Unity Page Slider

Core Components

- Page Slider: This core component manages your entire paginated UI system. It houses your pages, triggers page change events, and provides overall control.
- Page Scroller: This component handles the smooth scrolling of your pages. You can configure scrolling behavior and receive notifications when a page transition begins or ends
- Page View: This serves as the individual page within your paginated system. Here, you build and customize the layout of each page. It also triggers events when a page is activated or deactivated.

Optional Components

- Page Dots Indicator: This optional component provides a collection of dots to visually represent your pages and aid navigation. Users can interact with these dots to jump to specific pages.
- Page Dot: This represents a single dot within the Page Dots Indicator component. Each dot corresponds to a specific page within your paginated UI.

Setup

You can create a Page Slider in two ways: either with a predefined set of pages and content, or by loading content dynamically into the slider at runtime. Check the following guides for step-by-step tutorials:

- Using Predefined Content
- Implementing Dynamic Content

Helpful links

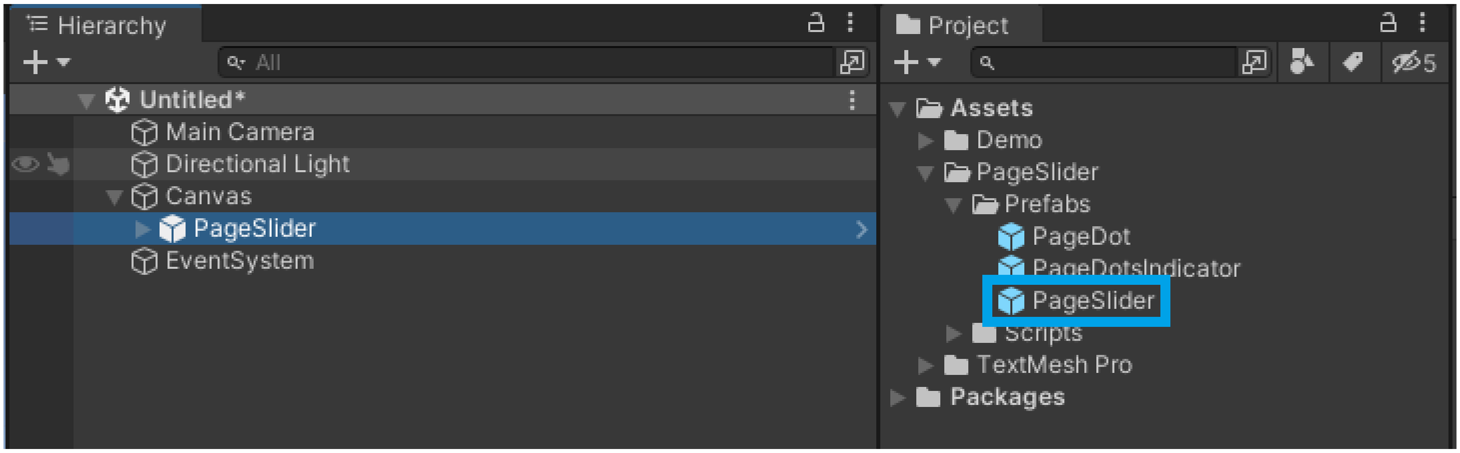
- For a more in-depth explanation, check out this video.
- Explore the API Reference for more details.
- Modify the source code available on GitHub.
- Install directly from the Unity Asset Store.
- Get in touch with me at tomazsaraiva.com.

Using Predefined Content

This guide walks you through creating a Page Slider with a predefined set of pages and content.

