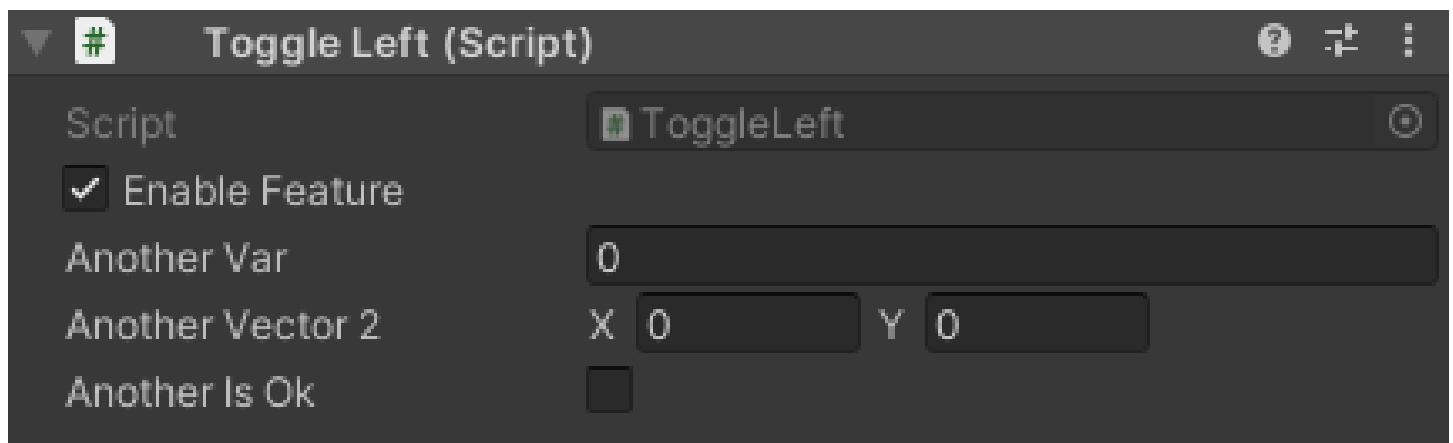
using EnvStudios.EnvInspector;
using UnityEngine;

public class ToggleLeft : MonoBehaviour
{
    [ToggleLeft]
    [SerializeField] private bool enableFeature;

    [IfBool("enableFeature", showProperty: true)][SerializeField] private int anotherVar;
    [IfBool("enableFeature", showProperty: true)][SerializeField] private Vector2 anotherVector2;
    [IfBool("enableFeature", showProperty: true)][SerializeField] private bool anotherIsOk;
}

```

This attribute interchanges the position of boolean variable name and checkbox in inspector and works well with IfBool attribute.



→ BoolInfoBox Attribute

```
using EnvStudios.EnvInspector;
using UnityEngine;

public class BoolInfoBox : MonoBehaviour
{
    [SerializeField] private bool someCondition;

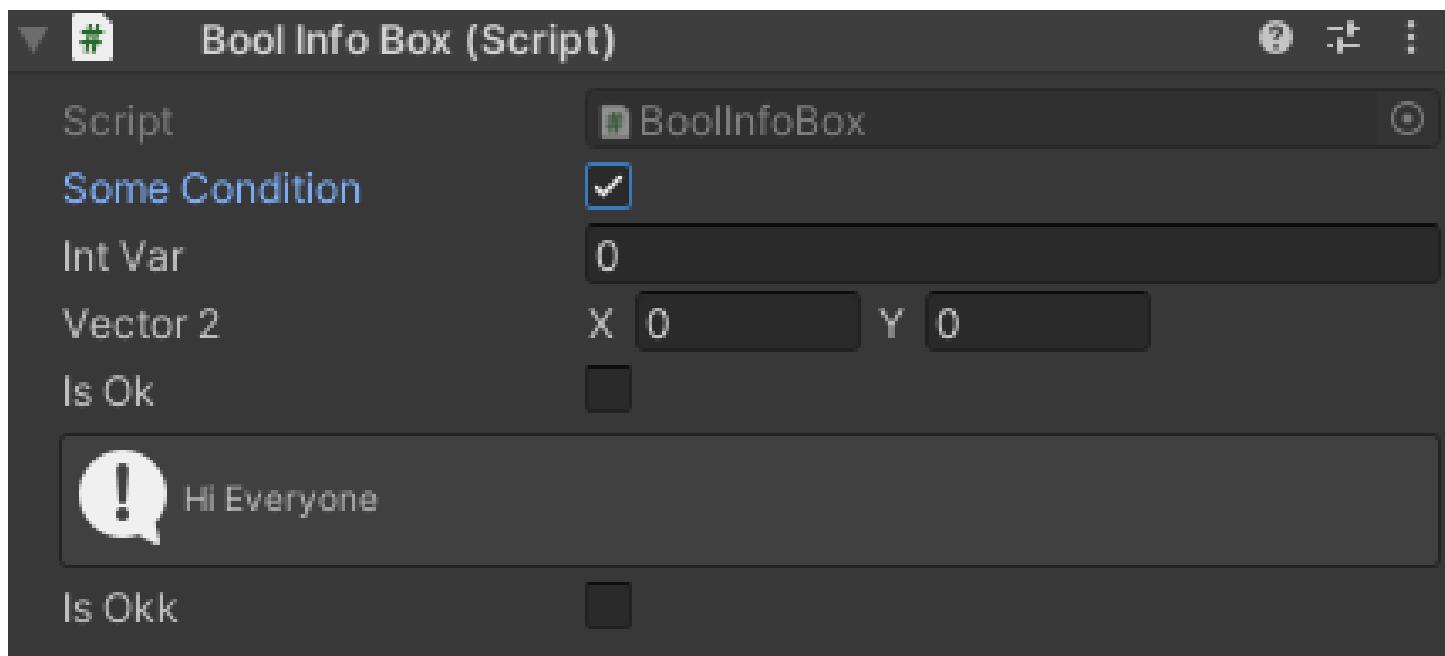
    [IfBool("someCondition", showProperty: true)][SerializeField] private int intVar;
    [IfBool("someCondition", showProperty: true)][SerializeField] private Vector2 vector2;
    [IfBool("someCondition", showProperty: true)][SerializeField] private bool isOk;

    [BoolInfoBox("Hi Everyone", "someCondition", true, spaceAbove:2, spaceBelow: 2, MessageType.Info)]
    [SerializeField] private bool isOkk;
}
```

The BoolInfoBox attribute displays information in an info box in the Unity Inspector based on a boolean condition.

- **message:** The message to display.
- **boolFieldName:** The name of the boolean field that determines the visibility of properties in the Unity Inspector.
- **expectedValue:** A boolean value used to toggle the visibility of properties in the Unity Inspector.

- **spaceAbove**: Adjusts space above the message.
- **spaceBelow**: Adjusts space below the message.
- **MessageType**: Specifies the message's appearance type (warning, info, or error).



→ IfEnum Attribute

```
using UnityEngine;
using EnivStudios.EnivInspector;

