DTT
APPS WEB GAMES

SINCE 2010

Paginated Scroll View

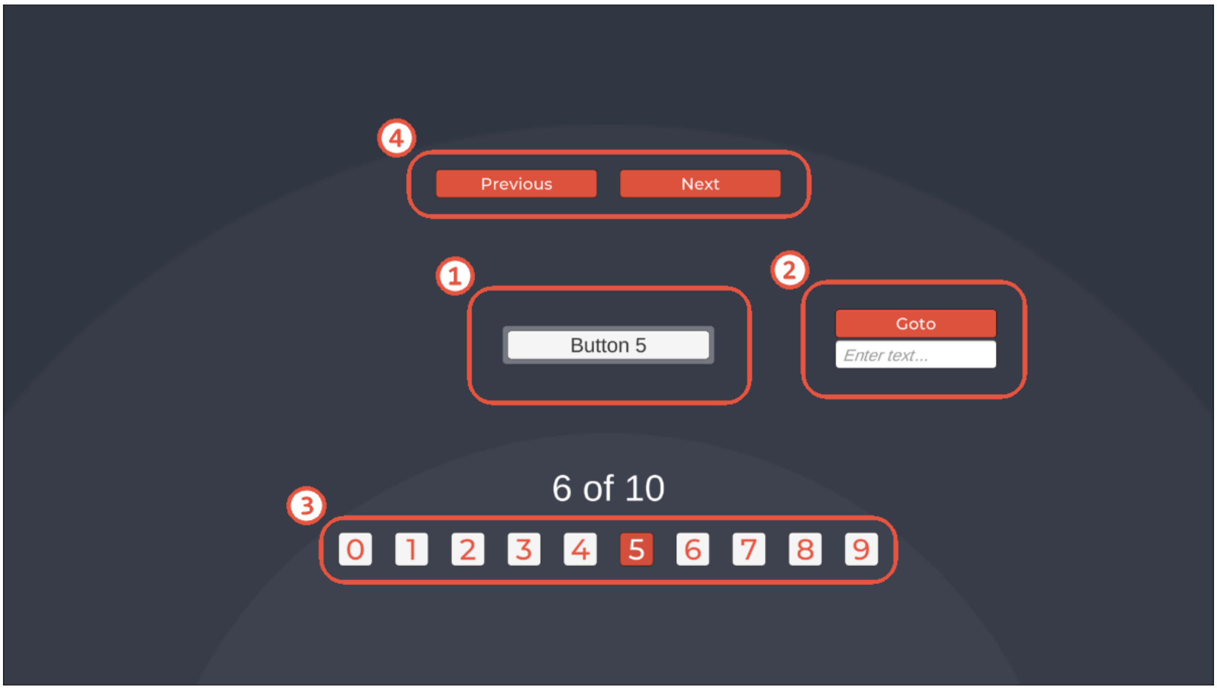
Documentation | 19-05-2022

Table of Contents

1. Get started quickly

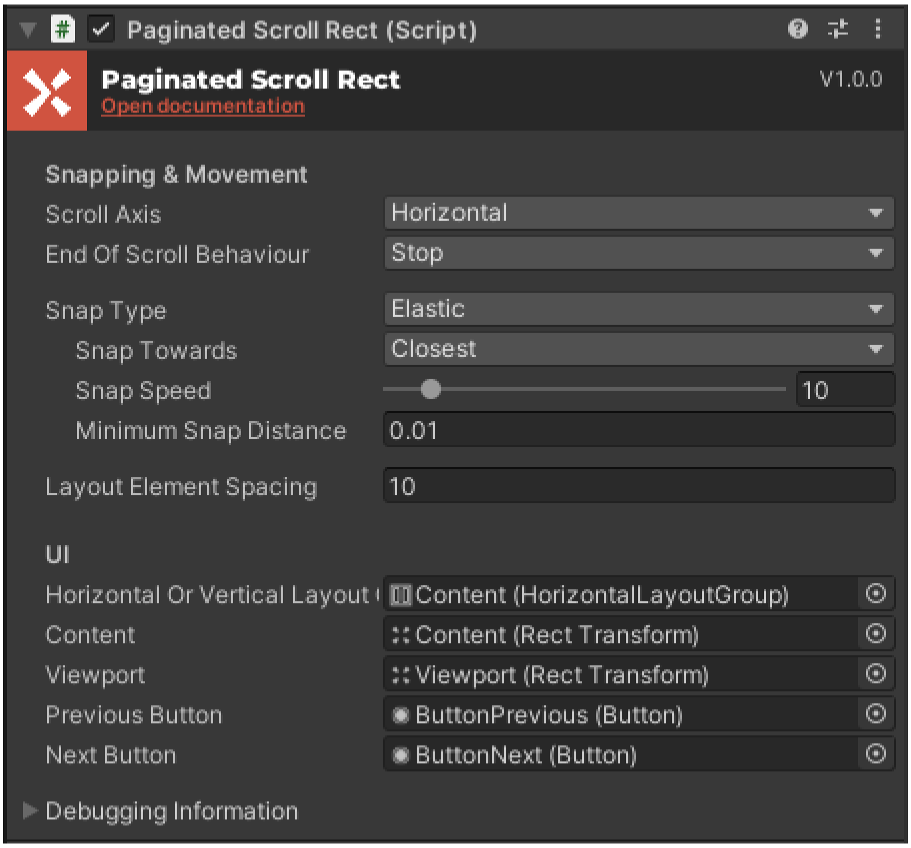
The **Paginated Scroll View** contains a demo scene that shows how the paginated scrolling view works with different navigation behaviors. Open the demo scene and press play. You have multiple ways of navigating through the scroll view:

1. By dragging elements up/down or left/right depending on which direction the scroll axis is set to.
2. By typing the page number and pressing the “Goto” button.
3. By clicking the page number boxes.
4. By clicking the “Previous” and “Next” buttons.



Once in play mode, you can change the **Paginated Scroll Rect** editor settings for the scrolling behavior. As seen in the editor window, there are options for changing the scroll axis, scrolling behavior, animations attributes for snapping or easing inertia of scroll movement, etc.

When the scrolling axis is changed during runtime, the layout group component of the *Content* child game object changes as well, depending on which direction the scrolling axis is changed to.



Paginated Scroll Rect (Script)

Paginated Scroll Rect V1.0.0
Open documentation

Snapping & Movement

Scroll Axis: Horizontal
End Of Scroll Behaviour: Stop
Snap Type: Elastic
Snap Towards: Closest
Snap Speed: 10
Minimum Snap Distance: 0.01
Layout Element Spacing: 10