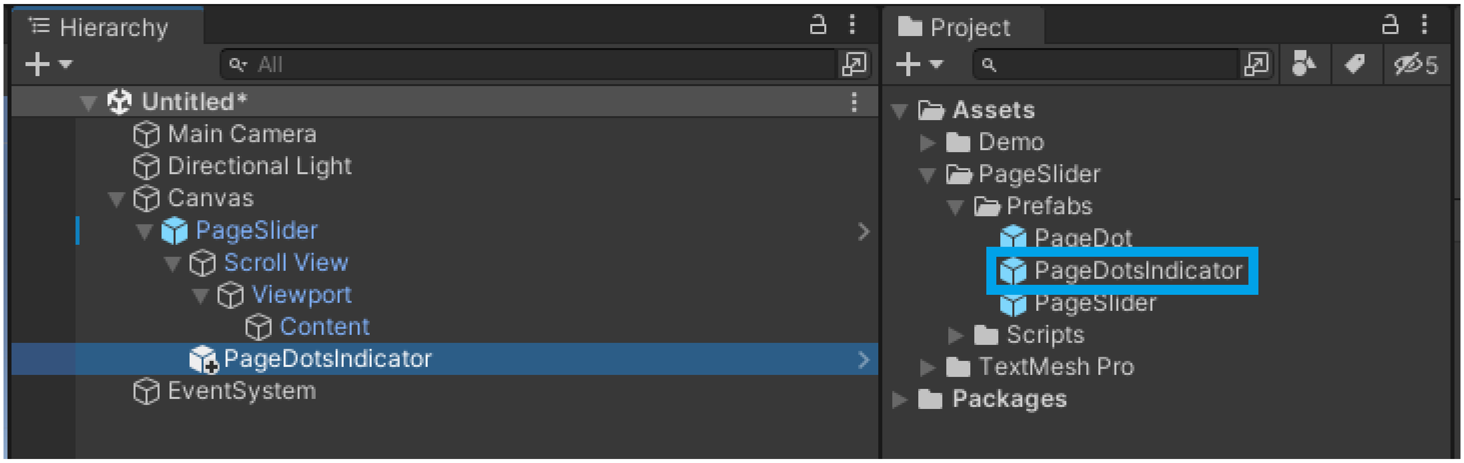
1. Add the Page Slider

1. Create a new `Canvas` GameObject.
2. In the Project window, locate the `PageSlider` prefab within your project's folder structure: `PageSlider/Prefabs/`.
3. Drag the `PageSlider` prefab from the Project window and make it a child of the `Canvas` in the Hierarchy.

