

How to Change Over

Memory Efficient Coroutines are called slightly differently than Unity's coroutines. The structure is exactly the same, so in most cases it's just a matter of find and replace inside your code.

First, you need to include the MEC namespace, like this:

```
1. using MEC;
```

Next, replace every instance of `StartCoroutine`, so this

```
1. StartCoroutine(CheckForWin());
```

is replaced with one of these three lines. (You have to pick the execution loop when you define the process.)

```
1. Timing.StartCoroutine(CheckForWin()); // To run in the Update loop.
2. Timing.StartFixedUpdateCoroutine(CheckForWin()); // To run in the
   FixedUpdate loop.
3. Timing.StartLateUpdateCoroutine(CheckForWin()); // To run in the
   LateUpdate loop.
```

The process' header will then need to be changed as well. It turns from this:

```
1. IEnumerator CheckForWin()
2. {
3. ...
4. }
```

To this:

```
1. IEnumerator<float> CheckForWin()
2. {
3. ...
4. }
```

Whenever you want to wait for the next frame, just yield return 0. So this,

```
1. IEnumerator CheckForWin()
2. {
3.     while (_cubesHit < TotalCubes)
4.     {
5.         WinText.text = "Have not won yet.";
6.
7.         yield return null;
8.     }
9.
10.    WinText.text = "You win!";
11. }
```

Would turn into this:

```
1. IEnumerator<float> CheckForWin()
2. {
3.     while (_cubesHit < TotalCubes)
4.     {
5.         WinText.text = "Have not won yet.";
6.
7.         yield return 0f;
8.     }
9.
10.    WinText.text = "You win!";
11. }
```

If you want to pause for some number of seconds rather than just one frame then you can either pass in the time that you want to resume the process (using something like “yield return Time.time + timeToWait;”) or you can use Timing.WaitForSeconds. So this,

```
1. IEnumerator CheckForWin()
2. {
3.     while (_cubesHit < TotalCubes)
4.     {
5.         WinText.text = "Have not won yet.";
6.
7.         yield return new WaitForSeconds(0.1f);
8.     }
9.
10.    WinText.text = "You win!";
11. }
```

Turns into this:

```
1. IEnumerator<float> CheckForWin()
2. {
3.     while (_cubesHit < TotalCubes)
4.     {
5.         WinText.text = "Have not won yet.";
6.
7.         yield return Timing.WaitForSeconds(0.1f);
8.     }
9.
10.    WinText.text = "You win!";
11. }
```

For the latest documentation, FAQ, or to contact the author visit <http://trinary.tech/category/mec/>