This package contains extensions of the MonoBehaviour class for simplifying the use of coroutines. With this package, you can simplify approximately 80% of coroutine usage without the need to create IEnumerator methods.

All the necessary code is located in the Assets/ThirdParty/EasyCoroutines directory. The Assets/ThirdParty/EasyCoroutinesSample directory contains usage examples, and it would be beneficial for you to familiarize yourself with the EasyCoroutinesSample script.

The extension provides the following methods for scripts that inherit from MonoBehaviour:

```
- Coroutine Cor_WaitUnscaledTime(float waitForSeconds, System.Action afterWaiting);
- Coroutine Cor_WaitScaledTime (float waitForSeconds, System.Action afterWaiting);
- Coroutine Cor_SkipFrames(int countFrame, System.Action afterWaiting);?
- Coroutine Cor_UnscaledAction(float duration1, System.Action<float> corAction1,
            float duration2 = 0, System.Action<float> corAction2 = null,
            float duration3 = 0, System.Action<float> corAction3 = null,
            float duration4 = 0, System.Action<float> corAction4 = null
            )
- Coroutine Cor_ScaledAction(float duration1, System.Action<float> corAction1,
            float duration2 = 0, System.Action<float> corAction2 = null,
            float duration3 = 0, System.Action<float> corAction3 = null,
            float duration4 = 0, System.Action<float> corAction4 = null
            )
```

When you use the Easy Coroutines extension's this.Cor_UnscaledAction or this.Cor_ScaledAction, you are guaranteed that the 't' value is clamped between 0 and 1.

Example use of Cor_WaitUnscaledTime/Cor_WaitScaledTime:

```
void Start()
{
   this.Cor_WaitUnscaledTime(1.5f, () => Debug.Log("Print After waiting for 1.5
seconds"));
}
```

Example use of Cor_ UnscaledAction/Cor_ScaledAction:

```
Void Start()
{
    Vector3 A = Vector3.zero;
    Vector3 B = Vector3.one;
    Vector3 C = Vector3.one*3;
    this.Cor_UnscaledAction(
        2, t => transform.position = Vector3.Lerp(A, B, t),
        2, t => transform.position = Vector3.Lerp(B, C, t),
        2, t => transform.position = Vector3.Lerp(C, A, t),
        0, _ => gameObject.SetActive(false)
        );
}
```

To introduce a wait period within the UnscaledAction/ScaledAction chain, simply pass **5, _ => {}** to wait for 5 seconds.

If you need to execute specific code once within the sequence of actions, set the duration parameter to 0, like this: **0, _ => { Debug.Log("Print something"); }**.

For additional examples, please refer to the samples.

Stop Coroutines:

To stop coroutines works similar to stopping regular Coroutines).You may use coroutine objects like this:

```
       Coroutine coroutine = this.Cor_WaitUnscaledTime(1.5f, () => Debug.Log("Executed after waiting for 1.5 seconds"));
      StopCoroutine(coroutine);
```

Using *StopAllCoroutines();* will also affect coroutines created by this extension in a similar way.