Super ScrollView for UGUI

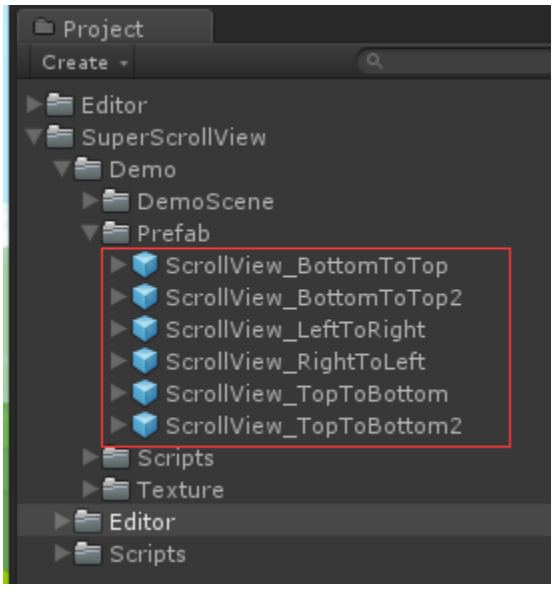
Overview

LoopListView is a component attaching to the same gameobject of UGUI ScrollRect. It helps the UGUI ScrollRect to support any count items with high performance and memory-saving.

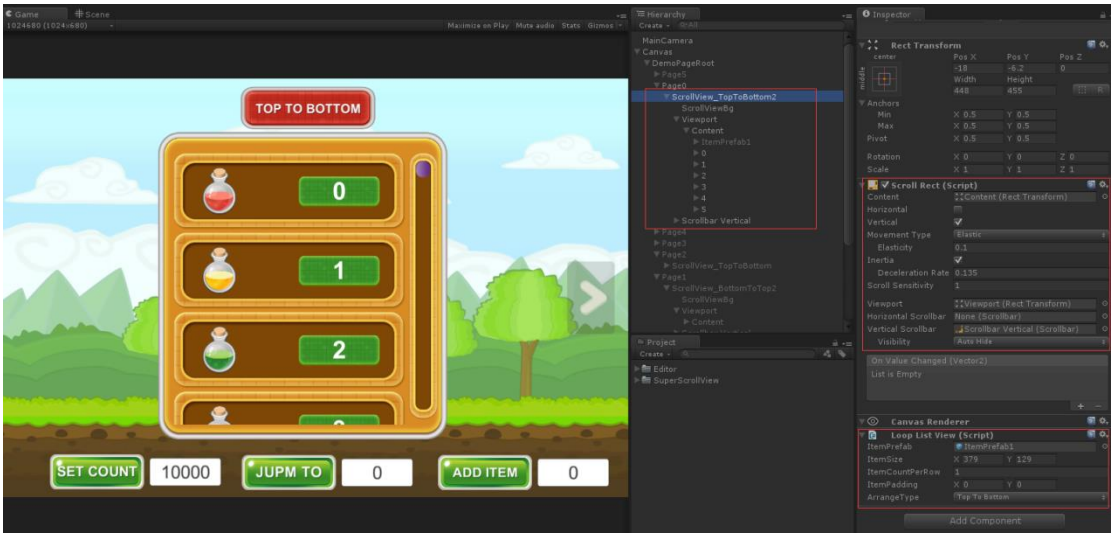
For a ScrollRect with 10,000 items, LoopListView does not really create 10,000 items, but create a few items based on the size of the viewport.

When the ScrollRect moving up, for example, the LoopListView component would check the topmost item's position, and once the topmost item is out of the viewport, then the LoopListView component would reposition the topmost item to be the downmost and calls the **onItemIndexUpdated** handler to update the item content.

There are six example prefabs to help you learn the LoopListView component, in the folder with path : Assets -> SuperScrollView -> Demo -> Prefab.



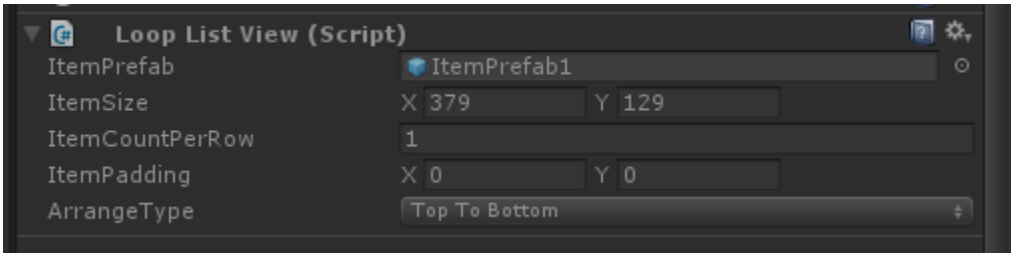
The following picture is what a TopToButtom arranged scrollrect looks like:



In the above picture,the scrollrect have 10000 items,but in fact,only 6 items really created.