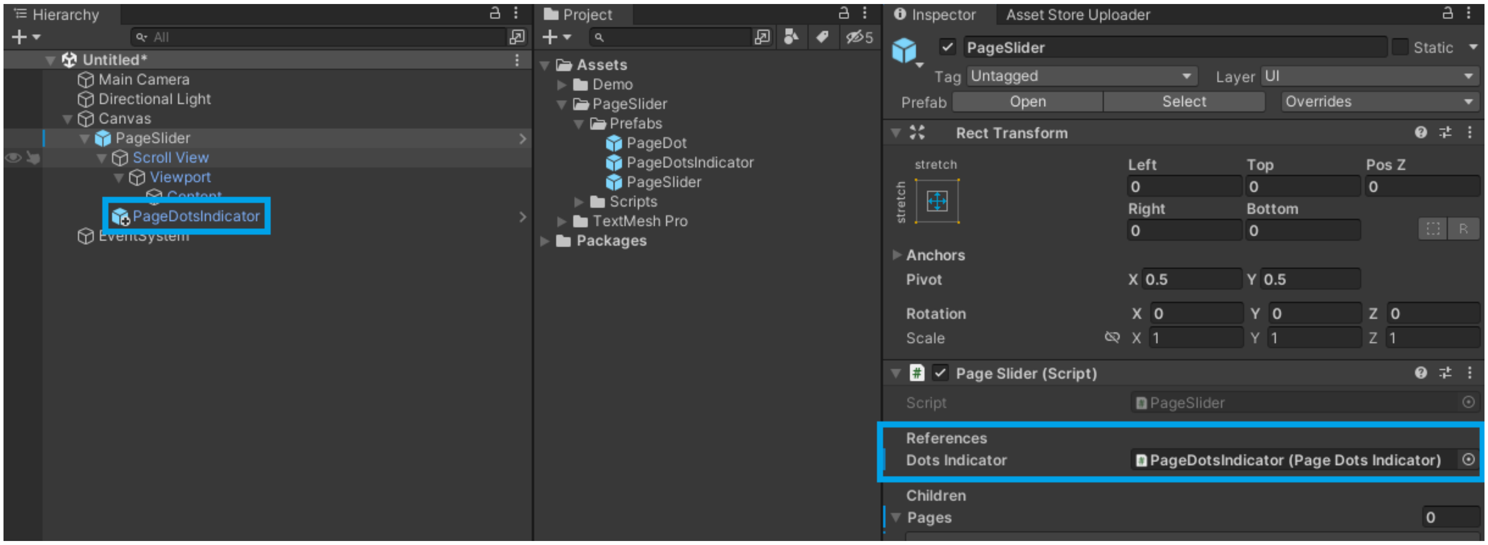


2. (Optional) Add Page Dots Indicator

1. In the Project window, locate the `PageDotsIndicator` prefab within your project's folder structure: `PageSlider/Prefabs/`.
2. Drag the `PageDotsIndicator` prefab and make it a child of the `PageSlider` in the Hierarchy.



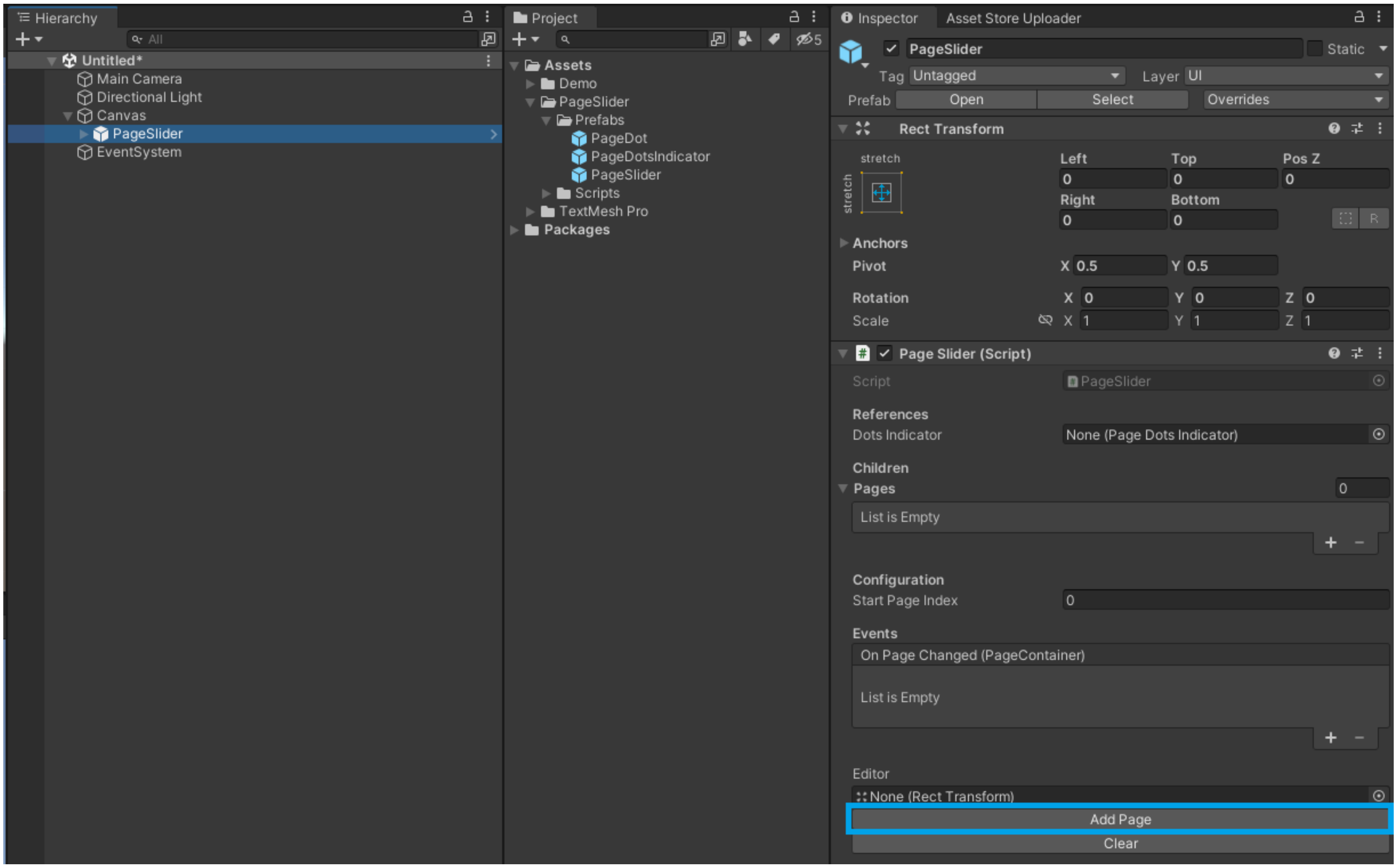
3. In the Inspector window, assign the `PageDotsIndicator` to the `Dots Indicator` field.