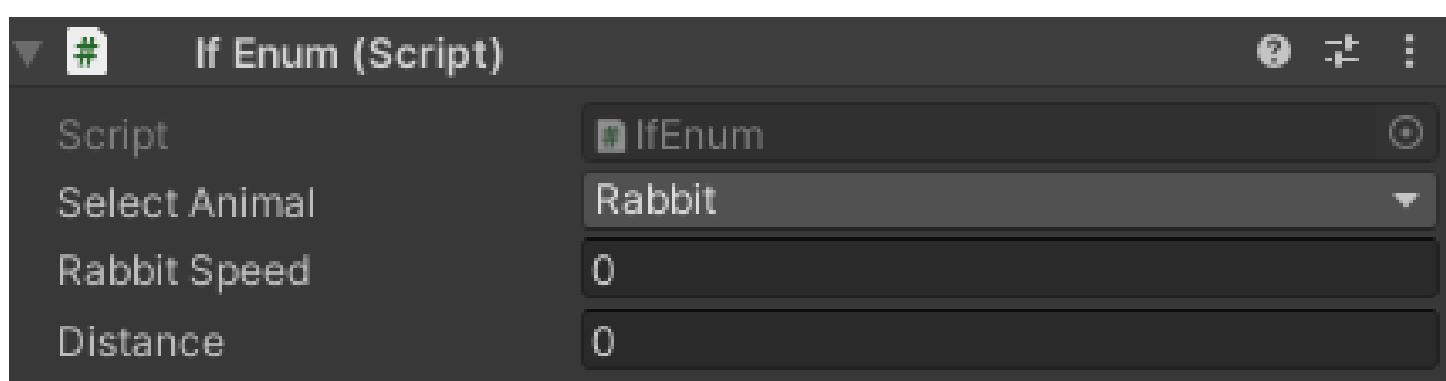
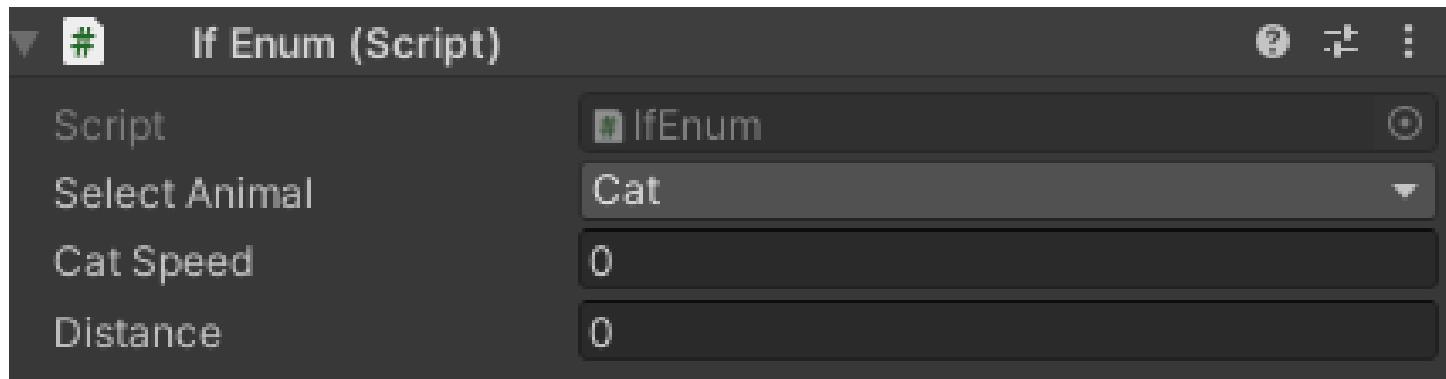
public class IfEnum : MonoBehaviour
{
    private enum AnimalType { Cat, Dog, Rabbit, Parrot }
    [SerializeField] private AnimalType selectAnimal;

    [IfEnum("selectAnimal", (int)AnimalType.Cat)] [SerializeField] private int catSpeed;
    [IfEnum("selectAnimal", (int)AnimalType.Dog)] [SerializeField] private int dogSpeed;
    [IfEnum("selectAnimal", (int)AnimalType.Rabbit)] [SerializeField] private int rabbitSpeed;
    [IfEnum("selectAnimal", (int)AnimalType.Parrot)] [SerializeField] private int parrotSpeed;

    [IfEnum("selectAnimal", (int)AnimalType.Cat, (int)AnimalType.Dog, (int)AnimalType.Rabbit,
    (int)AnimalType.Parrot)] [SerializeField] private int distance;
}
```

This attribute facilitates dynamic control over property visibility within the Unity Inspector based on the selected enum value.

- **enumFieldName:** The name of the enum field whose value determines property visibility in the Unity Inspector.
- **displayIndices:** An array of integers representing the indices of enum values for which the associated properties should be visible in the Unity Inspector.



→ **EnumInfoBox Attribute**

```
using EnvStudios.EnvInspector;
using UnityEngine;

