

Namespace TS.PageSlider

Classes

[PageContainer](#)

This class represents a container for a page in a paginated view. It handles assigning content to the container and manages the active state of the contained page.

[PageDot](#)

This class represents a single dot indicator used for navigation in a paginated view. It provides properties for its active state and index, and events for state changes and presses.

[PageDotsIndicator](#)

This class manages a collection of page dots used for navigation in a paginated view. It provides methods to add, clear, and change the active dot.

[PageScroller](#)

The PageScroller class manages scrolling within a PageSlider component. It handles user interaction for swiping between pages and snapping to the closest page on release.

[PageSlider](#)

The PageSlider class manages a collection of pages within a PageScroller component. It provides functionality for adding, removing, and keeping track of pages, as well as handling page change events.

[PageView](#)

This class represents a view or page within a paginated view system. It provides events to signal changes in the active state of the page.

Class PageContainer

Namespace: [TS.PageSlider](#)

This class represents a container for a page in a paginated view. It handles assigning content to the container and manages the active state of the contained page.

```
public class PageContainer : MonoBehaviour
```

Inheritance

System.Object ← PageContainer

Methods

AssignContent(RectTransform)

Assigns content (RectTransform) to this container. If no content is provided, it creates a new GameObject with a RectTransform and a PageView component. The assigned content is then parented to this container and its properties are set to ensure proper positioning and scaling.

```
public void AssignContent(RectTransform content)
```

Parameters

content RectTransform

The RectTransform representing the content to be assigned.

ChangeActiveState(bool)

Calls the ChangeActiveState method on the contained PageView component with the provided active state.

```
public void ChangeActiveState(bool active)
```

Parameters

active [bool](#)

True to set the page to active, False to set it to inactive.

ChangingToActiveState()

Calls the ChangingToActiveState method on the contained PageView component, to signal a transition to an active state.

```
public void ChangingToActiveState()
```

ChangingToInactiveState()

Calls the ChangingToInactiveState method on the contained PageView component, to signal a transition to an inactive state.

```
public void ChangingToInactiveState()
```

Class PageDot

Namespace: [TS.PageSlider](#)

This class represents a single dot indicator used for navigation in a paginated view. It provides properties for its active state and index, and events for state changes and presses.

```
public class PageDot : MonoBehaviour
```

Inheritance

System.Object ← PageDot

Fields

OnActiveStateChanged

```
public UnityEvent<bool> OnActiveStateChanged
```

Field Value

UnityEvent<[bool](#)>

OnPressed

UnityEvent with an integer parameter that is invoked when the dot is pressed. The parameter represents the index of the pressed dot.

```
public UnityEvent<int> OnPressed
```

Field Value

UnityEvent<[int](#)>

Properties

Index

Gets or sets the index of the page dot within the paginated view.

```
public int Index { get; set; }
```

Property Value

[int](#)

IsActive

Gets the active state of the page dot.

```
public bool IsActive { get; }
```

Property Value

[bool](#)

Methods

ChangeActiveState(bool)

Changes the active state of the page dot and invokes the OnActiveStateChanged event.

```
public virtual void ChangeActiveState(bool active)
```

Parameters

active [bool](#)

True to set the dot active, False to set it inactive.

Press()

Invokes the OnPressed event with the dot's index when the dot is pressed.

```
public void Press()
```

Class PageDotsIndicator

Namespace: [TS.PageSlider](#)

This class manages a collection of page dots used for navigation in a paginated view. It provides methods to add, clear, and change the active dot.

```
public class PageDotsIndicator : MonoBehaviour
```

Inheritance

System.Object ← PageDotsIndicator

Fields

OnDotPressed

```
public UnityEvent<int> OnDotPressed
```

Field Value

UnityEvent<[int](#)>

Properties

IsVisible

Gets or sets the visibility of the PageDotsIndicator game object.

```
public bool IsVisible { get; set; }
```

Property Value

[bool](#)

Methods

Add()

Adds a new page dot indicator to the collection.

```
public void Add()
```

ChangeActiveDot(int, int)

Changes the active state of the page dots. It deactivates the dot at the 'fromIndex' and activates the dot at the 'toIndex'.

```
public void ChangeActiveDot(int fromIndex, int toIndex)
```

Parameters

fromIndex [int](#)

The index of the dot to deactivate.

toIndex [int](#)

The index of the dot to activate.