PageSlider Static
Tag Untagged Layer UI
Prefab
Rect Transform
Left 0 Top 0 Pos Z 0
Right 0 Bottom 0
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Page Slider (Script)
Script PageSlider
References
Dots Indicator PageDotsIndicator (Page Dots Indicator)
Children
Pages 0

3. Add a new Page

1. Select the `PageSlider` GameObject.
2. Click the `Add Page` button in the Inspector window.

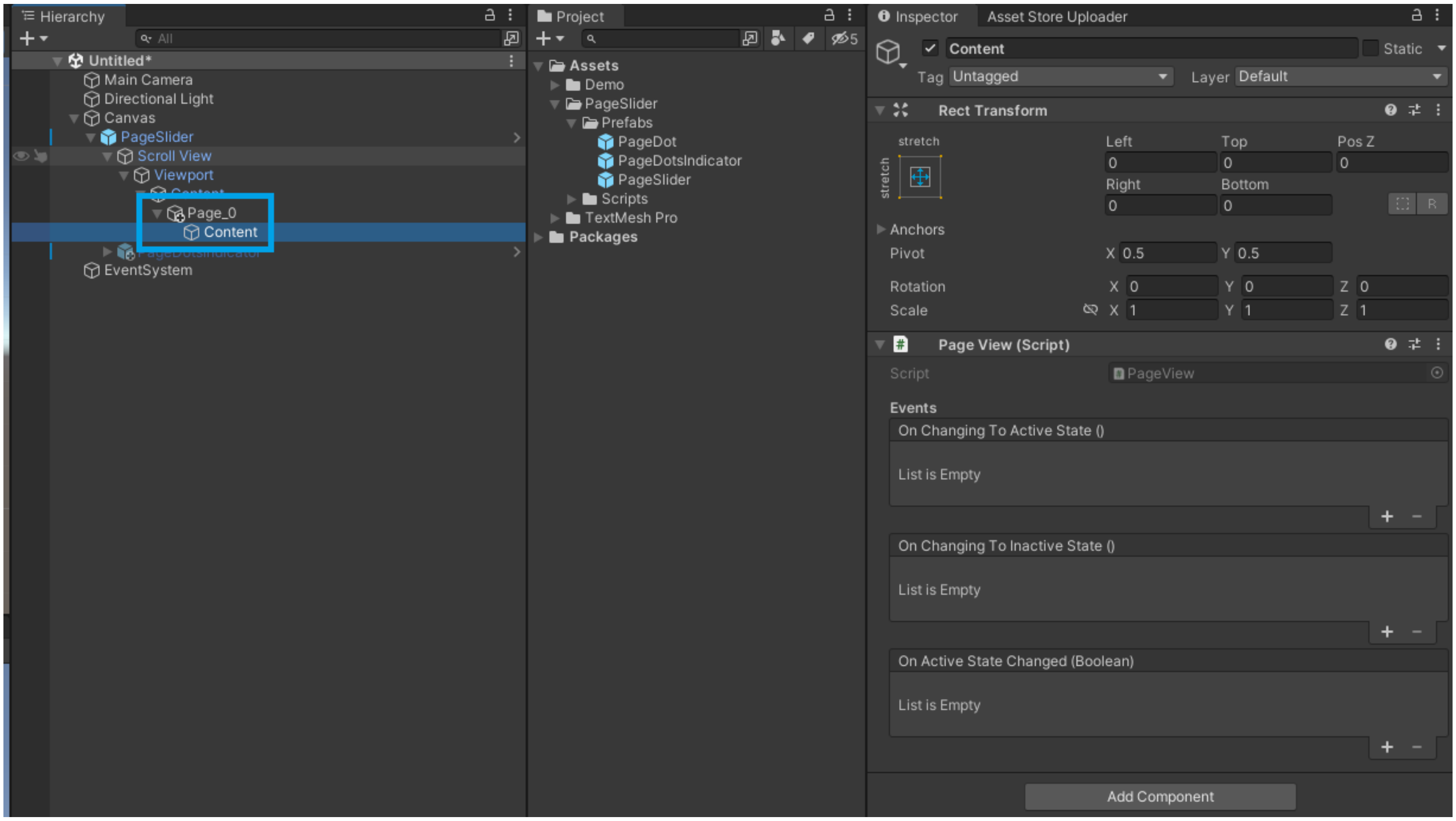


PageSlider
Tag Untagged
Layer UI
Prefab

Rect Transform
stretch
Left 0 Top 0 Pos Z 0
Right 0 Bottom 0
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Page Slider (Script)
Script PageSlider
References
Dots Indicator None (Page Dots Indicator)
Children
Pages 0
List is Empty
Configuration
Start Page Index 0
Events
On Page Changed (PageContainer)
List is Empty
Editor
None (Rect Transform)