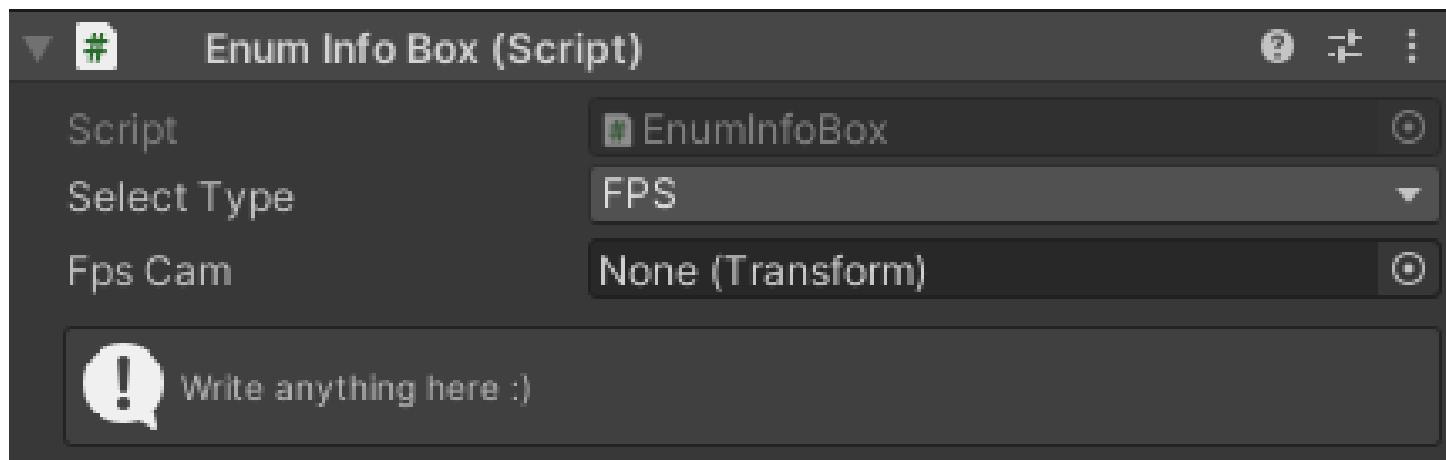
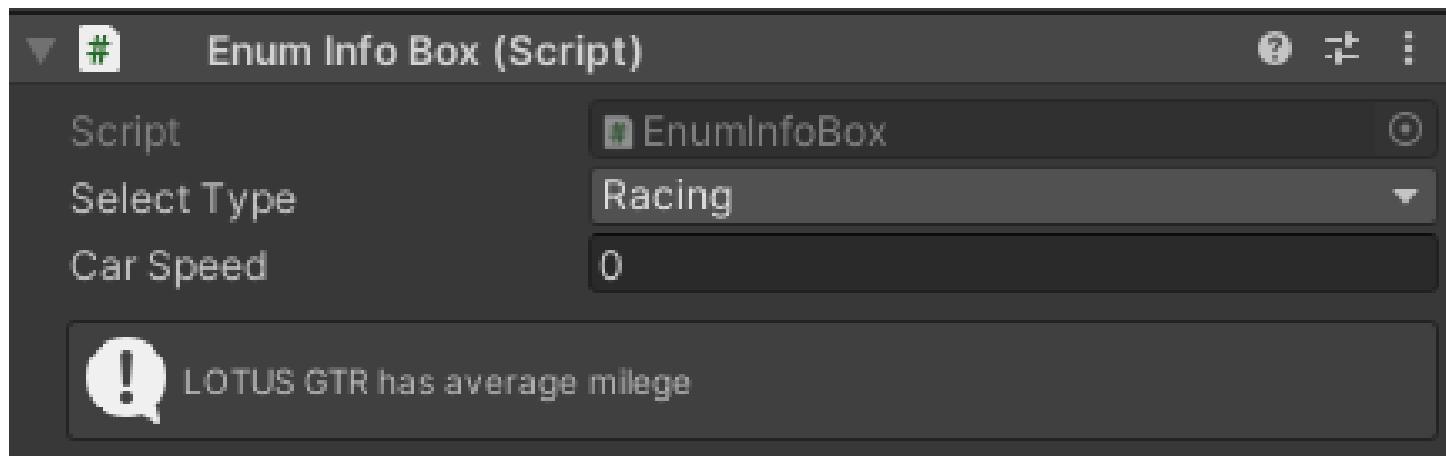
public class EnumInfoBox : MonoBehaviour
{
    private enum GameType { FPS, TPS, Racing, Arcade }
    [SerializeField] private GameType selectType;

    [EnumInfoBox("LOTUS GTR has average mileage", spaceAbove: 2, spaceBelow: 0, MessageType.Info,
    "selectType", (int)GameType.Racing)]
    [SerializeField] private int carSpeed;

    [EnumInfoBox("Write anything here :) ", spaceAbove: 2, spaceBelow: 0, MessageType.Info, "selectType",
    (int)GameType.FPS)]
    [SerializeField] private Transform fpsCam;
}
```

This attribute provides a way to display information in an info box within the Unity Inspector based on the selected enum value.

- **message:** The message to display.
- **spaceAbove:** Adjusts space above the message.
- **spaceBelow:** Adjusts space below the message.
- **MessageType:** Specifies the message's appearance type (warning, info, or error).
- **enumFieldName:** The name of the enum field whose value determines property visibility in the Unity Inspector.
- **displayIndices:** An array of integers representing the indices of enum values for which the associated properties should be visible in the Unity Inspector.



→ Title Attribute



```
using UnityEngine;
using EnivStudios.EnivInspector;

public class Title : MonoBehaviour
{
    [Title(null, spaceAbove: 8f, spaceBelow: 8f, lineColor: CommonColors.White)]
    public int someVar_1;
    public int someVar_2;
    public int someVar_3;

    [Title("Simple Title", spaceAbove: 8f, spaceBelow: 8f, lineColor: CommonColors.White)]
    public int anotherVariables_1;
    public int anotherVariables_2;
    public int anotherVariables_3;

    [Title("Title with specific line color", spaceAbove: 8f, spaceBelow: 8f, lineColor: CommonColors.Red)]
    public int anotherVariables_4;
    public int anotherVariables_5;
    public int anotherVariables_6;

    [Title("Title with specific heading color", spaceAbove: 8f, spaceBelow: 8f, headingColor: CommonColors.Yellow)]
    public int anotherVariables_7;
    public int anotherVariables_8;
    public int anotherVariables_9;

    [Title("Title with specific heading and line color", spaceAbove: 8f, spaceBelow: 8f, lineColor: CommonColors.Aqua, headingColor: CommonColors.Green)]
    public int anotherVariables_10;
    public int anotherVariables_11;
    public int anotherVariables_12;
}
```

The **Title** attribute is used to display a title above a property field in the Unity Inspector, with a line that seamlessly passes through the title without intersecting it.

- **heading:** Specifies the title to be displayed.
- **spaceAbove:** Controls the space above the title.
- **spaceBelow:** Controls the space below the title.
- **lineColor:** Sets the color of the line.
- **headingColor:** Sets the color of the title text.

Title (Script)	
Script	Title
Some Var_1	0
Some Var_2	0
Some Var_3	0
<hr/> Simple Title <hr/>	
Another Variables_1	0
Another Variables_2	0
Another Variables_3	0
<hr/> Title with specific line color <hr/>	
Another Variables_4	0
Another Variables_5	0
Another Variables_6	0
<hr/> Title with specific heading color <hr/>	
Another Variables_7	0
Another Variables_8	0
Another Variables_9	0
<hr/> Title with specific heading and line color <hr/>	
Another Variables_10	0
Another Variables_11	0
Another Variables_12	0

→ BGTitle Attribute

```
using UnityEngine;
using EnivStudios.EnivInspector;

public class BGTitle : MonoBehaviour
{
    [BGTitle("Title With InfoBox Color And No Spacing", colorScheme: ColorScheme.InfoBox, centerTitle: false, spaceAbove:0f, spaceBelow:0f, boxHeight: 2.2f)]
    public int someVar_1;
    public int someVar_2;
    public int someVar_3;

    [BGTitle("Title With InfoBox Color And Spacing", colorScheme: ColorScheme.InfoBox, centerTitle: false, spaceAbove: 15f, spaceBelow: 15f, boxHeight: 2.2f)]
    public int someVar_4;
    public int someVar_5;
    public int someVar_6;

