

# 1.Linq 与 UniRx 操作符

UniRx 有非常多的操作符，比如（Where，First）等等。

这些操作符的意思与 LINQ 的操作符基本一致。

在讲解 UniRx 操作符之前，要先简单介绍一下 LINQ 的操作符。

因为学习 LINQ 操作符可以加深我们对 UniRx 的理解。

## LINQ 简介

LINQ 是在 C# 3 发布的一个概念。主要是一种查询语法的实现，它可以像 SQL 语句一样查询 C# 的数据列表、XML 文件和数据库。

最简单的就是 List 的 ForEach。

ForEach 是 LINQ 的链式写法。

其功能和我们常用的 foreach 是一样的。

代码如下：

```
var testList = new List<int> { 1, 2, 3};
```

```
testList.ForEach(num => Debug.Log(num));
```

输出结果为：

```
1  
2  
3
```

非常简单。

那么我们之前在 UniRx 的操作符也是通用的。

```
var firstNumber = testList.First(); // 1
```

```
Debug.Log(firstNumber);

testList.Where(num => num > 1)
    .ForEach(number => Debug.Log(number));
```

输出结果为:

```
1
2
3
```

Where 就是过滤的意思。经过第一章，大家应该都了解吧？

## 为什么先介绍 LINQ?

这个答案，在前边已经讲过了。

就是“学习 LINQ 可以加深我们对 UniRx 的理解”。

而且，有的操作符比较晦涩，使用 LINQ 可以排除 UniRx 异步的一个干扰。只要把事件源当做一个 List 就好理解了。

接下来我们简单回顾一下 Rx 的历史。

## Rx 的历史

ReactiveX 是 Reactive Extensions 的缩写，一般简称为 Rx，最初是 LINQ 的一个扩展，由微软的架构师 Erik Meijer 领导的团队开发，在 2012 年 11 月开源，Rx 是一个编程模型，目标是提供一致的编程接口，帮助开发者更方便的处理异步数据流，Rx 库支持 .NET、JavaScript 和 C++，Rx 近几年越来越流行了，现在已经支持几乎全部的流行编程语言了，Rx 的大部分语言库由 ReactiveX 这个组织负责维护，比较流行的有 RxJava/RxJS/Rx.NET，而 Unity 的版本，就是 UniRx。

## 什么是 Rx

微软给的定义是，Rx 是一个函数库，让开发者可以利用可观察序列和 LINQ 风格查询操作符来编写异步和基于事件的程序，使用 Rx，开发者可以用 Observables 表示异步数据流，用 LINQ 操作符查询异步数据流，用 Schedulers 参数化异步数据流的并发处理，Rx 可以这样定义：Rx = Observables + LINQ + Schedulers。

ReactiveX.io(官方) 给的定义是，Rx 是一个使用可观察数据流进行异步编程的编程接口，ReactiveX 结合了观察者模式、迭代器模式和函数式编程的精华。

这里所说的观察者模式、迭代器模式、函数式编程等会在第三章原理部分进行介绍。

那么以上就算是对 Rx 的一个严谨的定义了。

那么接下来，在介绍每个操作符之前呢，都会先通过 LINQ 进行介绍。这样做是因为 LINQ 相比 UniRx 更好理解，UniRx 因为是在时间上大部分是异步的，所以会对同学们造成干扰。

本质上 LINQ 和 UniRx 是非常相似的，这个具体会在后边介绍。

## Rx 的应用

像比较知名的 Microsoft、Netflix、Github、Trello、SoundCloud 都在用 Rx 的各个语言实现版本。

而 Unity 的 uFrame、QFramework 等框架都集成了 UniRx。还有一些框架如 BindingRx、EcsRx 等也是非常优秀的框架，只不过不太出名。但是也非常值得去研究。

今天的内容就这些。