Inspector Settings

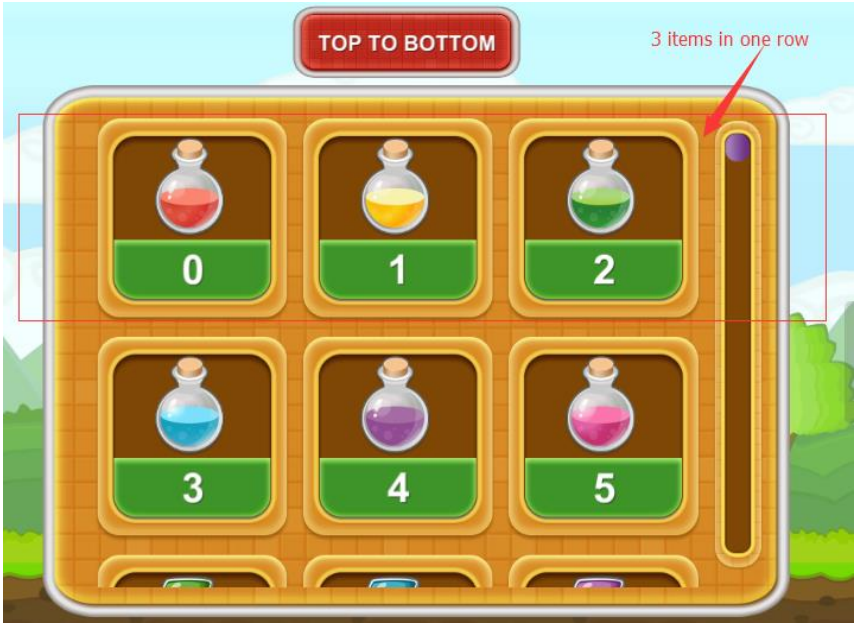
In the Inspector, to make sure the LoopListView component works well, there are several parameters need to set:



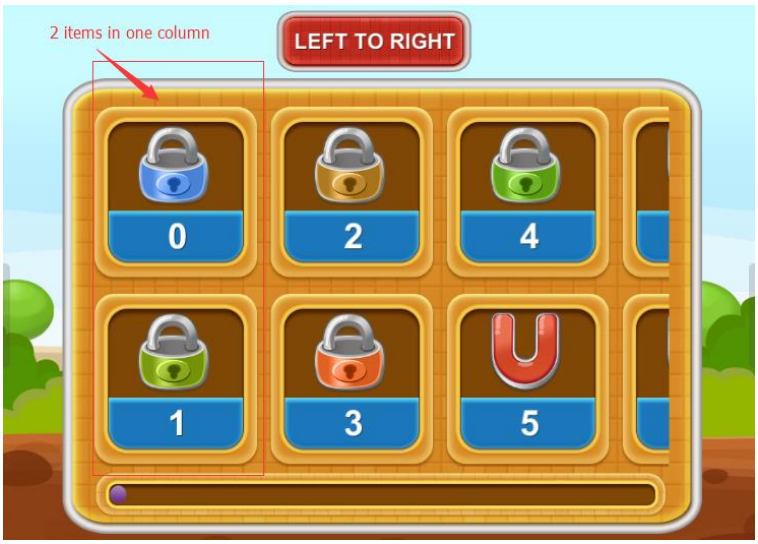
ItemPrefab: this is the existing item to clone.

ItemSize: the width and height of **itemPrefab**.

ItemCountPerRow: when the scrollrect is **vertical**, this parameter would show, and means the item count in one **row**, and the scrollrect would looks like a grid. when this parameter is set to 3, for example, which means there are 3 items in one row, the scrollrect looks like the following:



ItemCountPerColumn: when the scrollrect is **horizontal**, this parameter would show, and means the item count in one **column**, and the scrollrect would looks like a grid. when this parameter is set to 2, for example, which means there are 2 items in one column , the scrollrect looks like the following:



ItemPadding: the amount of spacing between each item in the scrollrect.

ArrangeType: the scroll direction, there are four types:

(1) **TopToBottom**: this type is used for a vertical scrollrect, and the item0,item1,…itemN are positioned one by one **from top to bottom** in the scrollrect viewport, just like the following:

(2) **BottomToTop**: this type is used for a vertical scrollrect, and the item0,item1,…itemN are positioned one by one **from bottom to top** in the scrollrect viewport, just like the following:

(3) **LeftToRight**: this type is used for a horizontal scrollrect, and the item0,item1,…itemN are positioned one by one **from left to right** in the scrollrect viewport.