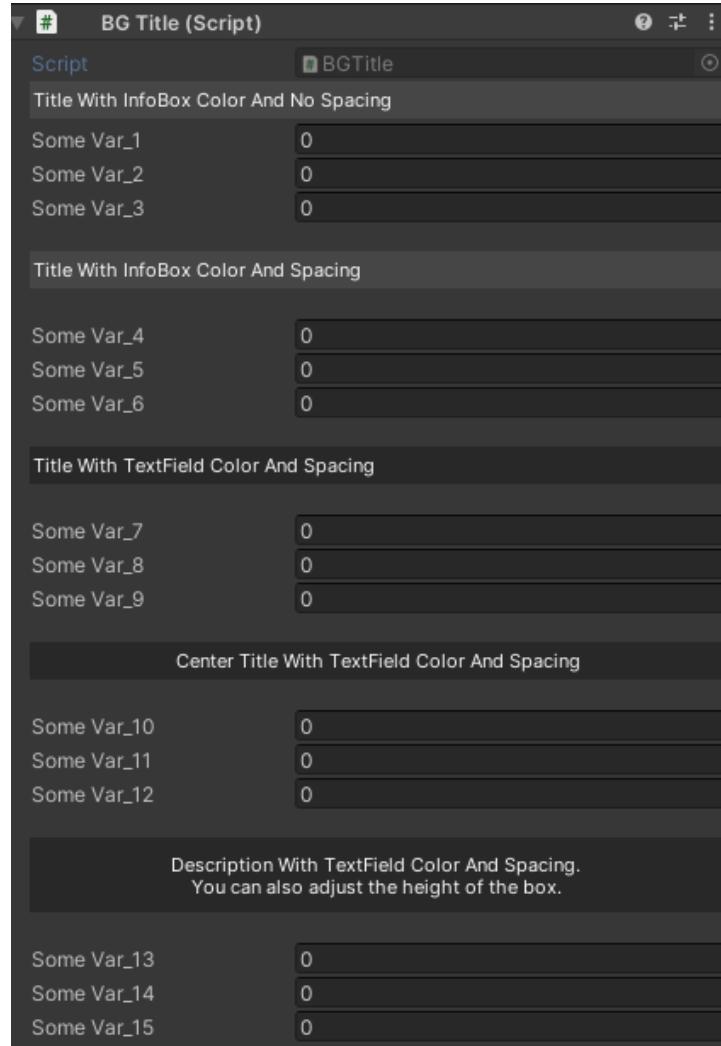
    [BGTitle("Title With TextField Color And Spacing", colorScheme: ColorScheme.TextField, centerTitle: false, spaceAbove: 15f, spaceBelow: 15f, boxHeight: 2.2f)]
    public int someVar_7;
    public int someVar_8;
    public int someVar_9;

    [BGTitle("Center Title With TextField Color And Spacing", colorScheme: ColorScheme.TextField, centerTitle: true, spaceAbove: 15f, spaceBelow: 15f, boxHeight: 2.2f)]
    public int someVar_10;
    public int someVar_11;
    public int someVar_12;

    [BGTitle("Description With TextField Color And Spacing.\nYou can also adjust the height of the box.", colorScheme: ColorScheme.TextField, centerTitle: true, spaceAbove: 15f, spaceBelow: 15f, boxHeight: 4.4f)]
    public int someVar_13;
    public int someVar_14;
    public int someVar_15;
}
```

The **BGTitle** attribute is used to display a title above a property field in the Unity Inspector with a background.

- **heading:** Specifies the title to be displayed.
- **centerTitle:** Determines whether the title is centered.
- **colorScheme:** Specifies the color scheme for the background (TextField or InfoBox).
- **spaceAbove:** Controls the space above the title background.
- **spaceBelow:** Controls the space below the title background.
- **boxHeight:** Sets the height of the title background box.

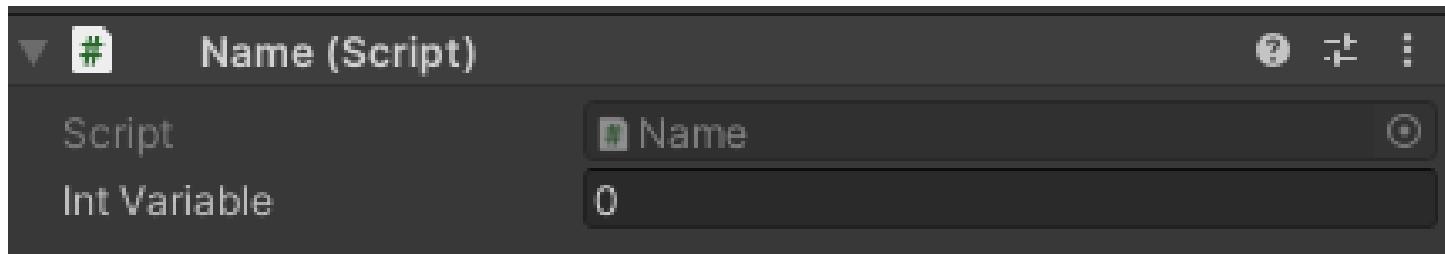


→ Name Attribute

```
using UnityEngine;
using EnvStudios.EnvInspector;

① Unity Script (1 asset reference) | 0 references
public class Name : MonoBehaviour
{
    [Name("Int Variable")]
    [SerializeField] private int var;
}
```

This attribute is used to change the name of the variable.

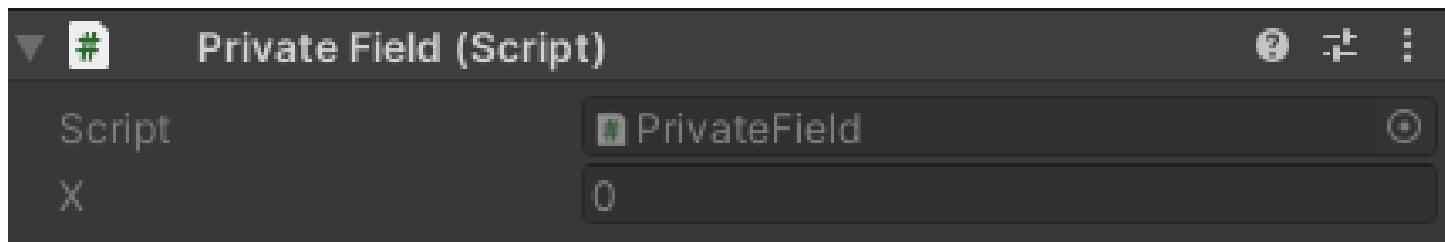


→ PrivateField Attribute

```
using EnvStudios.EnvInspector;
using UnityEngine;