3. Expand the `PageSlider` hierarchy to locate the newly created `Page_X/Content` GameObject.

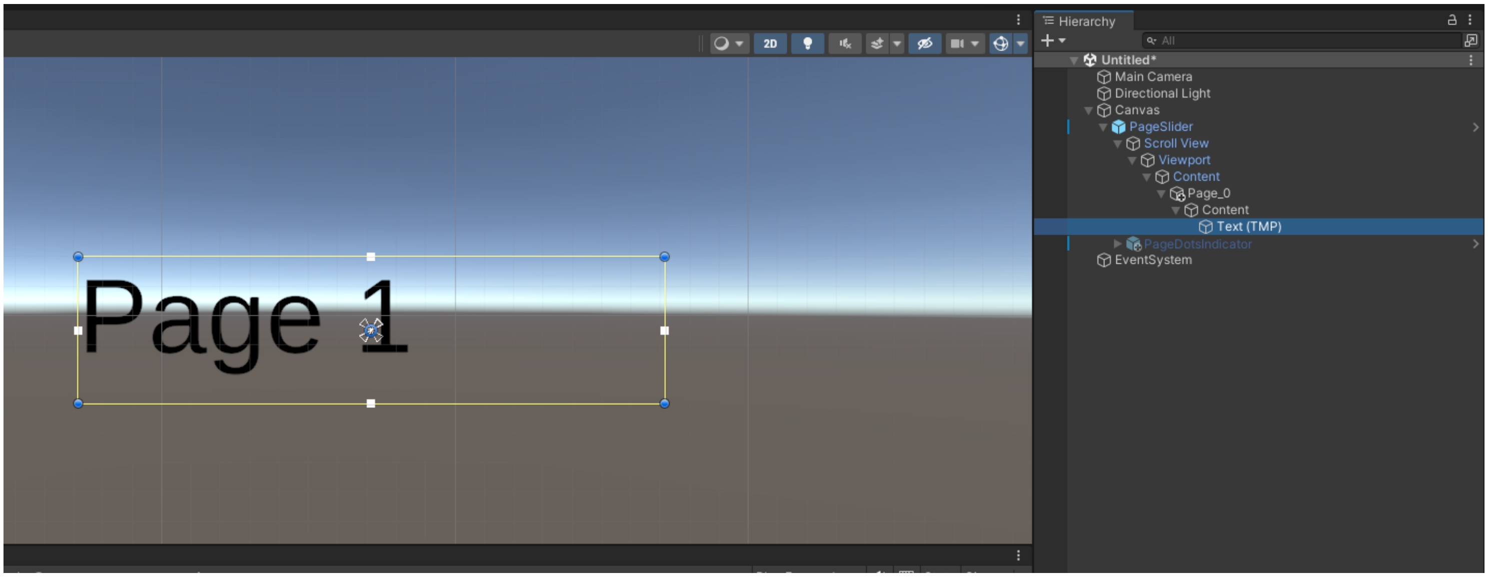


Content
Tag Untagged
Layer Default

Rect Transform
stretch
Left 0 Top 0 Pos Z 0
Right 0 Bottom 0
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Page View (Script)
Script PageView
Events
