


```
[UnityTest]
```

```
public IEnumerator Waaaaaaaai()
```

```
{
```

```
    /*
```

IEnumeratorを返すメソッドでテストを書くことができる。

これによって、特定の条件が整うまで非同期で待ったり、
好きなタイミングでコードに干渉できたりする。

```
    */
```

```
while (true)
```

```
{
```

```
    yield return null;
```

```
}
```

```
}
```


現在のUnityでのテスト

欠点

- 実機を有線接続しないと実機テストができない
- Setup/Tearardownがvoidしか返せないの非同期的なことしかできず、自由度が低い

```
[SetUp]  
public void Setup2()  
{  
    Debug.Log("OK");  
}
```

UnityTestのいいところ

```
[UnityTest]
```

```
public IEnumerator Waaaaaaaai()
```

```
{
```

```
    /*
```

IEnumeratorを返すメソッドでテストを書くことができる。

これによって、特定の条件が整うまで非同期で待ったり、好きなタイミングでコードに干渉できたりする。

```
    */
```

```
    while (true)
```

```
    {
```

```
        yield return null;
```

```
    }
```

```
}
```

このテストメソッドが、毎フレーム、
Updateのあと、LateUpdateの前に呼ばれる。