⊕ Unity Script (1 asset reference) | 0 references
public class PrivateField : MonoBehaviour
{
    [SerializeField]
    [PrivateField] private int x;
}
```

The PrivateField attribute displays private fields in the Unity Inspector but they are not editable.

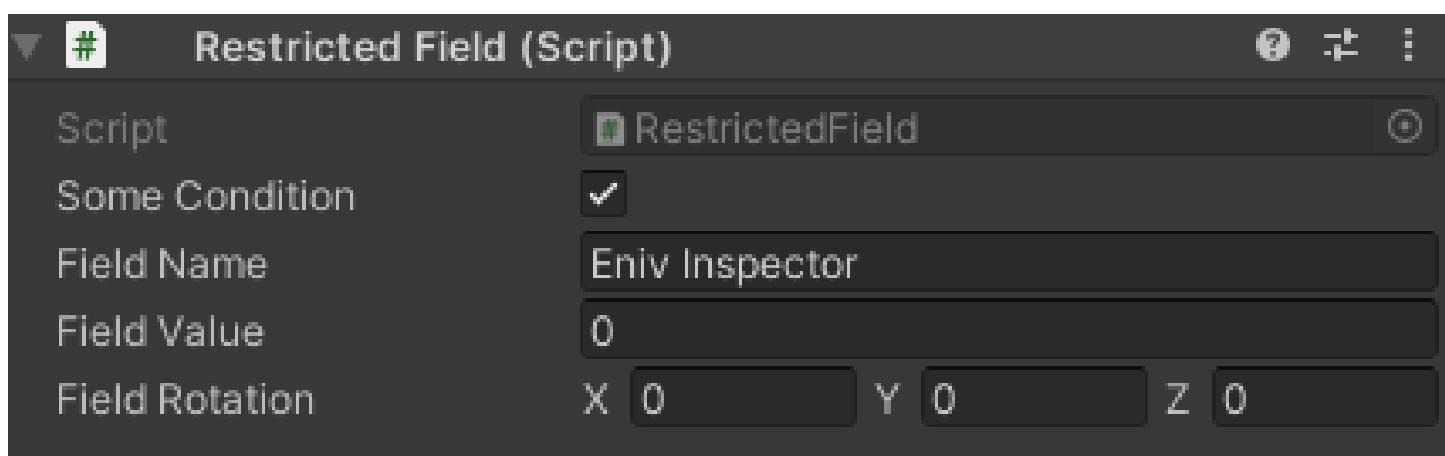
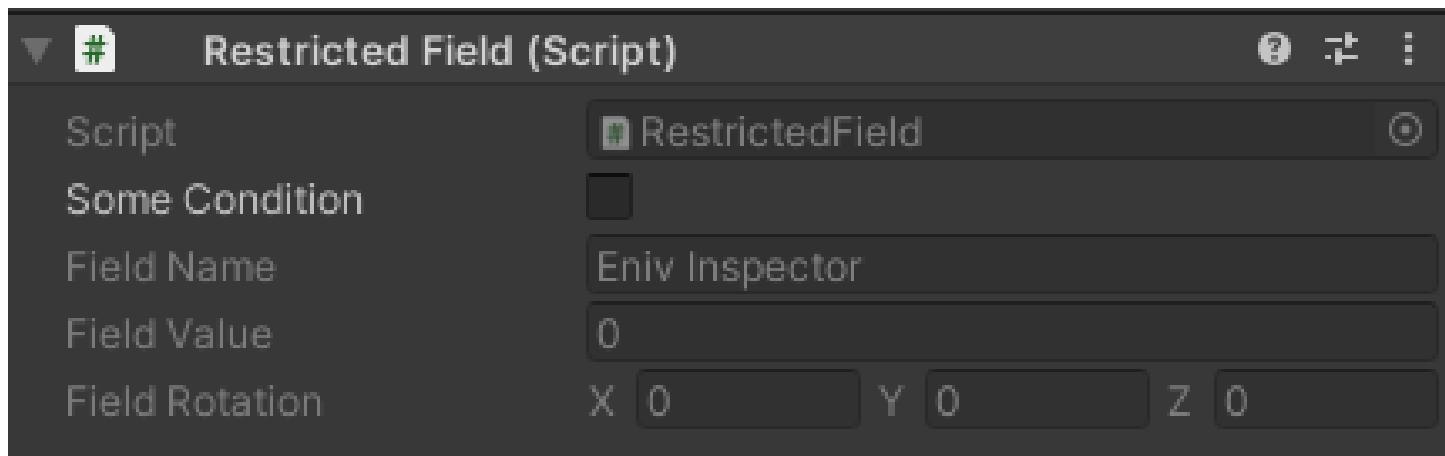


→ RestrictedField Attribute

```
using UnityEngine;
using EnivStudios.EnivInspector;
Unity Script (1 asset reference) | 0 references
public class RestrictedField : MonoBehaviour
{
    [SerializeField] private bool someCondition;

    [RestrictedField("someCondition")][SerializeField] private string fieldName;
    [RestrictedField("someCondition")][SerializeField] private int fieldValue;
    [RestrictedField("someCondition")][SerializeField] private Quaternion fieldRotation;
}
```

The `RestrictedField` attribute limits editing of property fields in the Unity Inspector based on a boolean condition.



→ **Vector2Slider Attribute**

```

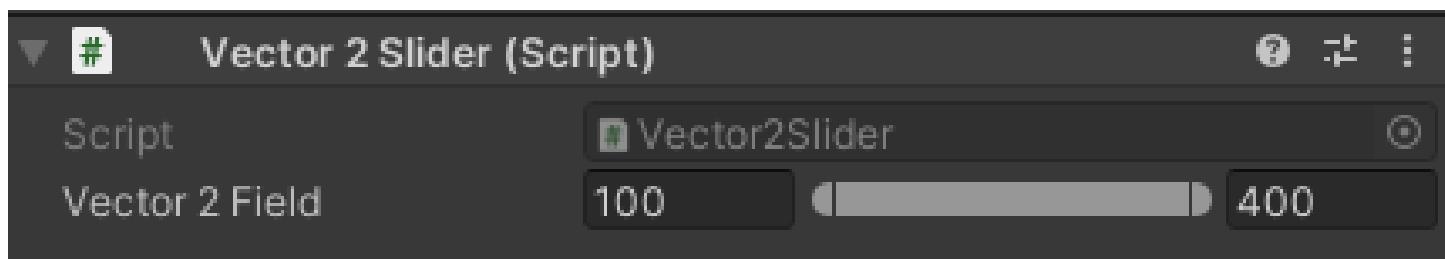
using UnityEngine;
using EnivStudios.EnivInspector;

Unity Script (1 asset reference) | 0 references
public class Vector2Slider : MonoBehaviour
{
    [Vector2Slider(xValue:100f,yValue:400f)]
    [SerializeField] private Vector2 vector2Field;
}

```

The Vector2Slider attribute enables easy adjustment of Vector2 properties within a specified range using sliders in the Unity Inspector.

- **xValue:** Sets the minimum value for the X-axis slider.
- **yValue:** Sets the maximum value for the Y-axis slider.



→ BoxGroup Attribute

```

using EnivStudios.EnivInspector;
using UnityEngine;

