

Coroutines

Works like a “thread”

WaitUntil

WaitForSeconds

WaitForEndOfFrame

yield

```
public float totalTimeInSeconds = 5f;

void Start()
{
    StartCoroutine(DoTimer());
}

IEnumerator DoTimer()
{
    float timer = 0;
    bool timerIsDone = false;

    while (timerIsDone != true)
    {
        timer += Time.deltaTime;
        if (timer > totalTimeInSeconds)
        {
            Debug.Log("timer is done!");
            timerIsDone = true; // this will end the loop and the coroutine
        }
        yield return null; // wait until the end of the frame
    }
}
```



InvokeRepeating

- Run a function repeatedly
- CancelInvoke will stop all instances of InvokeRepeating

```
private float startDelayTime = 1f; // wait 1 second, then start
private float intervalInSeconds = 2f; // call the function every 2 seconds

private void Start()
{
    InvokeRepeating("Jump", startDelayTime, intervalInSeconds);
}

public void Jump()
{
    _rb.velocity = new Vector2(_rb.velocity.x, JumpVelocity);
}
```

```
private void OnDisable()
{
    CancelInvoke();
}
```