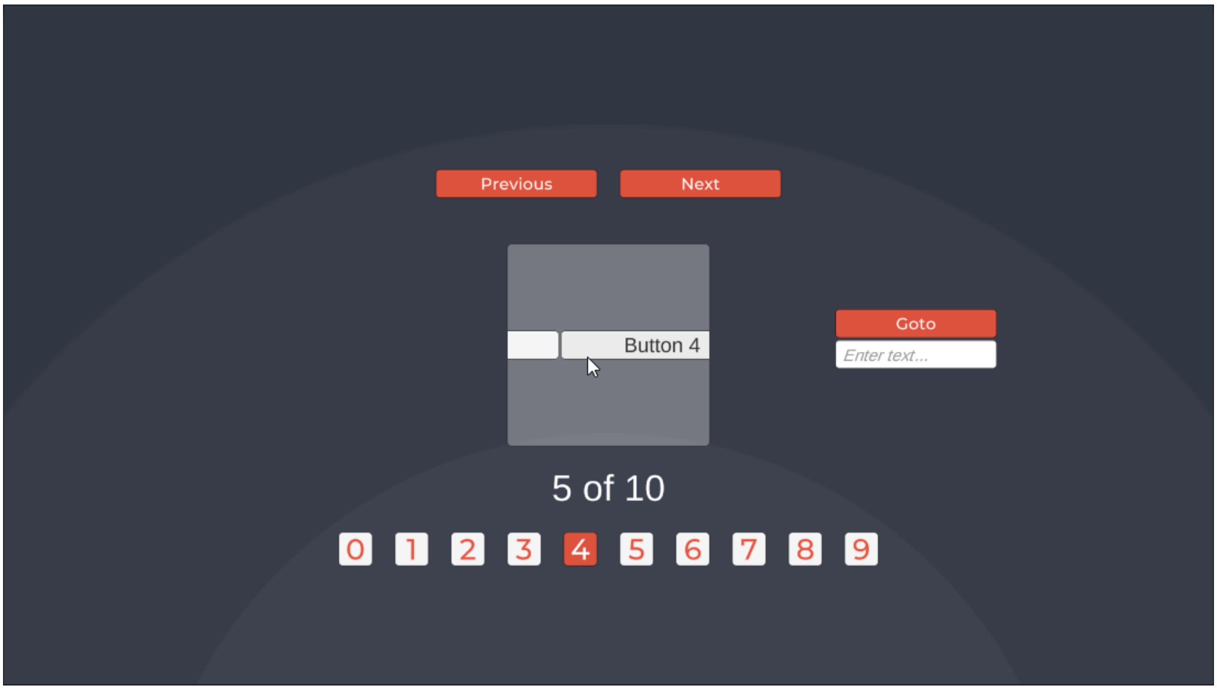
UI

Horizontal Or Vertical Layout: Content (HorizontalLayoutGroup)
Content: Content (Rect Transform)
Viewport: Viewport (Rect Transform)
Previous Button: ButtonPrevious (Button)
Next Button: ButtonNext (Button)

2. Introduction

DTT Paginated Scroll View is an asset that allows you to easily implement scrolling behavior for your UI elements. The main navigation behavior is the dragging of elements in the scroll view, the other three navigating behaviors are optional, meaning that you can include them based on your needs or not use them at all.

The main scrolling behavior logic resides in the PaginatedScrollRect.cs, this makes it possible to tweak the scrolling axis, speed, easing in/out of the pages, inertia of scrolling, elasticity, etc. all in one editor window.