On Changing To Active State ()
List is Empty
On Changing To Inactive State ()
List is Empty
On Active State Changed (Boolean)
List is Empty

4. Select the `Content` GameObject and add your desired UI elements to configure the page layout.



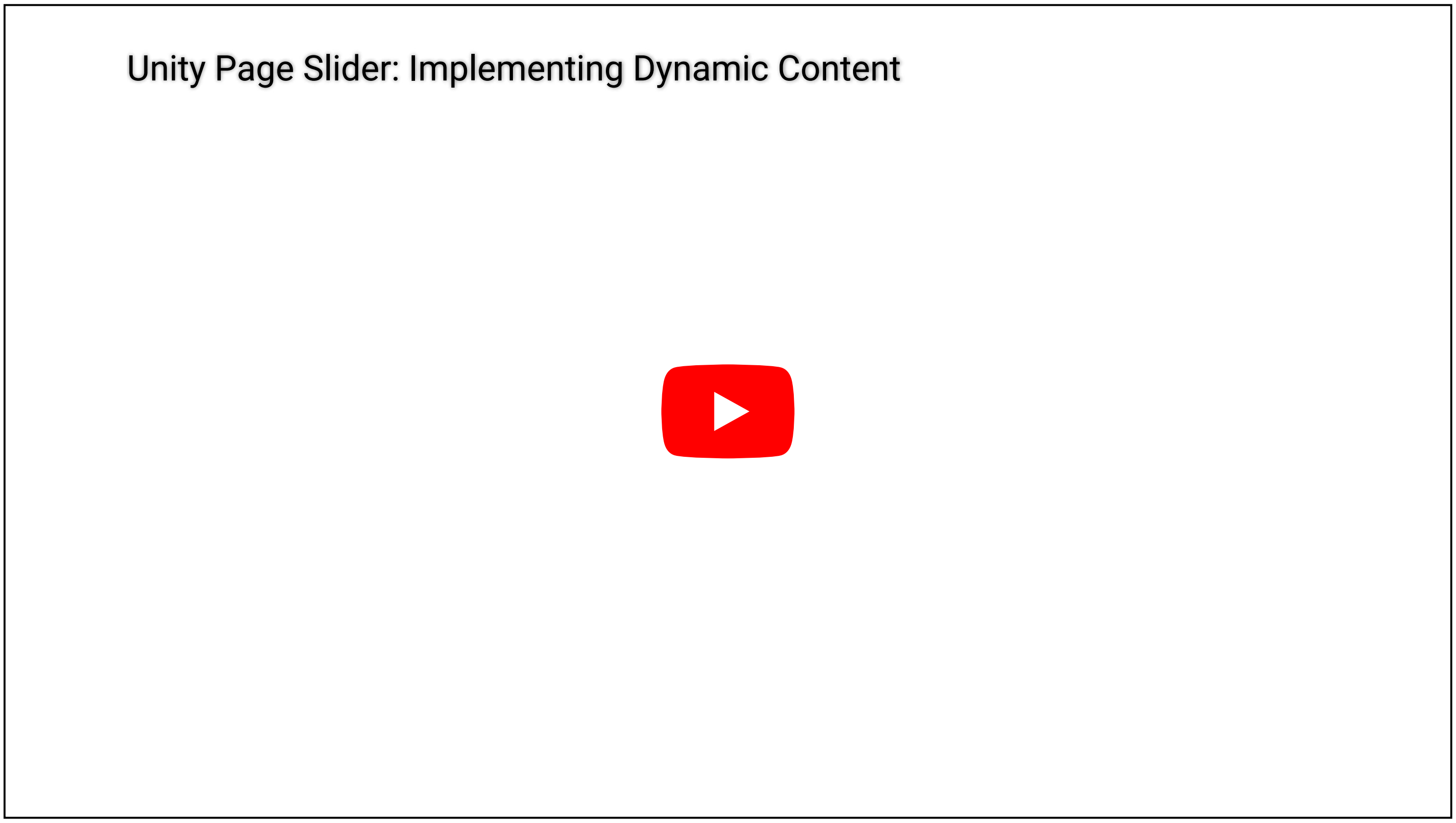
5. Repeat the previous steps to add more pages.

