

UI Toolkit Scroll View Pro - Manual

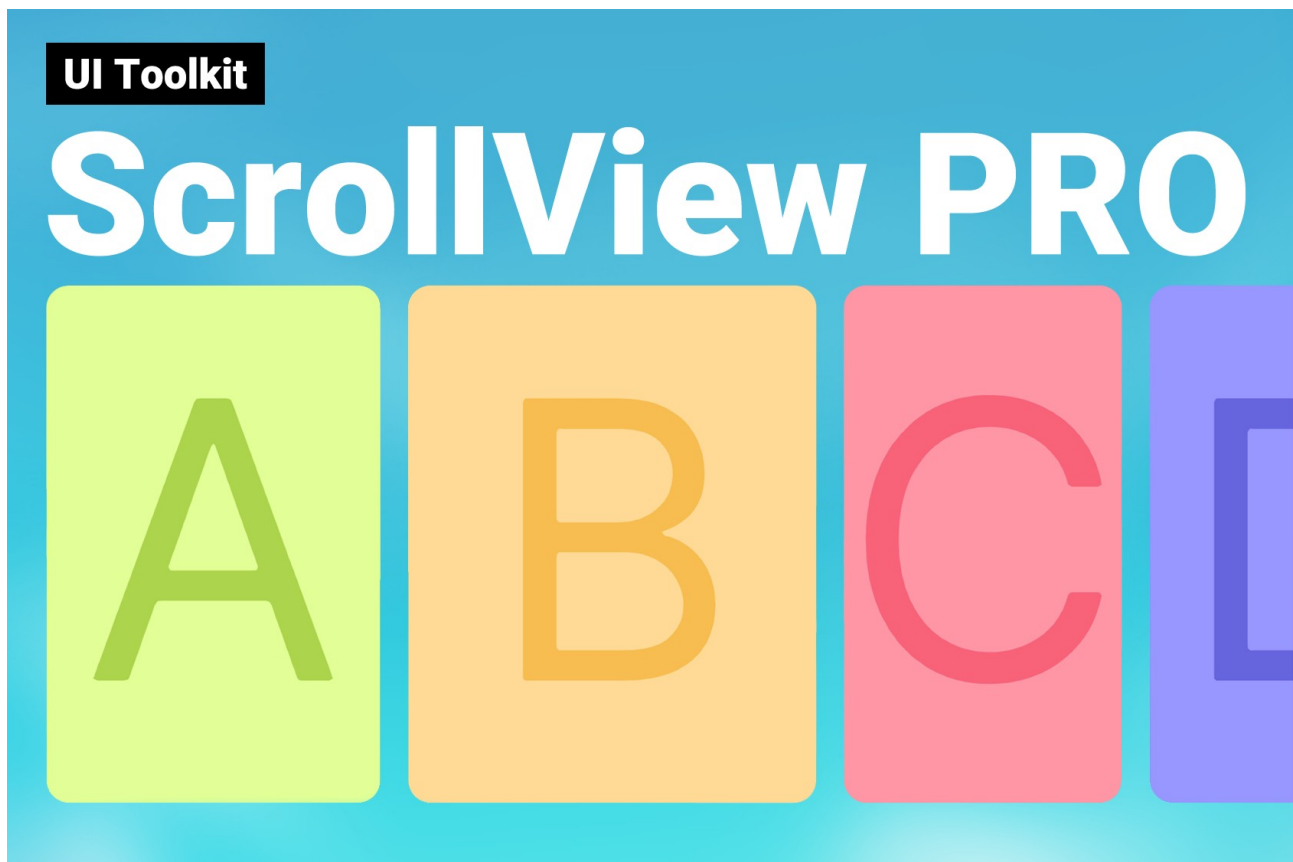


Table of contents

Requirements.....	2
Adding the ScrollView.....	3
Default Attributes.....	3
Scroll View Pro Attributes.....	4
Scroller Buttons.....	4
Focusable Scroll Bars.....	5
Mouse Wheel Scroll Size.....	5
Horizontal/Vertical Page Size.....	5
Touch Scroll Type.....	5
Scroll Deceleration Rate.....	5
Elasticity.....	5
Animation Fps.....	5
Snapping.....	6
Snap Target Classes.....	6
Snap Target Focusables.....	6
Snap Duration.....	6
Snap Align X / Y.....	7
Snap Ease.....	7
Snap Include Margin.....	7