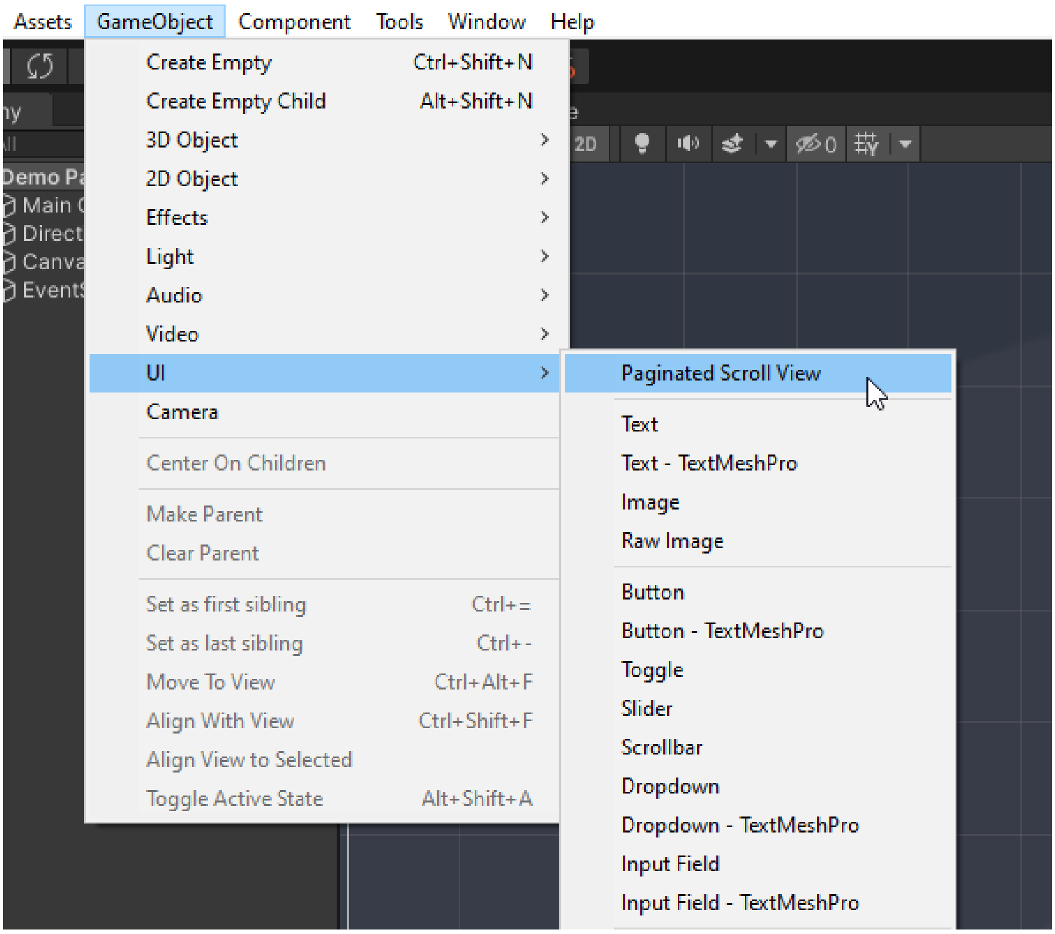


3. Set-Up

