ObservableList

Events

Event handlers for add, remove and clear events are available on ObservableList objects.

```
private void OnEnable()
{

    list.AddEvent += OnAddEvent;
    list.ClearEvent += OnClearEvent;
    list.RemoveEvent += OnRemoveEvent;

}
```

When adding a handler to any event, make sure and remove it when the script it is associated with is disabled.

```
private void OnDisable()
{

    list.AddEvent -= OnAddEvent;
    list.ClearEvent -= OnClearEvent;
    list.RemoveEvent -= OnRemoveEvent;

}
```

Count

Gets the number of elements contained in the ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

Debug.Log(list.Count);
```

IsReadOnly

Gets a value indicating whether the ObservableList is read-only.

```
ObservableList<int> list = new ObservableList<int>();

Debug.Log(list.IsReadOnly);
```

Add

Adds an object to the end of the ObservableList.

```
ObservableList<int> list = new ObservableList<int>();

list.Add(1);
```

Clear

Removes all objects from the ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

list.Clear();
```

Contains

Determines whether an element is in the ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

Debug.Log(list.Contains(2));
```

CopyTo

Copies all items in the ObservableList to the array starting at the `arrayIndex`.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

int[] array = new int[10];

list.CopyTo(array, 0);
```

IndexOf

Searches for the specified object and returns the zero-based index of the first occurrence within the entire ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

Debug.Log(list.IndexOf(2));
```

Insert

Inserts an element into the ObservableList at the specified index.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

list.Insert(1, 6);
```

Remove

Removes the first occurrence of a specific object from the ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

list.Remove(1);
```

RemoveAt

Removes the element at the specified index of the ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

list.RemoveAt(0);
```

GetRange

Creates a shallow copy of a range of elements in the source ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

ObservableList<int> newList = list.GetRange(1, 2);
```

AddRange

Adds the elements of the specified collection to the end of the ObservableList.

```
ObservableList<int> list = new ObservableList<int>();

list.AddRange(new List<int> { 1, 2, 3 });
ObservableList<int> list = new ObservableList<int>();

list.AddRange(new ObservableList<int> { 1, 2, 3 });
```

RemoveRange

Removes a range of elements from the ObservableList.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

list.RemoveRange(1, 2);
```

Shuffle

Creates a new copy of an Observablelist and shuffles the values.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

ObservableList<int> shuffledList = numberRange.Shuffle();
```

Slice

Returns a shallow copy of a portion of an Observablelist.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

ObservableList<int> slicedItemsList = numberRange.Slice(2);
```

Splice

Removes and returns a shallow copy of a portion of an Observablelist.

```
ObservableList<int> list = new ObservableList<int> { 1, 2, 3, 4, 5 };

ObservableList<int> removedItemsList = numberRange.Splice(1, 2);
```