ListScrollRect Documentation

Overview

The ListScrollRect script extends the functionality of Unity's ScrollRect script to provide a component that handles vertical and horizontal scrolling lists. It takes care of placing each list item in the proper positions, taking into account list item sizes and any spacing / padding. ListScrollRect also recycles objects by placing objects that scroll out of view into a pool and reusing those objects when new items scroll into view.

ListScrollRect

The ListScrollRect component setup is exactly the same as setting up Unity's ScrollRect component. When the application runs, ListScrollRect will control certain RectTransform properties on the ScrollRect Content as well as populating the Content with list items.

Inspector Properties

Below are the inspector variables that can be set on the ListScrollRect component. The rest of the inspector properties are the same as ScrollRect.

Property	Description
Content Filler Interface	The GameObject that has a component that implements the IContentFiller interface.
Initialize On Start	If this is true, ListScrollRect will call its Initialize method on Start. If this is false then Initialize will need to be called before anything will be displayed in the list.
Scroll Direction	The scroll direction, either Vertical or Horizontal.
Padding	The padding that goes the top/bottom if the scroll direction is vertical and left/right if the scroll direction is horizontal.
Spacing	The amount of spacing between each item in the list.

Public Methods

Method Name	Description
Initialize	Initializes the ListScrollRect.
RebuildContent	Destroys all list items and rebuilds the list from the beginning. Call this if the prefab used for the list items change or the orientation of the list has changed.
RefreshContent	Sends all list items back to the pool and calls GetListItem for each one again. Call this if the content of a list item that is currently on screen has changed, the number of items in the list has changed, or the padding/spacing has changed.
GotoListItem	Shifts the content so the the list item with the specified index appears at the top of the list.

ScrollToListItem	Scrolls the list to the list item with the specified index. A scrollDelta can be specified which controls the speed the list will scroll at (the higher the faster.

IContentFiller

The IContentFiller interface is what ListScrollRect uses to populate the list with items. To use it, create a script that extends Monobehaviour and have that script implement the IContentFiller interface. Attach the script that implements the IContentFiller to a GameObject in the scene then drag that GameObject into the Content Filler Interface property in the ListScrollRect inspector. There are three methods in IContentFiller that must be implemented:

- **GetListItem** - This method is call when ListScrollRect determines that a new list item just scrolled into view. This method expects a non null object to be return. It It takes three arguments:
    - index : The index of the list item in the list.
    - itemType : The type of the item.
    - obj : This will either be null or a pool list item.
- **GetItemCount** - This method is called to get the number of items in the list.
- **GetItemType** - This method is called to get the type of an item at a certain index. ListScrollRect uses the item type to know what pooled item it can pass into GetListItem.

In general, IContentFiller should be implement in the following manner:

```
// The prefab for a list item, assigned in the inspector
public GameObject listItemPrefab;

// Some class that holds all the info to be displayed in a list item
private class ContentInfo
{
    ...
}

// An array or list of data that contains the information for each list item
private ContentInfo[] contentInfoArray;

// Called by ListScrollRect when a list item at “index” scrolls into view
public GameObject GetListItem(int index, int itemType, GameObject obj)
{
    if (obj == null)
    {
        // Instantiate a new instance of the list item prefab
        obj = Instantiate(listItemPrefab);
    }
```

```
        // Get the content info for the list item at the given index
        ContentInfo contentInfo = contentInfoArray[index];

        /* Setup obj's components using contentInfo */

        return obj;
    }

    // Called by ListScrollRect to get the number of list items in the list
    public int GetItemCount()
    {
        return contentInfoArray.Length;
    }
```

Best Practices: Content and List Item RectTransform

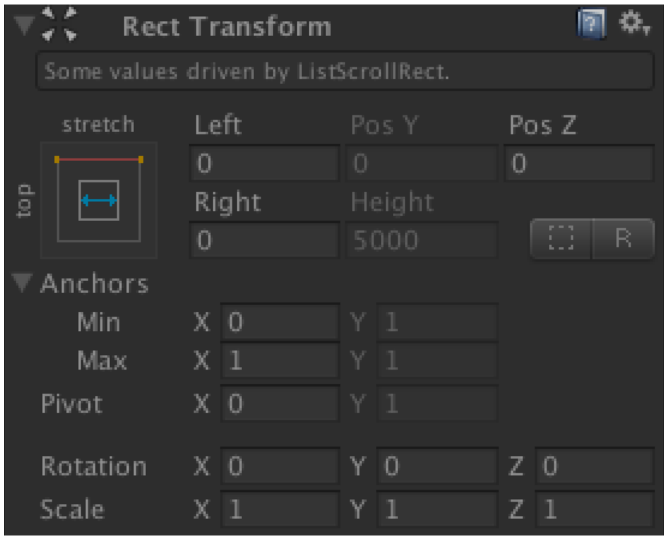
ListScrollRect will only control certain properties in RectTransforms when placing items in the list. If the list is vertical then its up to the user to specify the width of content and the widths of the list items and if the list is horizontal then the heights must be specified.

The best way to set up the Content and list items if the list is **vertical** are as follows:

Content:

- Set the anchors to top – stretch.
- Set the position so Left is 0 and Right is 0.

This will setup the content to stretch its width to fit the viewports width.



