

# 0.什么是 UniRx

UniRx 是一个 Unity3D 的编程框架。

专注于解决异步逻辑，使得异步逻辑的实现更加简洁优雅。

## 简洁优雅如何体现？

比如，实现一个”只处理第一次鼠标点击事件”这个功能，使用 UniRx 实现如下：

```
Observable.EveryUpdate()  
    .Where(_ => Input.GetMouseButtonUp(0))  
    .First()  
    .Subscribe(_ => { // do something });
```

这行代码做的事情很简单。

1. 开启一个 Update 的事件监听。
2. 每次事 Update 事件被调用时，进行一个鼠标是否抬起的判断。
3. 如果判断通过，则进行计数，并且只获取第一次的点击的事件。
4. 订阅/处理事件。

如果使用传统的方式实现”只处理第一次鼠标点击事件“，这个功能，不知道要写多少行代码，还要创建一个成员变量来记录点击次数，或者是否点击过。还要在脚本中创建一个 Update 方法来监听鼠标抬起事件。

如果在 Update 方法中，除了实现鼠标事件监听这个功能之外，还要实现其他的功能。那么 Update 里就会充斥着大量的状态判断等逻辑。代码非常不容易阅读。

而 UniRx 提供了一种编程思维，使得平时一些比较难实现的异步逻辑（比如当前这种），使用 UniRx 轻松搞定，并且不失代码的可读性。

当然 UniRx 的强大不仅仅如此。

它还可以：

- 优雅实现 MVP（MVC）架构模式。
- 对 UGUI/Unity API 提供了增强，很多需要些大量代码的 UI 逻辑，使用 UniRx 优雅实现。
- 轻松实现非常复杂的异步任务处理。
- 等等。

最最重要的是，它可以提高我们的编码效率。还给我们的大脑提供一个强有力的编程模型。

UniRx 非常值得研究学习，就连大名鼎鼎的 uFrame 框架，在 1.6 版本之后使用 UniRx 做了大幅重构，底层使用 UniRx 强力驱动。

连 凉鞋老师的 QFramework 里也是引入了 UniRx，有一大批的框架用户都是为了 UniRx 慕名而来。

## 为什么要用 UniRx?

UniRx 就是 Unity Reactive Extensions。是 Unity 版本的 Reactive Extensions。

Reactive Extensions 的擅长的部分是处理时间上异步的逻辑。

游戏很多的系统都是在时间上异步的，所以 Unity 开发者要实现的异步（在时间上）任务，是非常多的。

这也是为什么 Unity 官方提供了 Coroutine（协程）这样的概念。

在游戏中，大部分的逻辑都是在时间上异步的。比如动画的播放、声音的播放、网络请求、资源加载/卸载、Tween、场景过渡等都是在时间上异步的逻辑。甚至是游戏循环（Every Update, OnCollisionEnter, etc），传感器数据（Kinect, Leap Motion, VR Input, etc.）都是(时间上)异步的事件。

我们往往在进行以上逻辑实现的时候经常用到大量的回调，最终随着项目的扩张导致传说中的“回调地狱”。

相对较好地方法则是使用消息/事件的发送，结果导致“消息满天飞”，导致代码非常难以阅读。

使用 Coroutine 也是非常不错的，但是 Coroutine 本身的定义，是以一个方法的格式定义的，写起来是非常面向过程的。当逻辑稍微复杂一点，就很容易造成 Coroutine 嵌套 Coroutine，代码是非常不容易阅读的（强耦合）。

而 UniRx 的出现刚好解决了这个问题，它介于回调和事件之间。

它有事件的概念，只不过他的事件是像水一样流过来，而我们要做的则是简单地进行组织、变换、过滤、合并。

它也用到了回调，只不过事件组织之后，只有简单一个回调就可以进行事件的处理了。

它的原理和 Coroutine(迭代器模式) 非常类似，但是比 Coroutine 强大得多。

UniRx 将（时间上）异步事件转化为 (响应式的) 事件序列，通过 LINQ 操作可以很简单地组合起来，还支持时间操作。

为什么要用 UniRx 呢？

总结为一句话就是，游戏本身有大量的（时间上）异步逻辑，而 UniRx 恰好擅长处理（时间上）异步逻辑，使用 UniRx 可以节省我们的时间，同时让代码更简洁易读。

Rx 只是一套标准，在其他语言也有实现，如果在 Unity 中熟悉了这套标准，在其他语言上也是可以很快上手的。比如 RxJava、Rx.cpp、SwiftRx 等等。

## 课程内容:

- 实践并讲解开发中最常用的 UniRx API。
- 对比 Linq 的操作符系统地介绍 UniRx 的操作符。
- 感受 UniRx 的魅力

- 系统地掌握 UniRx 的 API

## 课程目标:

- 掌握 UniRx 的基本写法，能够看懂 UniRx 写的大部分代码，并能够直接将其用到工作开发中。
- 系统地掌握，UniRx 的概念和 API。

今天呢，对 UniRx 进行了一个比较粗浅地介绍。在大家学习完第一章之后，老师会进行比较客观严谨地介绍。

OK，让我们感受下 UniRx 的魅力吧。