Snap Velocity Threshold.....	8
Focus Snapping.....	9
Focus Snap On Pointer.....	9
Focus Snap Include Margin.....	9
Focus Snap Duration.....	10
Focus Snap Align X / Y.....	10
Focus Snap Ease.....	10
Focus Snap Inside.....	10
Focus Snap Repetition Delay.....	11
Infinite Scrolling.....	12
Infinite Scrolling Caveats.....	12
Dragging.....	13
How to make elements NOT draggable.....	13
Events.....	14
SnapEvent.....	14
Known Issues.....	15
Releasing the drag handle in a scroller does not trigger a Snap on MouseUp.....	15
Frequently Asked Questions.....	16
My scroll view does not snap to the children in my grid.....	16
My buttons are not snapped to if focused. I can not change the focused element with the arrow keys or the controller.....	16
Why should ,selectableScrollbars' and ,focusSnap' not be combined?.....	16
I have added some dynamic children to the scroll view but they are not draggable.....	16
Can I mix regular ScrollView and ScrollViewPro?.....	17

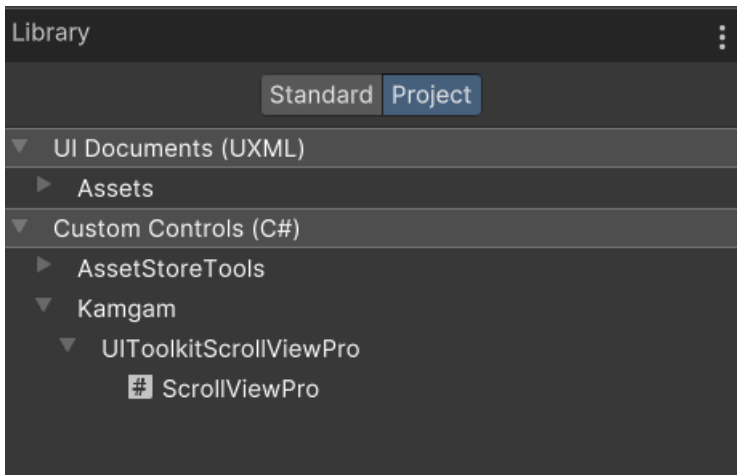
Requirements

Unity 2021.2 or higher is required. Though it may work in earlier versions too (they have not been tested).

Keep in mind, UI Toolkit as a whole is still a work in progress and not quite ready for prime time. Unity itself still recommends using UGUI instead of UI Toolkit for runtime applications ([source](#)).

Adding the ScrollView

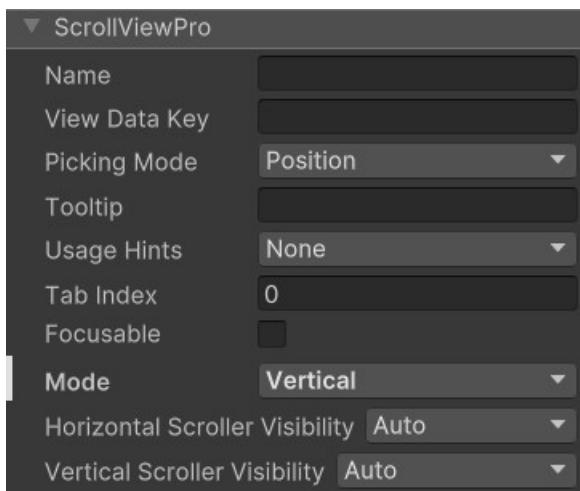
You can find the ScrollViewPro control in the **Library** of your **UI Builder** window.



Simply drag and drop it into your Hierarchy and you are good to go. It works just like the normal ScrollView control.