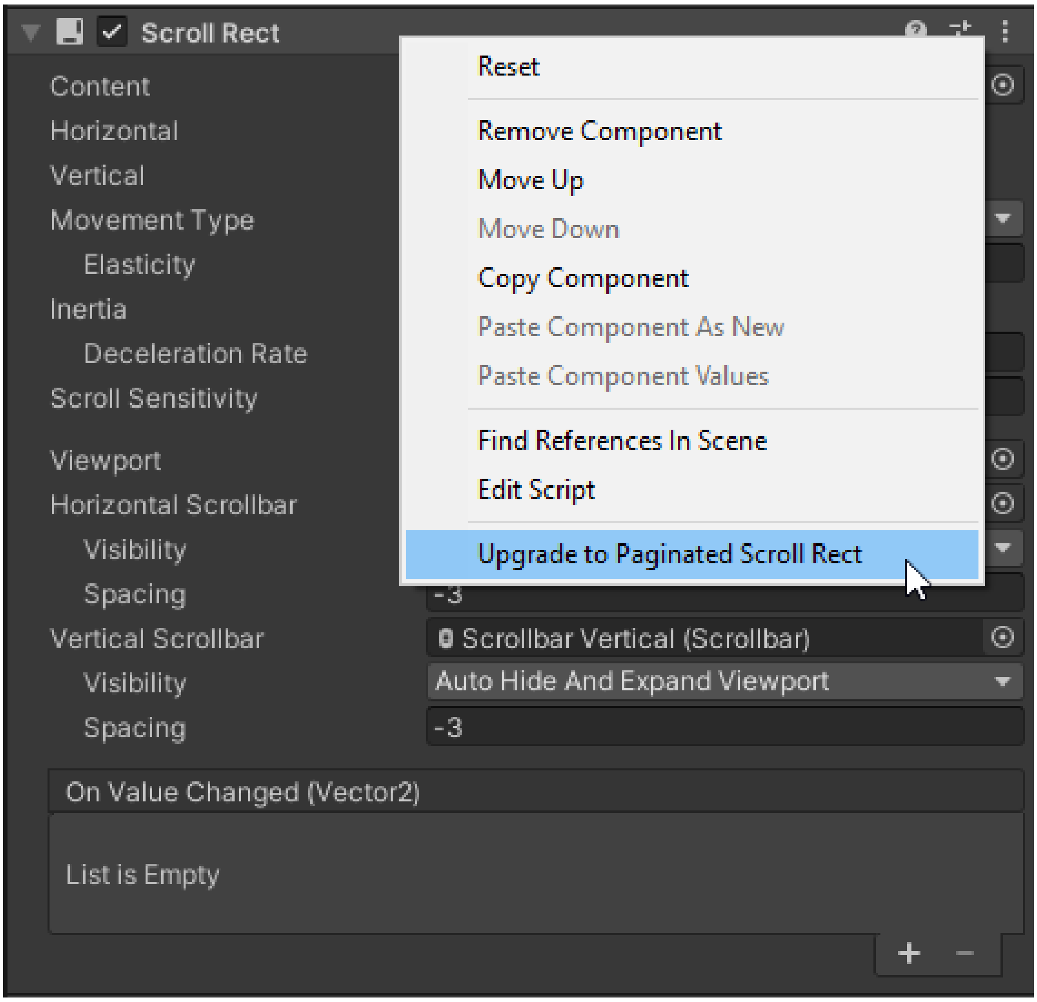
Paginated Scroll view.