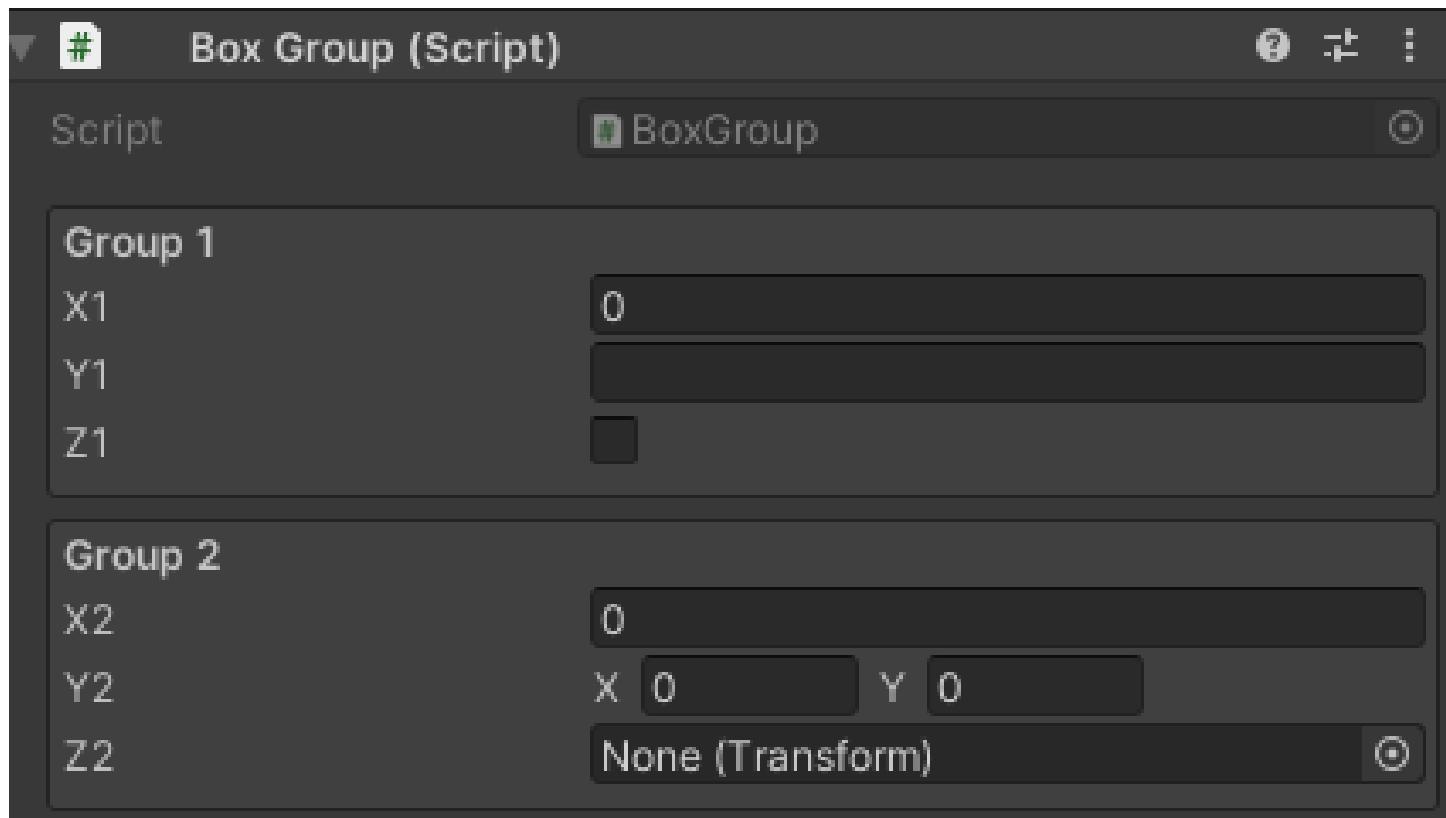
Unity Script (1 asset reference) | 0 references
public class BoxGroup : MonoBehaviour
{
    [BoxGroup("Group 1", spaceAbove:10f, spaceBelow:2f)][HideInInspector] public int x1;
    [BoxGroup("Group 1")][HideInInspector] public string y1;
    [BoxGroup("Group 1")][HideInInspector] public bool z1;

    [BoxGroup("Group 2", spaceAbove: 0f, spaceBelow: 10f)][HideInInspector] public float x2;
    [BoxGroup("Group 2")][HideInInspector] public Vector2 y2;
    [BoxGroup("Group 2")][HideInInspector] public Transform z2;
}

```

The **BoxGroup** attribute wraps all property fields in a box with a specified title in the Unity Inspector.

- **boxName:** Specifies the title of the box group.
- **spaceAbove:** Adjusts the space above the box group.
- **spaceBelow:** Adjusts the space below the box group.



→ **FoldoutGroup Attribute**

```

using EnvStudios.EnvInspector;
using UnityEngine;

Unity Script (1 asset reference) | 0 references
public class FoldoutGroup : MonoBehaviour
{
    [FoldoutGroup("Cat Properties", spaceAbove: 10f, spaceBelow: 0)]
    [HideInInspector] public string catName;

    [FoldoutGroup("Cat Properties")]
    [HideInInspector] public Color colorPicker;

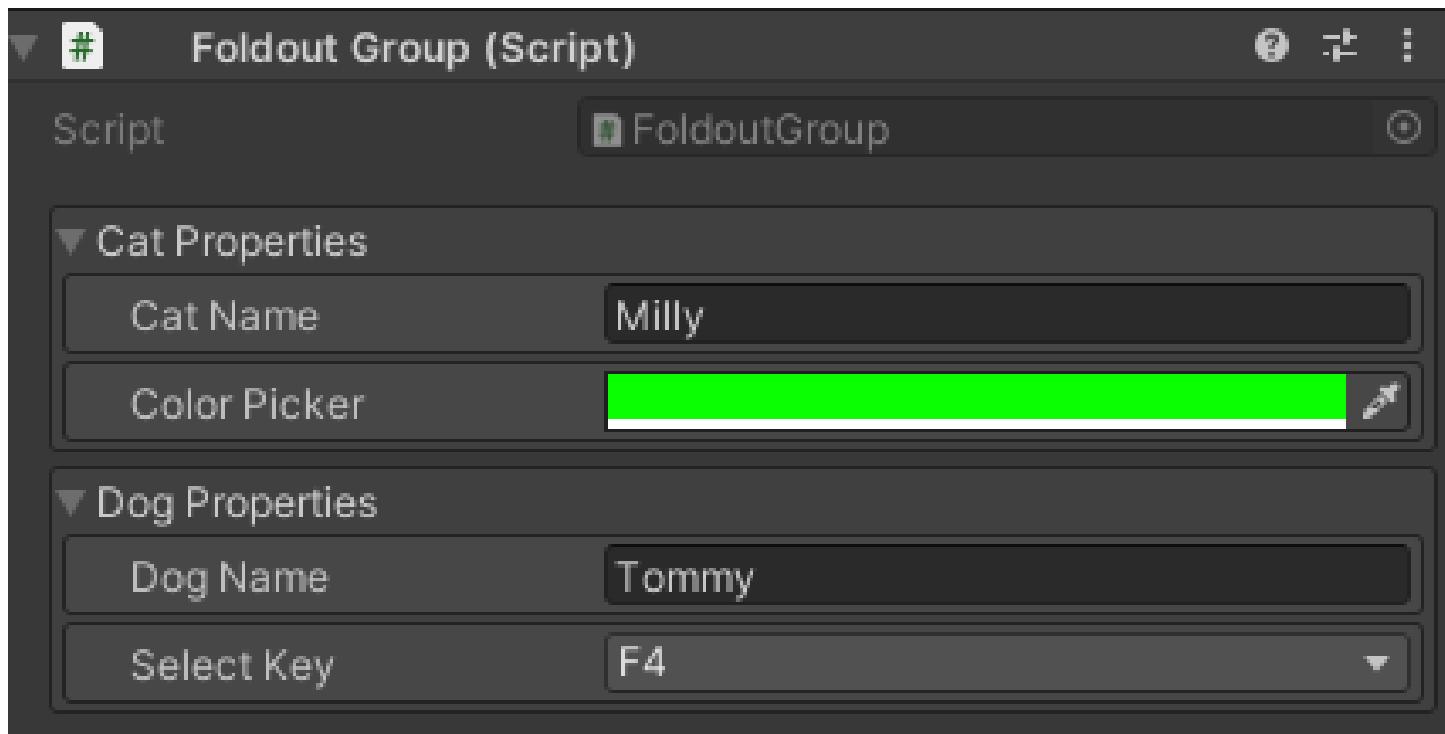
    [FoldoutGroup("Dog Properties", spaceAbove: 0, spaceBelow: 10f)]
    [HideInInspector] public string dogName;