Default Attributes

The ScrollView tries to be as compatible as possible with the default ScrollView to make the transition easier. There are some default attributes which work just like in the default scroll view.



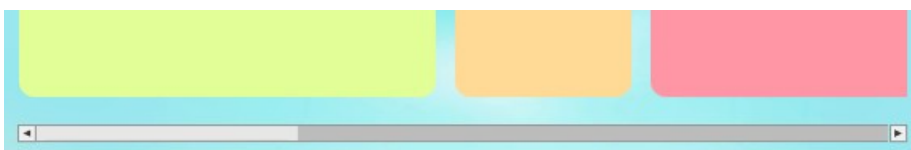
Scroll View Pro Attributes

In addition to the default attributes the Scroll View Pro adds a lot of new stuff for you to configure.

Scroller Buttons	<input checked="" type="checkbox"/>
Focusable Scrollbars	<input type="checkbox"/>
Mouse Wheel Scroll Size	18
Horizontal Page Size	-1
Vertical Page Size	-1
Touch Scroll Type	Clamped
Scroll Deceleration Rate	0.135
Elasticity	0.1
Animation Fps	-1
Snap	<input type="checkbox"/>
Snap Target Classes	
Snap Target Focusables	<input type="checkbox"/>
Snap Duration Sec	0.4
Snap Align X	Center
Snap Align Y	Center
Snap Ease	Ease
Snap Include Margin	<input type="checkbox"/>
Snap Velocity Threshold	1.25
Focus Snap	<input checked="" type="checkbox"/>
Focus Snap On Pointer	<input checked="" type="checkbox"/>
Focus Snap Include Margin	<input checked="" type="checkbox"/>
Focus Snap Duration Sec	0.4
Focus Snap Align X	Visible
Focus Snap Align Y	Center
Focus Snap Ease	Ease
Focus Snap Inside	<input type="checkbox"/>
Focus Snap Repetition Delay	0
Infinite	<input type="checkbox"/>

Scroller Buttons

The scroller buttons defines whether or not the arrow buttons of the scroll view should be show. By default they are enabled, like this:



If you disable them then only the middle part of the scrollbar will remain, like this:



You may want to style these since the borders are usually on the buttons, not the middle.

Focusable Scroll Bars

If you enable this then the scroll bars will be reachable via default focus navigation (usually they are only accessible via pointer click or tab).

Mouse Wheel Scroll Size

It basically work like a scroll speed value for when a mouse wheel is used. 18 is the default and should be find if you panels resolution is at the default of around 1000 units.

Horizontal/Vertical Page Size

This define how much the scroll view will scroll if the [PageUp] or [PageDown] keys are used. It also influences how fast the scroll view will scrill if the arrow buttons are used.

Touch Scroll Type

This works just like the normal touch scroll type only that this also support touch scrolling on Pc via a mouse and it also supports elasticity.

Scroll Deceleration Rate

This works just like in the normal scroll view. It defines how fast the scroll view decelerates.

Elasticity

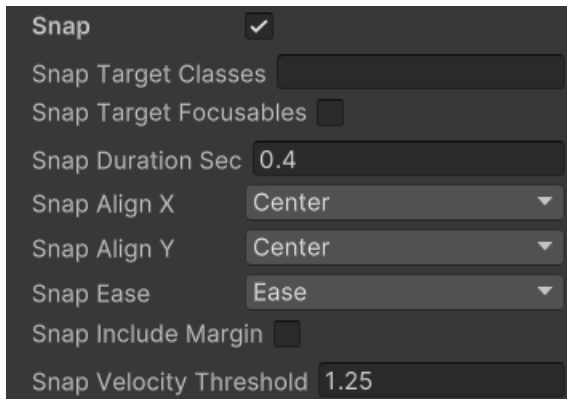
Elasticity tells you how much the scroll you can overshoot when dragged. Once released it will snap back.

Animation Fps

In the default scroll view the animation fps is limited to about 30 (30 milliseconds). In ScrollViewPro you can set a custom frame rate. If left to -1 then the current target frame rate of the application is used.