The Paginated Scroll Rect game object can be added to the scene in two different ways:

1. The first method and the easiest for creating a Paginated Scroll Rect game object is by clicking on the ***GameObject*** menu in the toolbar of the editor. You can find the menu option by navigating through the drop menu in this order "***GameObject -> UI -> Paginated Scroll View***".



2. The second method to add a Paginated Scroll Rect game object, is to create a default ***Scroll View*** UI game object in your scene's canvas, then go to the inspector of the ***Scroll View*** game object and right-click on the ***Scroll Rect*** component; a drop menu will pop up with the option of upgrading this component to a ***Paginated Scroll Rect*** component. This will remove the horizontal and vertical scrollbars from the default ***Scroll View*** game object, but that will not be needed at all.



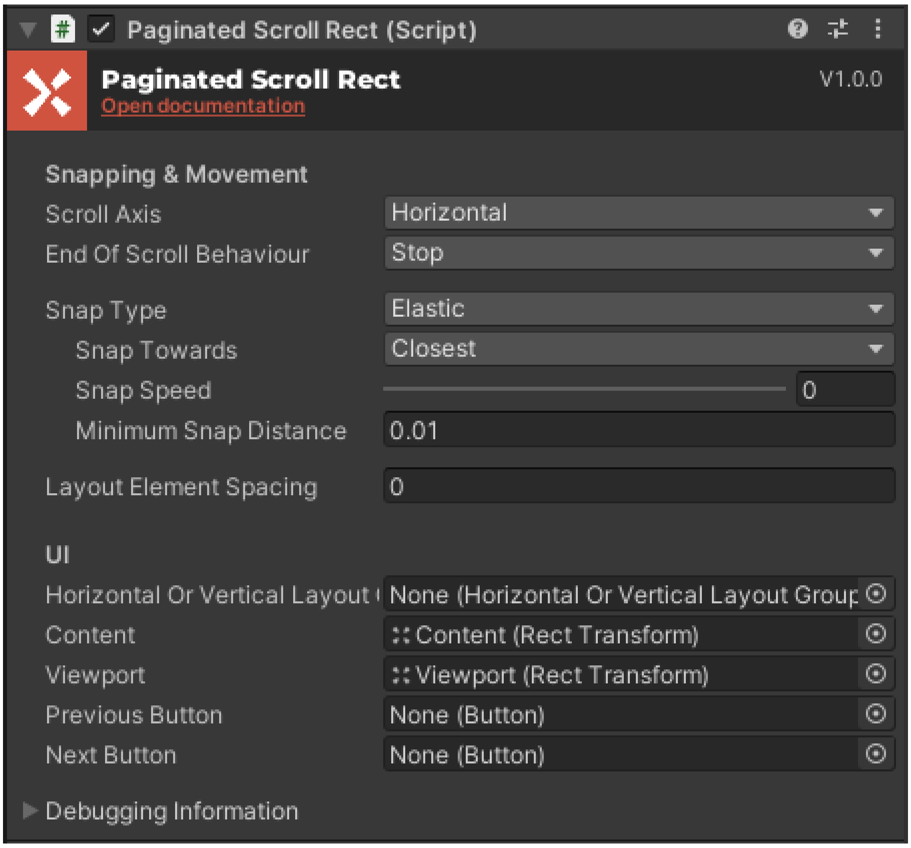
Spacing: -3

Vertical Scrollbar: Scrollbar Vertical (Scrollbar)

Visibility: Auto Hide And Expand Viewport

Spacing: -3

The ***Previous Button*** and ***Next Button*** are not necessary for this component to work, these are references only if you decide to implement a scrolling behavior with buttons.



