10.ReactiveCollection 与 ReactiveDictionary

ReactiveCollection

ReactiveCollection 类似于 List。

我们可以使用如下的操作符:

```
ObserverAdd // 当 新的 Item 添加则会触发
ObserverRemove // 删除
ObserverReplace // 替换(Update)
ObserverMove // 移动
ObserverCountChanged // 数量有改变(Add、Remove)
```

ReactiveCollection 示例代码

```csharp
/************************************************************************
 * http://sikiedu.com liangxie
 ***********************************************************************/

using UniRx;
using UnityEngine;

namespace UniRxLesson
{
    public class ReactiveCollectionExample : MonoBehaviour
    {
        ReactiveCollection<int> mAges = new ReactiveCollection<int>
        {
            1,2,3,4,5
        };
```

```csharp
    void Start()
    {
        mAges.ObserveAdd()
            .Subscribe(addAge => { Debug.LogFormat("add:{0}",
addAge); });

        mAges.ObserveRemove()
            .Subscribe(removedAge => { Debug.LogFormat("remove:{0}",
removedAge); });

        mAges.ObserveCountChanged()
            .Subscribe(count => { Debug.LogFormat("count:{0}",
count); });

        foreach(var age in mAges)
        {
            Debug.Log(age);
        }


        mAges.Add(6);
        mAges.Remove(2);
    }
}
```

输出结果为

1
2
3
4
5
add:Index:5 Value:6
count:6
remove:Index:1 Value:2
count:5

ReactiveDictionary

ReactiveDictionary 功能与 Dictionary 一样。

同样地，它支持了几个操作符:

```
ObserverAdd // 当 新的 Item 添加则会触发
ObserverRemove // 删除
ObserverReplace // 替换(Update)
ObserverMove // 移动
ObserverCountChanged // 数量有改变(Add、Remove)
```

示例代码如下:

```csharp
/****************************************************************************
 * http://sikiedu.com liangxie
 ****************************************************************************/

using UniRx;
using UnityEngine;

namespace UniRxLesson
{
    public class ReactiveDictionaryExample : MonoBehaviour
    {
        private ReactiveDictionary<string, string> mLanguageCode = new
ReactiveDictionary<string, string>()
        {
            {"en","英语"},
            {"cn","中文"}
        };

        // Use this for initialization
        void Start()
        {
            mLanguageCode.ObserveAdd()
```

```
                    .Subscribe(addedLanguage => { Debug.LogFormat("add:
{0}", addedLanguage.Value); });

        mLanguageCode.ObserveRemove()
                    .Subscribe(removedLanguage =>
{ Debug.LogFormat("remove:{0}", removedLanguage.Value); });

        mLanguageCode.ObserveCountChanged()
            .Subscribe(count => { Debug.LogFormat("count:{0}",
count); });

        mLanguageCode.Add("jp", "日语");
        mLanguageCode.Remove("en");
    }
  }
}
```

输出结果为

```
add:日语
count:3
remove:英语
count:2
```

今天的内容就这些。