Unity Help System by Xmax30060

English version

How to start using the help script?

1) Add "using Help;" at the top of the C# script that requires the functionalities.

2) Then use "Help.Help." followed by the name of the function you want to use.

Example:

```
using Help;
using UnityEngine;

public class Example : MonoBehaviour {
    public void Say()
    {
        Help.Help.Debug("hello");
    }
}
```

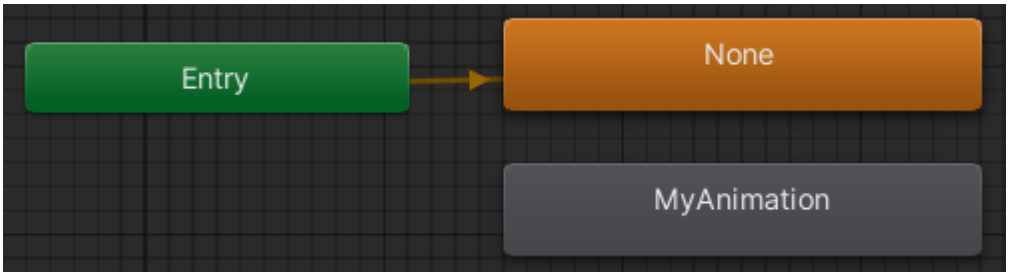
Which functions are available?

1) Debugs function's:

- `Debug(text): Call your message with the style of information.` 
- `Warn(text): Call your message with the style of warning.` 
- `Error(text): Call your message with the style of error.` 

2) Crossfade Animations functions:

- `GetAnimator(Your Game Object)`: This function initializes the animator to enable the use of animations in other animation functions. (To use this function, make sure to assign the Game Object for which you want to retrieve the animator. Additionally, ensure that there is an Animator component attached to this Game Object.)
- `ChangeAnimation(Name of the animation, optional crossfade of 0.2f by default)`: This function gives you the possibility to launch an animation directly in your code, practically without using the Unity Animator system. For example, you could achieve this in Unity Animator:



To launch the animation named "MyAnimation", you just need to write the following code :

```
using Help;
using UnityEngine;

public class Exemple : MonoBehaviour
{
    private void Start()
    {
        Help.Help.GetAnimator(gameObject); //be sure to make this line !!
        float opionalCroosFade = 1.0f;
        Help.Help.ChangeAnimation("MyAnimation", opionalCroosFade);
    }
}
```

The function ChangeAnimation() doesn't work if you don't use the GetAnimator() function!

3) functions for child's games objects

- `FindDeepChild(Transform parent, string name)`: This function allows you to find the Transform of a child of an object using just the Transform of the parent and the name of the child whose Transform you wish to obtain.

Example:

```
Transform parentTransform = gameObject.transform; // Replace with your actual GameObject
string targetChildName = "MyChildObject"; // Replace with the desired child name
Transform foundChild = parentTransform.FindDeepChild(targetChildName);
```

- `LayerChildUI(parentGameObject, newLayerName)`: This function allows you to change the layer of all children GameObjects based on a specified parent GameObject.

Example (Create a GameObject and then apply the "UI" layer to it and all its children):

```
GameObject monObjet = new GameObject("MonObjet");

SetLayerRecursively(monObjet, "UI");
```