Snapping

You can enable snapping by ticking the „snap“ checkbox.



The screenshot shows a configuration panel for snapping. It includes a 'Snap' checkbox which is checked. Below it are 'Snap Target Classes' (an empty text field), 'Snap Target Focusables' (an unchecked checkbox), 'Snap Duration Sec' (a text field with '0.4'), 'Snap Align X' (a dropdown menu with 'Center' selected), 'Snap Align Y' (a dropdown menu with 'Center' selected), 'Snap Ease' (a dropdown menu with 'Ease' selected), 'Snap Include Margin' (an unchecked checkbox), and 'Snap Velocity Threshold' (a text field with '1.25').


Snapping means the elements in your scroll view will be attracted to a certain position and they will move the scroll content accordingly.

By default the scroll view will snap to the closest element of the center of your scroll view. However, you can change that via the „Snap Align“ attributes.

Snap Target Classes

By default the snapping will only use the DIRECT children on the content container for snapping.

If you have your children in a sub-container (like a grid) then you will have to tell the scroll view how to find these snap targets. You can do that by setting the „snap target classes“. Then it will use a query to find all elements matching any of the given class names. Class names can be comma separated. Their combination logic is OR, not AND.



This screenshot is a close-up of the 'Snap Target Classes' field in the configuration panel. The 'Snap' checkbox is checked. The 'Snap Target Classes' text field contains the value 'unity-label,unity-button'. The 'Snap Target Focusables' checkbox is unchecked.

Snap Target Focusables

If enabled then the list of snap target is filtered and only those that are focusable will actually be snapped to.

Snap Duration

Defines how long the snapping animation is (in seconds).

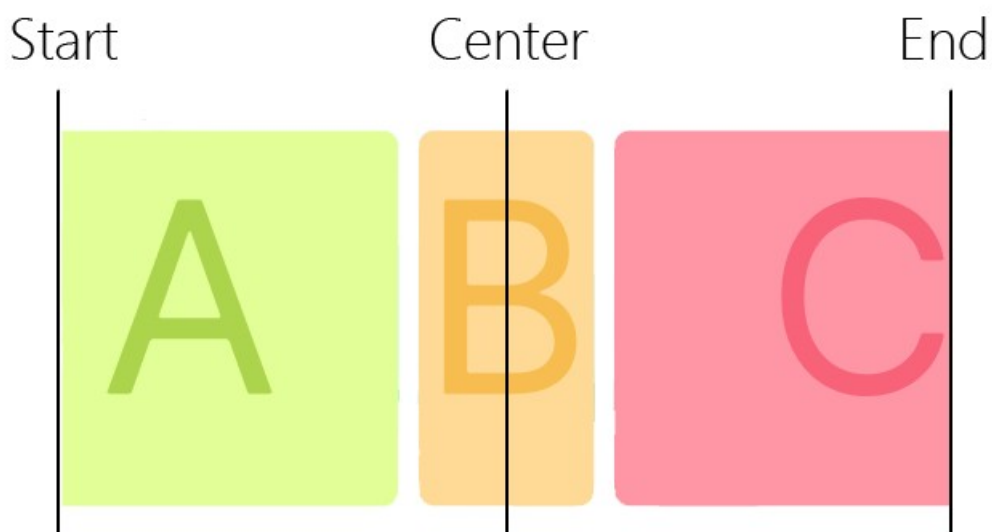
Snap Align X / Y

Snap Align X	Center
Snap Align Y	Center

These will define how the snap target will be determined and how it will be aligned. The alignment works just like the scroll-to alignment settings.

By default the focus of the snap is the center of your scroll view. If X is set to „Start“ then the focus will be the left edge of your scroll view. If X is set to „End“ then it will be the right edge. Y works similar in the vertical direction.

This can have four values: **Start**, **Center**, **End** and **Visible**



Visible is a special case as it is dynamic. This means it will animate the element into view so it becomes fully visible but no more. It will do nothing if the element is already fully visible. It's a bit tricky to getting used to but it can be handy in certain edge cases.