    [FoldoutGroup("Dog Properties")]
    [HideInInspector] public KeyCode selectKey;
}

```

The `FoldoutGroup` attribute wraps all property fields in a foldout with a specified title in the Unity Inspector.

- **groupName:** Specifies the title of the foldout group.
- **spaceAbove:** Adjusts the space above the foldout group.
- **spaceBelow:** Adjusts the space below the foldout group.



→ Tab Attribute

```
✓ using EnvStudios.EnvInspector;
└ using UnityEngine;

➊ Unity Script (1 asset reference) | 0 references
✓ public class Tab : MonoBehaviour
{
    [Tab("Tab 1", spaceAbove: 10f)]
    [HideInInspector] [SerializeField] private string property1;

    [Tab("Tab 1")]
    [HideInInspector] public int property2;

    [Tab("Tab 2", spaceAbove: 10f)]
    [HideInInspector] public float property3;

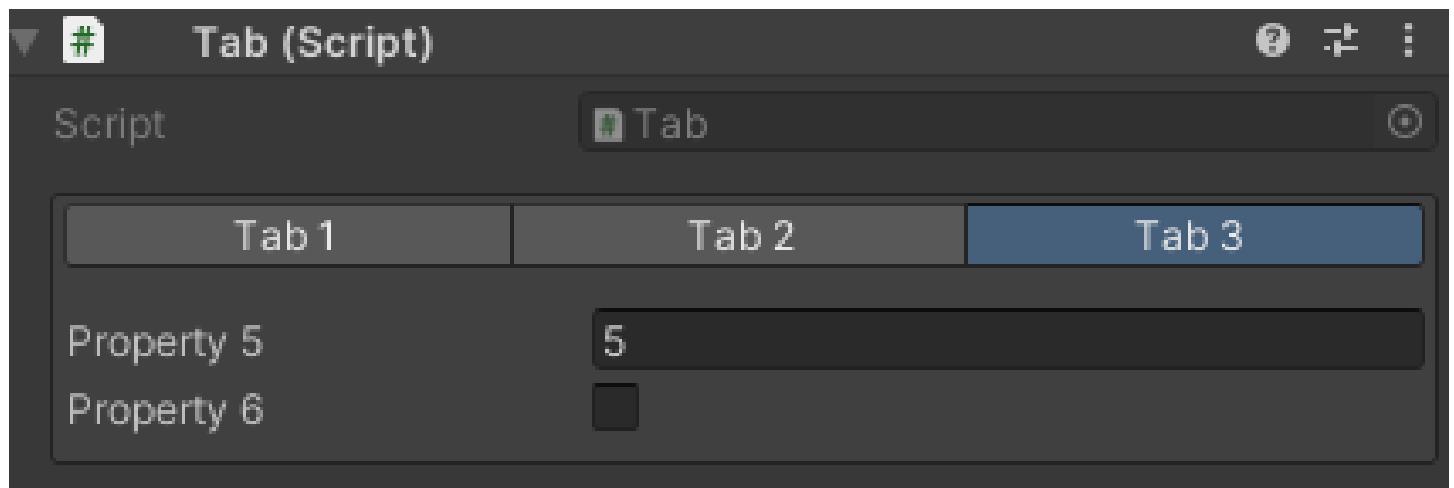
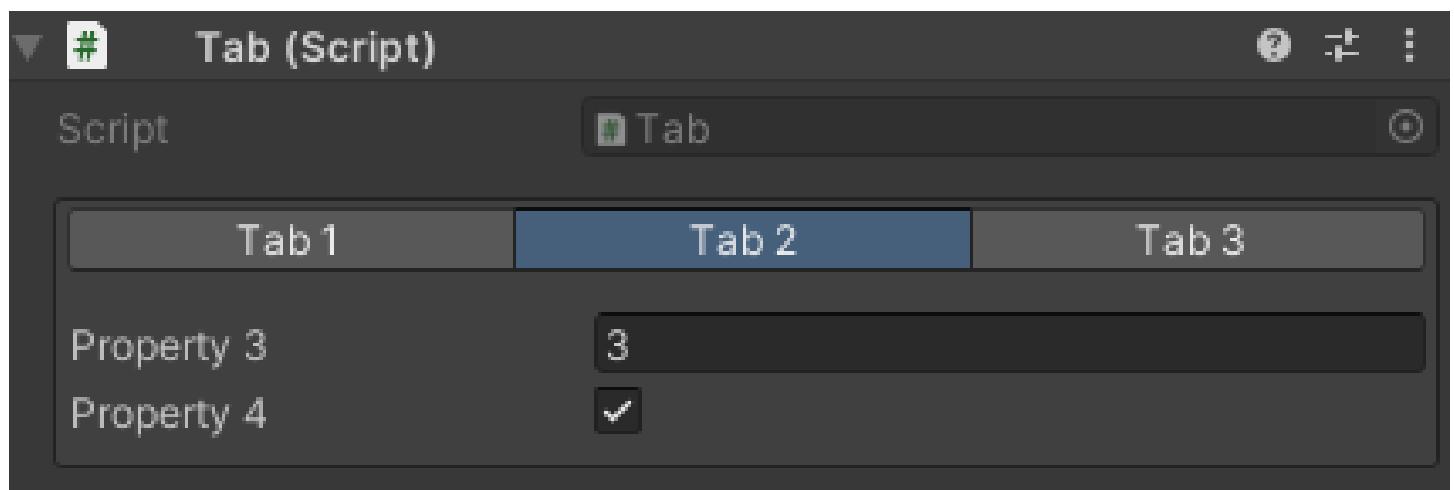
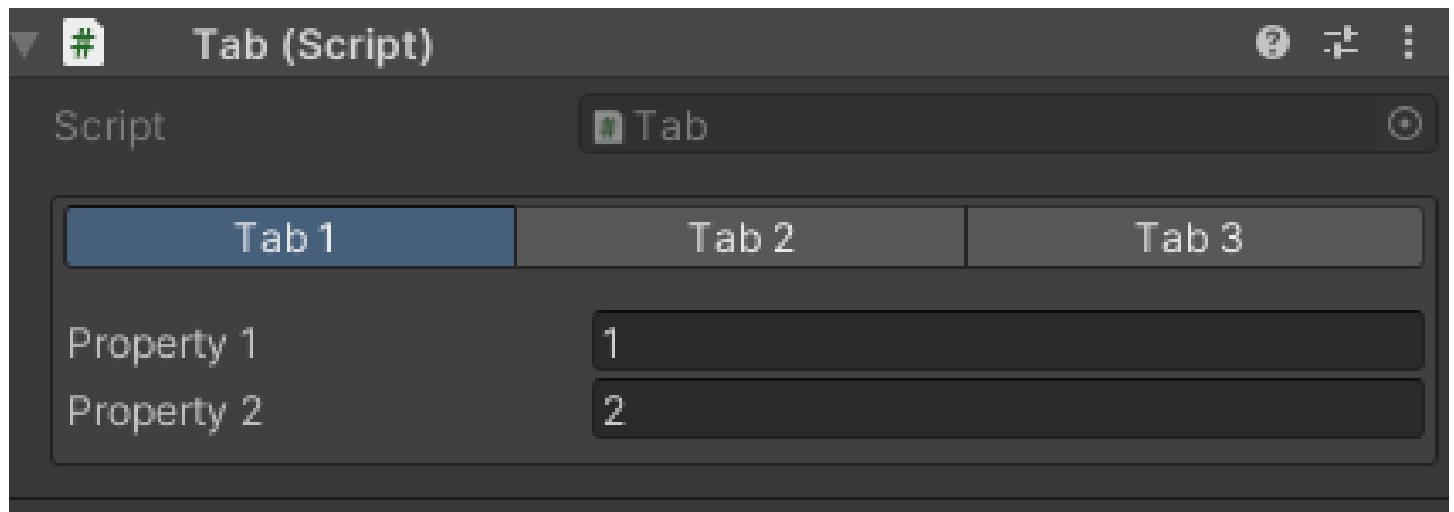
    [Tab("Tab 2")]
    [HideInInspector] public bool property4;