Paginated Scroll Rect (Script)

Paginated Scroll Rect V1.0.0

Open documentation

Snapping & Movement

Scroll Axis: Horizontal

End Of Scroll Behaviour: Stop

Snap Type: Elastic

Snap Towards: Closest

Snap Speed: 0

Minimum Snap Distance: 0.01

Layout Element Spacing: 0

UI

Horizontal Or Vertical Layout: None (Horizontal Or Vertical Layout Group)

Content: Content (Rect Transform)

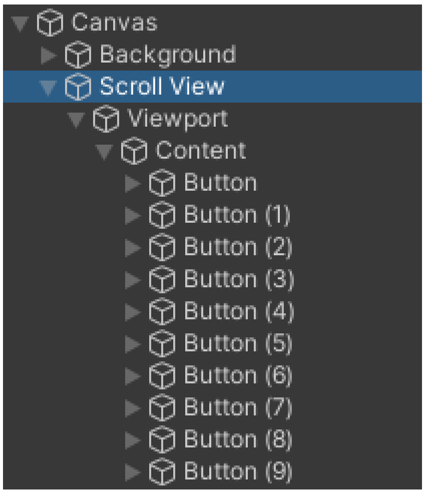
Viewport: Viewport (Rect Transform)

Previous Button: None (Button)

Next Button: None (Button)

Debugging Information

Now all you have to do is to place your elements or pages as children of the ***Content*** game object that is a child of the ***Scroll View*** game object.



```
Canvas
  Background
  Scroll View
    Viewport
      Content
        Button
        Button (1)
        Button (2)
        Button (3)
        Button (4)
        Button (5)
        Button (6)
        Button (7)
        Button (8)
        Button (9)
```

Naviagtion options.

DTT Paginated Scroll View provides multiple ways to navigate the scroll view. The differnt options consist of the following:

1. Dragging,
2. Previous and Next buttons.
3. Individual page buttons (Page Dots).
4. Search by input.

Each of the navigation options can be used seperatly from oneanother. This means that you don't have to all of them but that you can if you want.

Dragging.

The dragging navigation doesn't need anything to set up, simply create a Paginated Scroll View and start draging. depending on the options you choose in the Paginated Scroll View component you can drag the view from side to side or up and down.

Previous and Next buttons.

The Previous and Next button navigation is simple to set up. Once you created a Paginated Scroll View, there will be a field for both the previous and next button. Create two seperate buttons and drag their button components into these fields.

Each time the next button is clicked the Paginated Scroll View will scroll to the next page and the other way around for the previous button.

Individual page buttons (Page Dots).

The page dots are a bit more advanced to set up properly. First we recommend you to make a prefab of what you want your page buttons to look like. This prefab should contain only one **Button** component. Add the Page Dot Button script to the same gameObject that holds the **Button** component. The on and off color values decide the filling of the button, on is the color of the selected page and off for the not selected pages. the text color within the button will get the opposite value.

No that the prefab is complete we can add a Page Dot Manager to the scene. Create an **gameObject** and add the Page Dot Manager component to it. Provide the script with the prefab and the Paginated Scroll View you want to scroll through. It is also recommended to add a **Horizontal Layout Group** to the objecct to automatically spread out the Page Dots. Now hit the refresh button in the Page Dot Manager component and your Page Dots will be created.

Search by input.