Snap Ease

A list of easing functions defining how the snapping will animate. Pick one the make your snapping a little more exciting. Hint: „BackOut“ easing give a very nice effect.

Snap Include Margin

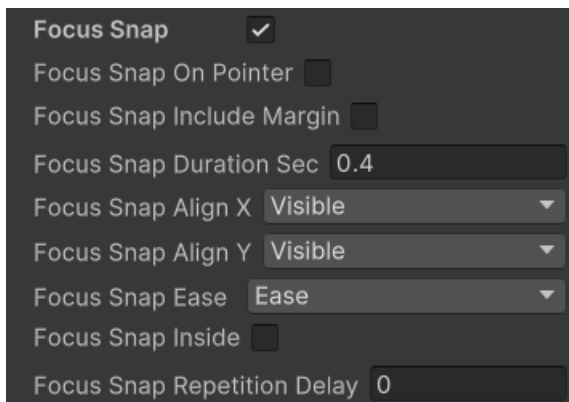
In UI Toolkit elements can have a margin (left, top, right, bottom). If you enable this then these margins are taken into account when snapping. If disabled then the margins are ignored.

Snap Velocity Threshold

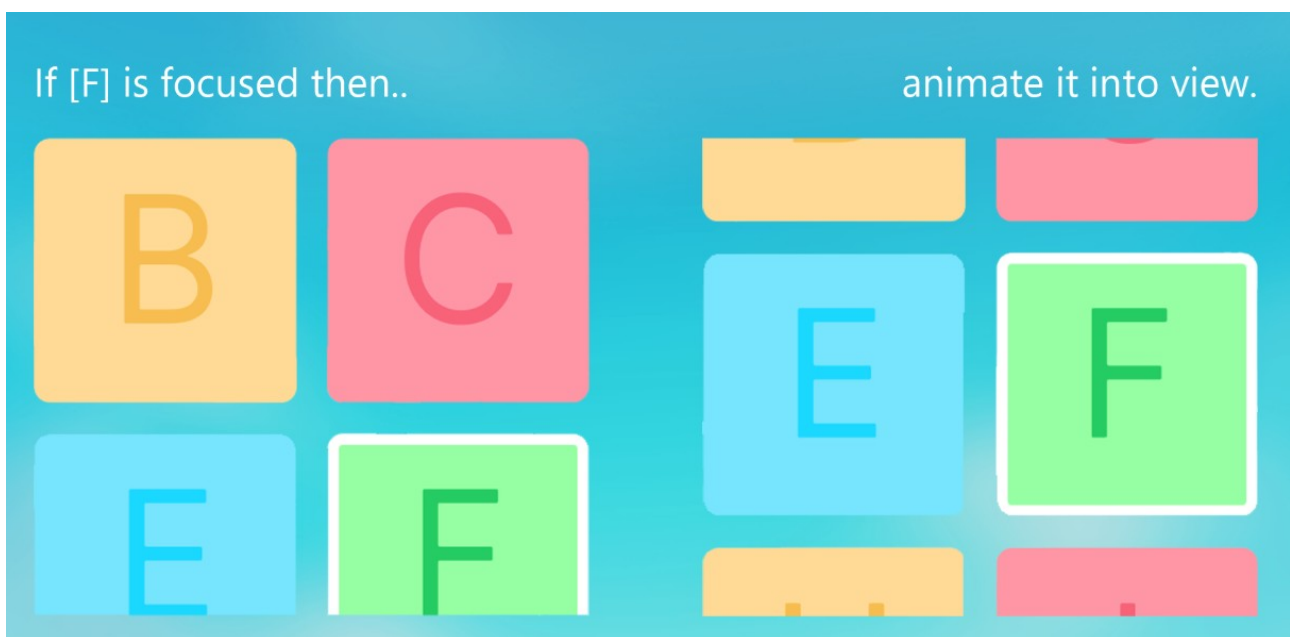
When you flick the scroll view into any direction and you have snapping enable then it will not snap immediately. Instead it will wait for the scroll view to scroll down and then it will start snapping. This defines that threshold.