Rect Transform

Some values driven by ListScrollRect.

stretch / top

Left: 0 — Pos Y: 0 — Pos Z: 0

Right: 0 — Height: 5000

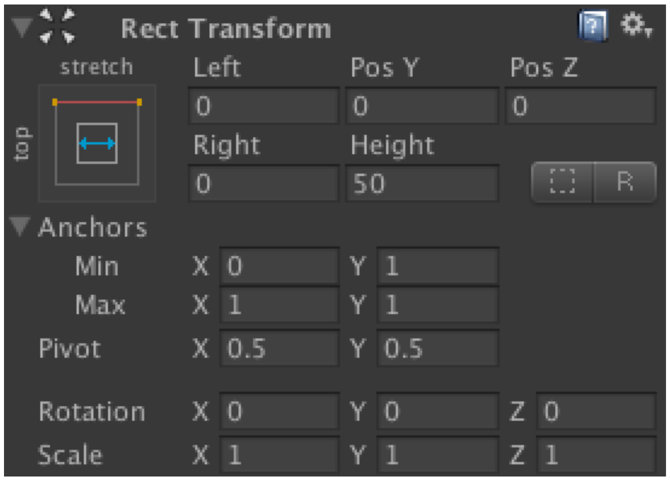
Anchors
Min X 0 Y 1
Max X 1 Y 1
Pivot X 0 Y 1

Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

List Item:

- Set the anchors to top – stretch.
- Set the position so Left is 0 and Right is 0.
- Set the Height to something (This can also be controlled by a ContentSizeFitter / LayoutGroups)

This will setup the list items to stretch their width to fit the contents width.

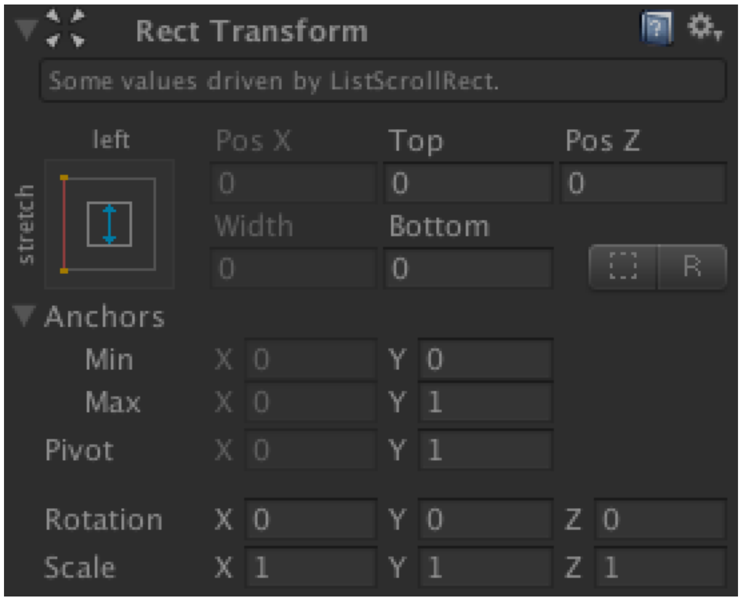

```
Rect Transform
stretch     Left        Pos Y       Pos Z
top         0           0           0
            Right       Height
            0           50
Anchors
   Min      X 0         Y 1
   Max      X 1         Y 1
Pivot       X 0.5       Y 0.5
Rotation    X 0         Y 0         Z 0
Scale       X 1         Y 1         Z 1
```

The best way to set up the Content and list items if the list is **horizontal** are as follows:

Content:

- Set the anchors to stretch – left.
- Set the position so Top is 0 and Bottom is 0.

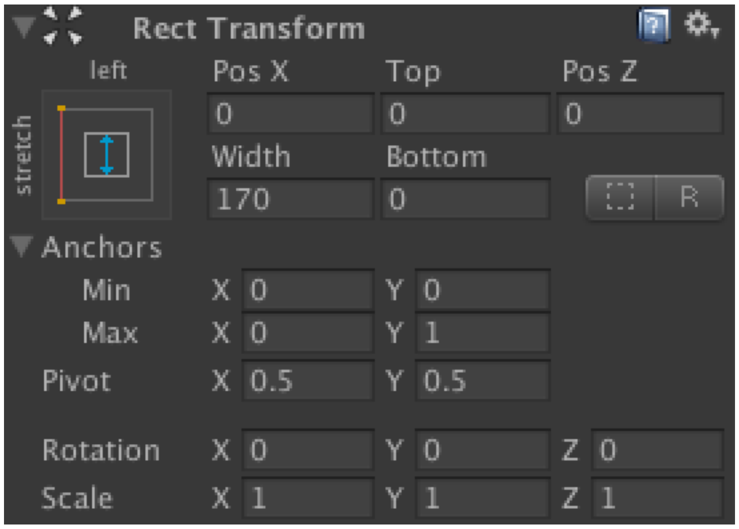
This will setup the content to stretch its height to fit the viewports height.


```
Rect Transform
Some values driven by ListScrollRect.
left        Pos X       Top         Pos Z
stretch     0           0           0
            Width       Bottom
            0           0
Anchors
   Min      X 0         Y 0
   Max      X 0         Y 1
Pivot       X 0         Y 1
Rotation    X 0         Y 0         Z 0
Scale       X 1         Y 1         Z 1
```

List Item:

- Set the anchors to stretch - left
- Set the position so Top is 0 and Bottom is 0
- Set the Width to something (This can also be controlled by a ContentSizeFitter / LayoutGroups)

This will setup the list items to stretch their height to fit the contents height.



Rect Transform

left / stretch

	Pos X	Top	Pos Z
	0	0	0
	Width	Bottom	
	170	0	

Anchors

Min	X	0	Y	0		
Max	X	0	Y	1		
Pivot	X	0.5	Y	0.5		
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1