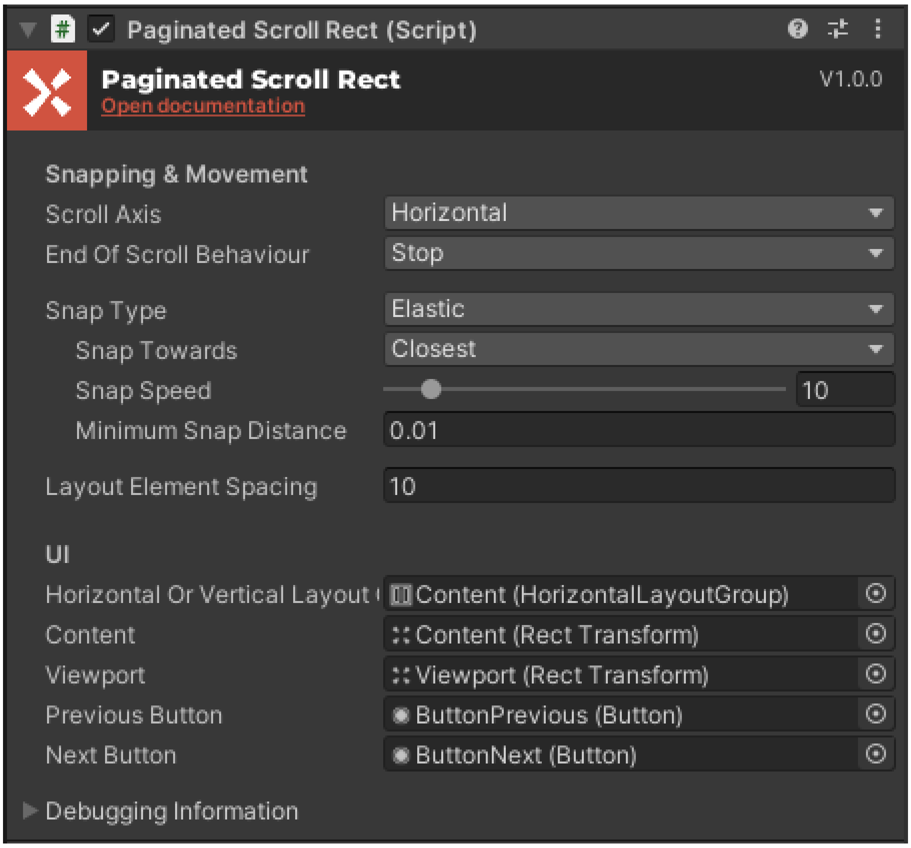
The implement the search by input navigation add an **Inputfield** to the scene. To this **Inputfield** add the Goto Page component, the component has two fields one for the Paginated Scroll View and one for the **Inputfield** you just created. Set the content type for the **Inputfield** to 'interger number'.

Now you can choose to add a **Button** to confirm for the search or to make it search for the page after pressing enter. Add the **Inputfield** object to the event you want it to listen to, for a **Button** this would be the **OnClick** event. After adding the **Inputfield** select GotoPage.SetPage in the event.

4. Editor

Overview

This is the DTT Paginated Scroll View Editor. It is a tool with which you can adjust your scrolling behavior for your elements inside the scroll view. In this chapter we will break down the functionality of each section of the editor in detail.



☑ Paginated Scroll Rect (Script)

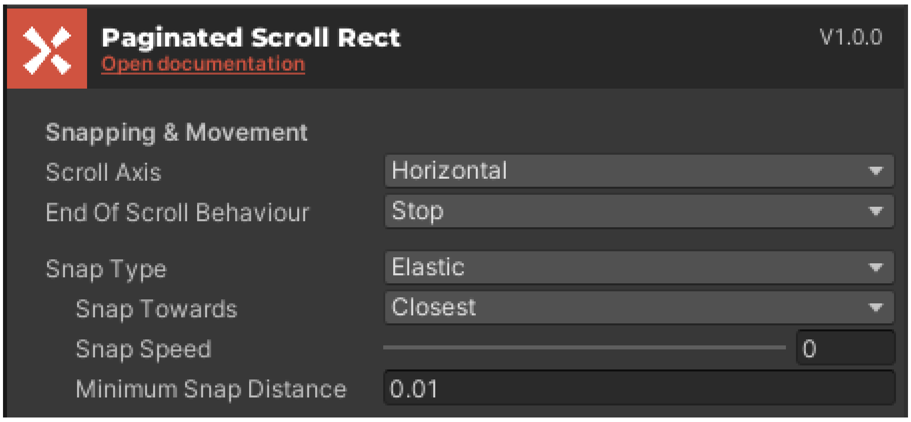
Paginated Scroll Rect V1.0.0
Open documentation

Snapping & Movement
Scroll Axis: Horizontal
End Of Scroll Behaviour: Stop
Snap Type: Elastic
Snap Towards: Closest
Snap Speed: 10
Minimum Snap Distance: 0.01
Layout Element Spacing: 10

UI
Horizontal Or Vertical Layout (: Content (HorizontalLayoutGroup)
Content: Content (Rect Transform)
Viewport: Viewport (Rect Transform)
Previous Button: ButtonPrevious (Button)
Next Button: ButtonNext (Button)

▶ Debugging Information

Snapping & Movements



Paginated Scroll Rect
Open documentation
V1.0.0

Snapping & Movement
Scroll Axis: Horizontal
End Of Scroll Behaviour: Stop
Snap Type: Elastic
Snap Towards: Closest
Snap Speed: 0
Minimum Snap Distance: 0.01

- **Scroll Axis** – sets the scrolling direction.