Focus Snapping

Focus snapping exists to make moving through a scroll view with a controller or a keyboard easier.



If an element outside the viewport is focused then it will be animated into view.



Focus Snap On Pointer

If enabled then the focus snapping will be triggered by pointers (mouse) too. It's disabled by default.

Focus Snap Include Margin

In UI Toolkit elements can have a margin (left, top, right, bottom). If you enable this then these margins are taken into account when snapping. If disabled then the margins are ignored.

Focus Snap Duration

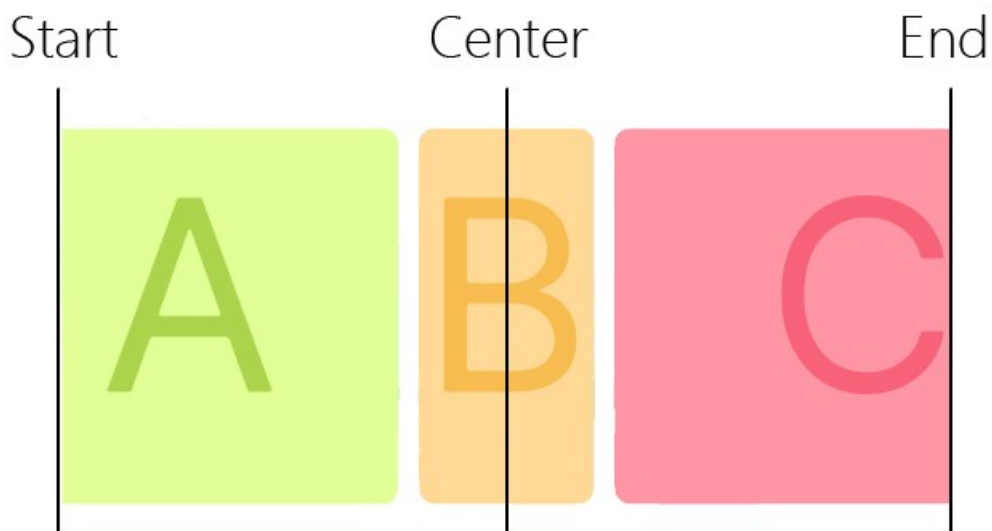
Defines how long the snapping animation is (in seconds).

Focus Snap Align X / Y

These will define how the snap target will be determined and how it will be aligned. The alignment works just like the scroll-to alignment settings.

By default the focus of the snap is the center of your scroll view. If X is set to „Start“ then the focus will be the left edge of your scroll view. If X is set to „End“ then it will be the right edge. Y works similar in the vertical direction.

This can have four values: **Start**, **Center**, **End** and **Visible**



Visible is a special case as it is dynamic. This means it will animate the element into view so it becomes fully visible but no more. It will do nothing if the element is already fully visible. It's a bit tricky to getting used to but it can be handy in certain edge cases.

Focus Snap Ease

A list of easing functions defining how the snapping will animate. Pick one to make your snapping a little more exciting.

Focus Snap Inside

Should the snap scroll animation (alignment) be applied even if the element is already fully visible? Enabling this is handy in combination with `ScrollViewPro.ScrollToAlignment.Center`. Together they make sure the selected element is in the center of your scroll view at all times.