Clear()

Clears all the page dot indicators from the collection and destroys their game objects.

```
public void Clear()
```


Class PageScroller

Namespace: [TS.PageSlider](#)

The PageScroller class manages scrolling within a PageSlider component. It handles user interaction for swiping between pages and snapping to the closest page on release.

```
public class PageScroller : MonoBehaviour, IBeginDragHandler, IEndDragHandler
```

Inheritance

System.Object ← PageScroller

Implements

IBeginDragHandler, IEndDragHandler

Fields

OnPageChangeEnded

Event triggered when a page change ends. The event arguments are the index of the current page and the index of the new active page.

```
public UnityEvent<int, int> OnPageChangeEnded
```

Field Value

UnityEvent<[int](#), [int](#)>

OnPageChangeStarted

```
public UnityEvent<int, int> OnPageChangeStarted
```

Field Value

UnityEvent<[int](#), [int](#)>

Properties

Content

Gets the RectTransform of the content being scrolled within the ScrollRect.

```
public RectTransform Content { get; }
```

Property Value

RectTransform

Rect

Gets the rectangle of the ScrollRect component used for scrolling.

```
public Rect Rect { get; }
```

Property Value

Rect

Methods

OnBeginDrag(PointerEventData)

```
public void OnBeginDrag(PointerEventData eventData)
```

Parameters

eventData PointerEventData

OnEndDrag(PointerEventData)

```
public void OnEndDrag(PointerEventData eventData)
```

Parameters

`eventData` `PointerEventData`

SetPage(int)

```
public void SetPage(int index)
```

Parameters

`index` [int](#)

Class PageSlider

Namespace: [TS.PageSlider](#)

The PageSlider class manages a collection of pages within a PageScroller component. It provides functionality for adding, removing, and keeping track of pages, as well as handling page change events.

```
public class PageSlider : MonoBehaviour
```

Inheritance

System.Object ← PageSlider

Fields

OnPageChanged

```
public UnityEvent<PageContainer> OnPageChanged
```

Field Value

UnityEvent<[PageContainer](#)>

Properties

Rect

Gets the rectangle of the PageSlider component.

```
public Rect Rect { get; }
```

Property Value

Rect

Methods

AddPage(RectTransform)

Adds a new page to the PageSlider. The content argument specifies the RectTransform of the content to be displayed on the new page.

```
public void AddPage(RectTransform content)
```

Parameters

content RectTransform

The RectTransform of the content to be displayed on the new page.

Clear()

Removes all pages from the PageSlider and clears the associated PageDotsIndicator (if exists).

```
public void Clear()
```

Class PageView

Namespace: [TS.PageSlider](#)

This class represents a view or page within a paginated view system. It provides events to signal changes in the active state of the page.

```
public class PageView : MonoBehaviour
```

Inheritance

System.Object ← PageView

Fields

OnActiveStateChanged

UnityEvent with a boolean parameter that is invoked when the active state of the page changes. The parameter is True if the page becomes active, False if it becomes inactive.

```
public UnityEvent<bool> OnActiveStateChanged
```

Field Value

UnityEvent<[bool](#) >

OnChangingToActiveState

```
public UnityEvent OnChangingToActiveState
```

Field Value

UnityEvent

OnChangingToInactiveState

UnityEvent that is invoked when the page is about to transition to the inactive state.

```
public UnityEvent OnChangingToInactiveState
```

Field Value

UnityEvent

Methods

ChangeActiveState(bool)

Invokes the OnActiveStateChanged event with the provided active state.

```
public void ChangeActiveState(bool active)
```

Parameters

active [bool](#)

True to signal the page becoming active, False for inactive.

ChangingToActiveState()

Invokes the OnChangingToActiveState event to signal that the page is about to become active.

```
public void ChangingToActiveState()
```

ChangingToInactiveState()

Invokes the OnChangingToInactiveState event to signal that the page is about to become inactive.

```
public void ChangingToInactiveState()
```