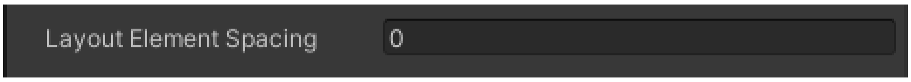
  – ***Horizontal***

  – ***Vertical***
- **End Of Scroll Behavior** – how the scrolling behavior should act when reaching the beginning or ending of the scroll view.

  – ***Stop*** – can not scroll beyond the first or last elements.

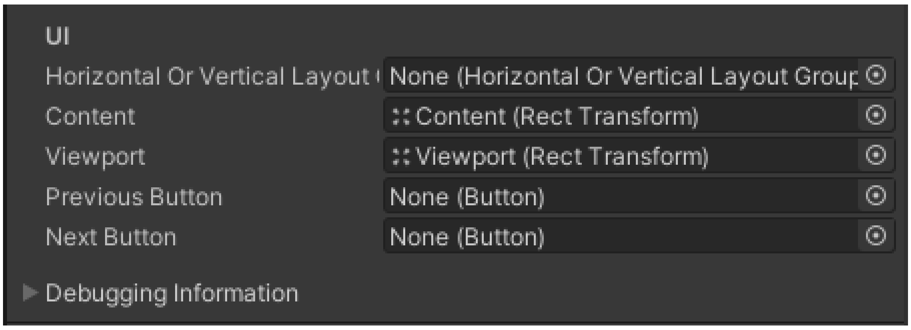
  – ***Elastic*** – can scroll more than the first or last elements but it swiftly goes back once the mouse button is released.
- **Snap Type** – how the pages should snap to the center of the scroll view when releasing the mouse button while scrolling.
  - **Elastic** – the elements snap elastically.
    - **Snap Towards** – to which element to snap to.
      - **Closest** – closest element to the center of view.
      - **Previous** – to the previous element.
      - **Next** – to the next element.
    - **Snap Speed** – the speed at which the elements are snapped
    - **Minimum Snapping Distance** – the minimum distance at which the snapping should be flagged as done.
  - **Clamped** – the elements snap directly without an elastic animation.

Layout Element Spacing – the spacing in between the elements within the *Content* game object's *LayoutGroup* component.



Layout Element Spacing	0

UI



UI

Horizontal Or Vertical Layout (	None (Horizontal Or Vertical Layout Grou
Content	Content (Rect Transform)
Viewport	Viewport (Rect Transform)
Previous Button	None (Button)
Next Button	None (Button)

▶ Debugging Information

These are references to the components that the *Scroll View* contains.

- **Horizontal Or Vertical Layout Group** – the layout group component of the *Content*'s game object. This is automatically assigned during runtime, depending on the selected *Scroll Axis*.
- ***Previous Button** – the button which will be used for navigating backwards through the elements in the scroll view.
- ***Next Button** – the button which will be used for navigating forwards through the elements in the scroll view.

***** – these are not necessary for the main scrolling (dragging) behavior to work, so they are optional depending on your needs.***

5. Support and feedback

If you have any questions regarding the use of this asset, we are happy to help you out. Always feel free to contact us at:

unity-support@d-tt.nl

(We typically respond within 1-2 business days)

We are actively developing this asset, with many future updates and extensions already planned. We are eager to include feedback from our users in future updates, be they 'quality of life' improvements, new features, bug fixes or anything else that can help you improve your experience with this asset. You can reach us at the email above.

Reviews and ratings are very much appreciated as they help us raise awareness and to improve our assets.

DTT stands for Doing Things Together

DTT is an app, web and game development agency based in the centre of Amsterdam. Established in 2010, DTT has over a decade of experience in mobile, game, and web based technology.

Our game department primarily works in Unity where we put significant emphasis on the development of internal packages, allowing us to efficiently reuse code between projects. To support the Unity community, we are publishing a selection of our internal packages on the Asset Store, including this one.

More information about DTT (including our clients, projects and vacancies) can be found here:

https://www.d-tt.nl/en/