Focus Snap Repetition Delay

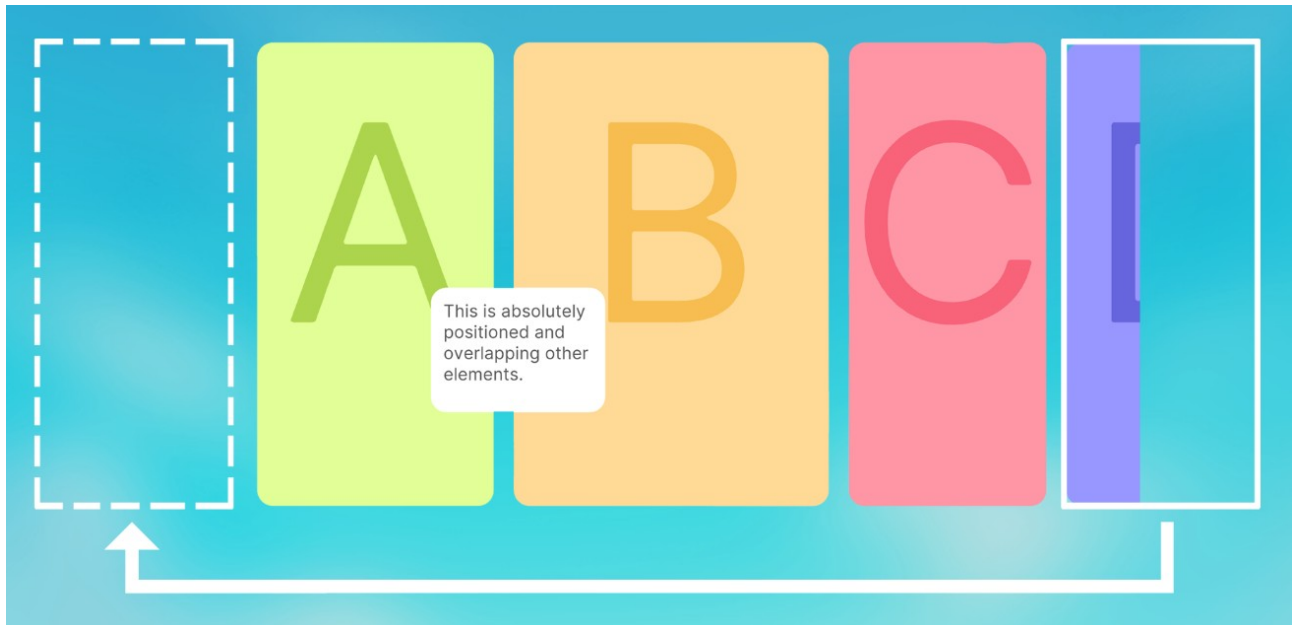
Usually you can leave this at 0. For infinite scroll views however you may want to set this to 0.1 or 0.2 seconds (more on that below in the Infinite Scrolling section)

Infinite Scrolling

Enabling infinite scrolling will make elements in your scroll view repeat.



Whenever an element moves out on one side it is teleported to the other side.



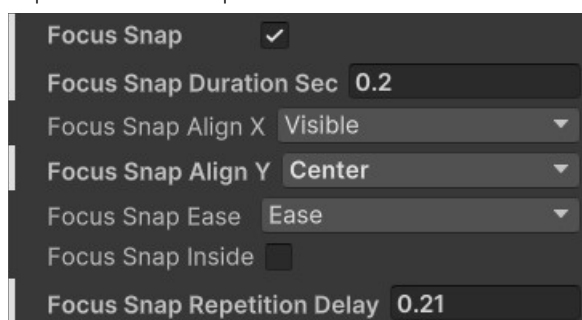
Be careful when enabling infinity and focusSnap together. Once the focus is in the infinite view it may never escape since it is infinitely repeating.

Infinite Scrolling Caveats

Infinite repositioning does not (yet) support USS transforms (scaling, rotation, position). Please keep the scroll view at a scale of 1 at all times if you want to use infinity.

If you want to use **focusSnap** and infinite scrolling then you should set a **repetition delay** or else fast switching of repeated elements may make the scroll view scroll in the opposite direction.

The reason for this is that in the background the elements are repositioned (teleported from one side to the other) while scrolling. If however the focus moves faster than the scroll speed then it may happen that an element is focused before it has been moved to the other side and then the scroll animation will animate to the old position. To avoid that you can increase the repetition delay or decrease the snap duration. Ideally the repetition delay is bigger than or equal to the snap duration.