    [Tab("Tab 3", spaceAbove: 10f)]
    [HideInInspector] public float property5;

    [Tab("Tab 3")]
    [HideInInspector] public bool property6;
}
```

The Tab attribute organizes different property fields within separate tabs in the Unity Inspector.

- **tabName:** Specifies the name of the tab.
- **spaceAbove:** Adjusts the space above the box.
- **spaceBelow:** Adjusts the space below the box.



→ **Button Attribute**

```

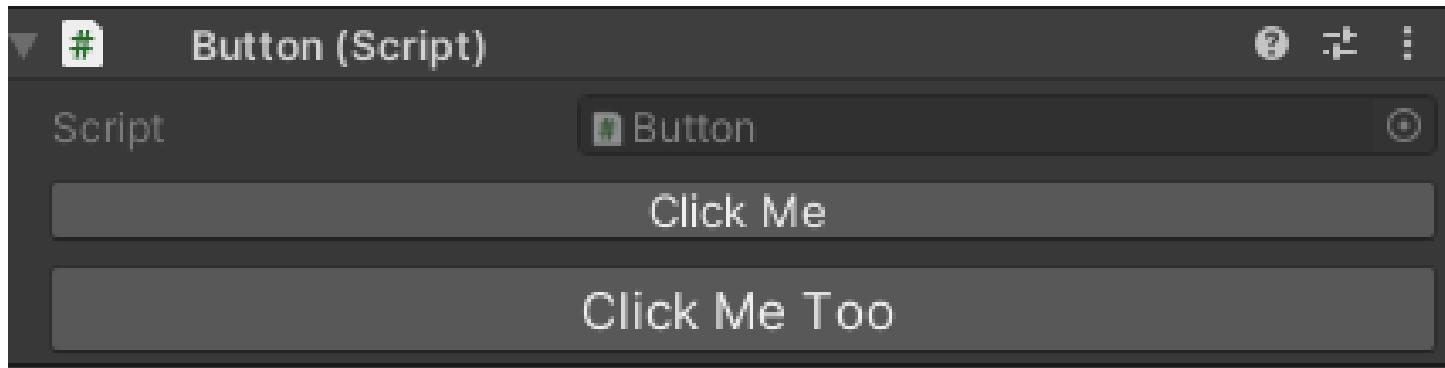
using UnityEngine;
using EnivStudios.EnivInspector;
@ Unity Script (1 asset reference) | 0 references
public class Button : MonoBehaviour
{
    [Button("SomeMethod", spaceAbove:5f, spaceBelow:0f)]
    [SerializeField] private string clickMe;

    [Button("AnotherMethod", labelFontSize:15, buttonSize:26f ,spaceAbove: 5f, spaceBelow: 2f)]
    [SerializeField] private string clickMeToo;
    0 references
    public void SomeMethod()
    {
        Debug.Log("Some Action Performed");
    }
    0 references
    public void AnotherMethod()
    {
        Debug.Log("Another Action Performed");
    }
}

```

The **Button** attribute adds a button to the Unity Inspector that triggers a specified method when clicked.

- **methodName:** Specifies the name of the method to be called when the button is clicked.
- **labelFontSize:** Sets the font size of the button label.
- **buttonSize:** Sets the size of the button.
- **spaceAbove:** Adjusts the space above the button.
- **spaceBelow:** Adjusts the space below the button.



→ **IconTitle Attribute**

```
using UnityEngine;
using EnivStudios.EnivInspector;

⑧ Unity Script (1 asset reference) | 0 references
public class IconTitle : MonoBehaviour
{
    [IconTitle("Assets/StylizeC#ScriptPro/Images/monitor-check.png", spaceAbove: 0, spaceBelow: 5)]
    [SerializeField] private string displaySettings; //Icon Title Name
    [SerializeField] private int someVar_1;
    [SerializeField] private Vector2 screenResolution;

    [IconTitle("Assets/StylizeC#ScriptPro/Images/headphones.png", spaceAbove: 5, spaceBelow: 5)]
    [SerializeField] private string soundSettings; //Icon Title Name
    [SerializeField] private int someVar_2;
    [SerializeField][Range(1,10)] private int soundVolume;

    [IconTitle("Assets/StylizeC#ScriptPro/Images/file-type.png", spaceAbove: 5, spaceBelow: 0)]
    [SerializeField] private string uISettings; //Icon Title Name
    [SerializeField] private Image logo;
    [SerializeField] private int someVar_3;
}
```

The **IconTitle** attribute adds an icon alongside a title in the Unity Inspector.

- **iconPath:** Specifies the path to the icon to be displayed.
- **spaceAbove:** Adjusts the space above the icon and title.

spaceBelow: Adjusts the space below the icon and title.