7.Start:让多线程更简单

多线程，是作为高级开发者必须具备的一种技术。了解了多线程可以让我们充分利用多核移动端的计算优势，也可以让我们的游戏体验更平滑。

在 Unity 中我们一般用 Thread.Start 开启一个线程。当逻辑非常复杂的时候多线程非常难以管理。

而 UniRx 改善了这一种状况。

一个”当所有线程运行完成后，在主线程执行某个任务” 这个功能，使用 UniRx 实现如下:

```
public class ThreadTest : MonoBehaviour
{
    void Start()
    {
        var threadAStream = Observable.Start(() =>
        {
            System.Threading.Thread.Sleep(TimeSpan.FromSeconds(1));
            return 10;
        });

        void threadBStream = Observable.Start() =>
        {
            System.Threading.Thread.Sleep(TimeSpan.FromSeconds(3));
            return 10;
        });

        Observable.WhenAll(threadAStream,threadBStream)
                .ObserveOnMainThread()
                .Subscribe(xs =>
                {
                    Debug.Log(xs[0] + ":" + xs[1]);
                });

    }
```

}

3 秒后，输出的结果如下:

```
10:10
```

这里有两个新的 API，一个是 Observable.Start，这个 API 意思开启一个线程流。

ObserveOnMainThread，意思是把 WhellAll 结果转到主线程上。这样 Subscribe 里的回调就可以使用 Unity 的 API 了（Unity 的很多 API 不可以在其他线程中使用 ）。

使用 UniRx 来处理线程逻辑非常简单。

线程和 Coroutine (协程）都可以使用 WhenAll 这种操作符。

除了 WhenAll 还有很多其他的操作符，我们在之后慢慢学习。

今天的内容就这些。