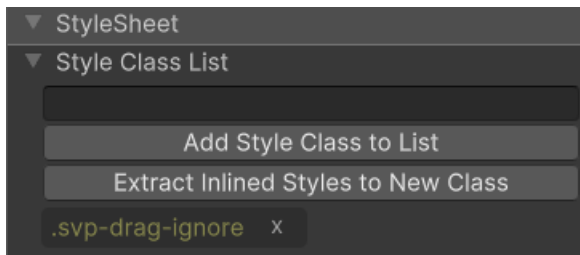


Dragging

Usually all areas (and children) within a scroll view are draggable.

How to make elements NOT draggable

If you want to make an element not draggable (like a slider) then you can do this by adding the „svp-drag-ignore“ USS class.



Since version 1.3.5 the svp-drag-ignore class is applied recursively, meaning that if it is added to any element it will also effect all the children. This is useful for interactive elements like sliders. You can re-active dragging with the „svp-drag-allow“ class. Both are recursive and allow takes precedence over ignore.

Events

There are some custom events in addition to the normal scroll view events you can register to.

SnapEvent

The SnapEvent is triggered before a snap is executed (start of snap animation or instant snap).

Example:

```
void OnEnable()
{
    var labels = ScrollView.Query<Label>().Build();
    foreach (var label in labels)
    {
        label.RegisterCallback<SnapEvent>(onSnap);
    }
}

void OnDisable()
{
    var labels = ScrollView.Query<Label>().Build();
    foreach (var label in labels)
    {
        label.UnregisterCallback<SnapEvent>(onSnap);
    }
}

private void onSnap(SnapEvent evt)
{
    Debug.Log("Snapped to: " + evt.target);
}
```

Known Issues

Please be aware that UI Toolkit is still under heavy development by Unity. If you can please always upgrade to the newest LTS version. The newer the better for UI Toolkit. Especially Unity 2021 and below still have some quirks with UI Toolkit. Even Unity itself does not yet recommend using UI Toolkit for production.

Releasing the drag handle in a scroller does not trigger a Snap on MouseUp.

This is a bug I am investigating. Sadly there is no solution yet. However you can register a callback yourself and call `Snap()` (or better `SnapDelayed()`) on the scroll view which will then trigger the snap.

Frequently Asked Questions

If you have any issues that are not listed here then please don't hesitate to write to me at [office\[at\]kamgam.com](mailto:office[at]kamgam.com)

My scroll view does not snap to the children in my grid.

By default the snapping will use the DIRECT children on the content container for snapping. If you have your children in a container (like a grid) then you will have to tell the scroll view how to find the snap targets. You can do that by setting the „snap target classes“. Then it will use a query to find all elements matching any of the given class names.



My buttons are not snapped to if focused. I can not change the focused element with the arrow keys or the controller.

In some versions of Unity 2022 there was a bug in the EventController by Unity which broke the focusing of elements via keyboard or controller. Please upgrade to the current [LTS](#), it was fixed in one of the more recent releases. - If the problem still persists even after upgrading the please ask in the Unity forum or write to support ([office\[at\]kamgam.com](mailto:office[at]kamgam.com)).

Why should 'selectableScrollbars' and 'focusSnap' not be combined?

Focus snap searches for elements outside the view port and selects those and scrolls the scroll view to them. If a scroll bar is focusable it is not clear whether the scroll bar should be selected or the hidden element outside the viewport should be selected. The code can not resolve this ambiguity.

I have added some dynamic children to the scroll view but they are not draggable.

The scroll view needs to keep track of its children and may have missed some. Usually it picks them up automatically (if they are added to the root of the „contentContainer“).

If the content you are adding is inserted into a nested child of the contentContainer then you'll have to call the „RefreshAfterHierarchyChange()“ method on the scroll view after adding them. This will force the scroll view to refresh the tracked children.

Can I mix regular ScrollView and ScrollViewPro?

Kinda, yes, you can BUT the scroll view pro usually assumes it is interacting (contains or is child of) other scroll view pros. I recommend using scroll view pro all the way. However, I have regular scroll view mixing in my test scenarios, it's just not as well tested as scroll view pros and there may be edge cases that can not be fixed due to ScrollView being changed and maintained by Unity.