(4) **RightToLeft:** this type is used for a horizontal scrollrect, and the item0,item1,…itemN are positioned one by one **from right to left** in the scrollrect viewport.

Public Method

```
public void InitListView(int itemTotalCount, System.Action<LoopListViewItem> onItemCreated, System.Action<LoopListViewItem> onItemIndexUpdated)
```

InitListView method is to initiate the LoopListView component. There are 3 parameters:

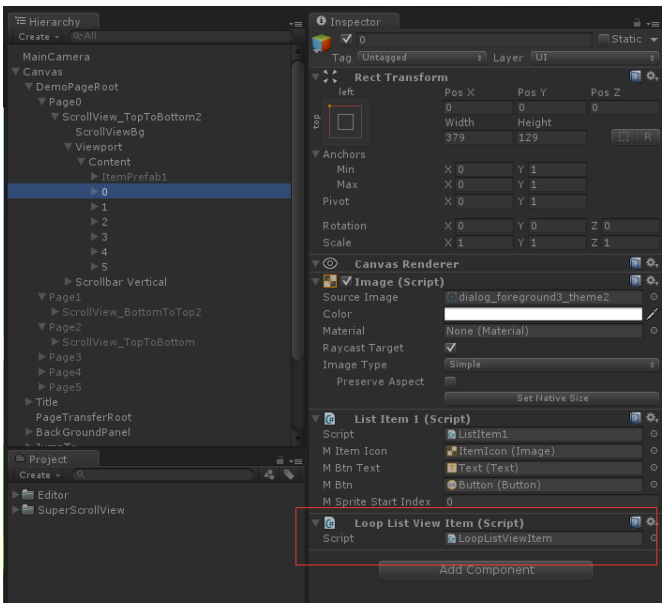
itemTotalCount: the total item count in the listview.

onItemCreated: when a new item is created, this Action will be called.

onItemIndexUpdated: when a item's index value is changed, this Action will be called.

LoopListViewItem is the parameter of both **onItemCreated** and **onItemIndexUpdated**

Every created item has a LoopListViewItem component attached:



LoopListViewItem component is very sample:

```
public class LoopListViewItem : MonoBehaviour
{
    int mItemIndex = -1;
    int mItemId = -1;
    LoopListView mParentListView = null;
```

The mItemIndex property indicates the item's index in the list, and the value starts at 0. When the scrollrect is scrolling, an item's index will change when the item is out of the viewport, and then LoopListView component would reposition the item and calls the **onItemIndexUpdated** handler to update the item content.

The mItemId property indicates the item's id. This property is set when the item created, and will no longer change in life.

```
public void SetListItemCount(int itemCount, bool resetPos = true)
```

This method may use to set the item total count of the listview at runtime.

If resetPos is set false, then the scrollrect's content position will not changed after this method finished.

```
public void ReflushListViewContent()
```

This method will reflush the listview.

```
public void MovePanelByScrollNormalizedValue(float normalizedVal)
```

This method will move the scrollrect content's position to the indicated value.

normalizedVal is from 0 to 1.

For a TopToBottom arranged scrollrect,normalizedVal 0 means the downmost point of the list can reach, and normalizedVal 1 means the topmost point of the list can reach.

For a BottomToTop arranged scrollrect, normalizedVal 0 means the topmost point of the list can reach, and normalizedVal 1 means the downmost point of the list can reach.

```
public void MovePanelToItemIndex(int itemIndex, float offset)
```

itemIndex starts at 0.

This method will move the scrollrect content's position to (the positon of itemIndex-th **item** + offset)

```
public void MovePanelToGroupIndex(int groupIndex, float offset)
```

groupIndex starts at 0.

when the scrollrect is horizontal, This method will move the scrollrect content's localposition.x to (the position of groupIndex-th **column** + offset)

when the scrollrect is vertical, This method will move the scrollrect content's localposition.y to (the position of groupIndex-th **row** + offset)

```
public void MovePanelByScrollValue(float val)
```

Val is indicating how much distance the scrollrect scrolled.

when the scrollrect is horizontal, val is from 0 to Mathf.Abs(Content.size.x – ViewPort.size.x), this method will move the scrollrect content's localposition.x to the indicated value.

when the scrollrect is vertical, val is from 0 to Mathf.Abs(Content.size.y – ViewPort.size.y), this method will move the scrollrect content's localposition.y to the indicated value.

```
public void MovePanelToPosition(float x, float y)
```

when the scrollrect is horizontal, this method will move the scrollrect content's localposition.x to the indicated x.
when the scrollrect is vertical, this method will move the scrollrect content's localposition.y to the indicated y.

```
public void ReflushAllCreatedGroup()
```

This method will reflush all the created items.