Helpful links

- Understand the main concepts by reading the getting started guide
- For a more in-depth explanation, check out this video.
- Explore the API Reference for more details.
- Modify the source code available on GitHub.
- Install directly from the Unity Asset Store.
- Get in touch with me at tomazsaraiva.com.

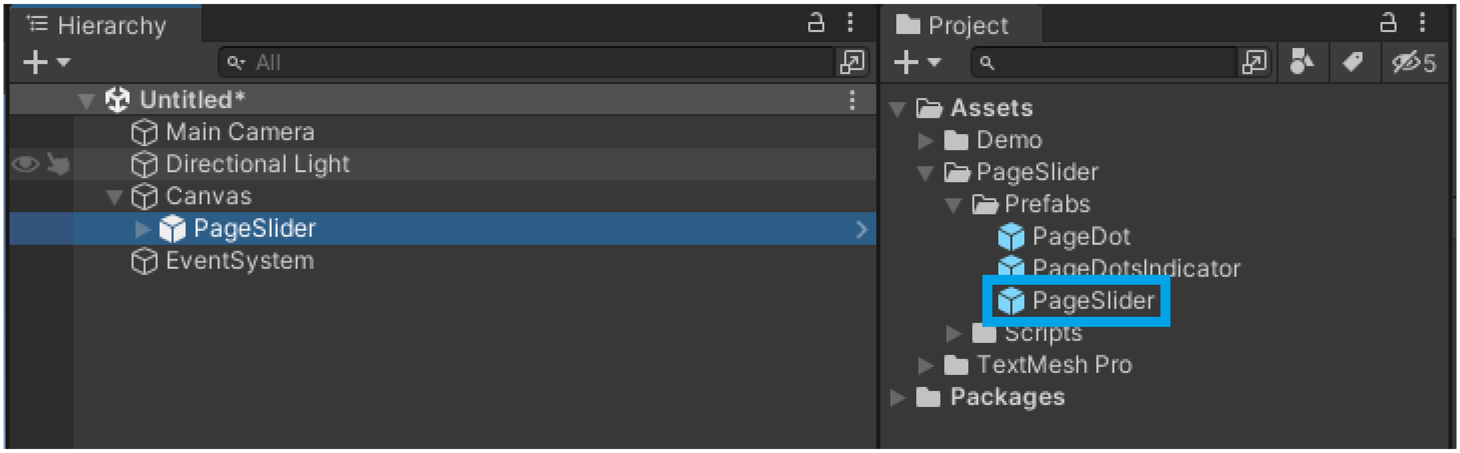
Implementing Dynamic Content

This tutorial explores loading content dynamically into your Page Slider at runtime, allowing for greater control and adaptability.



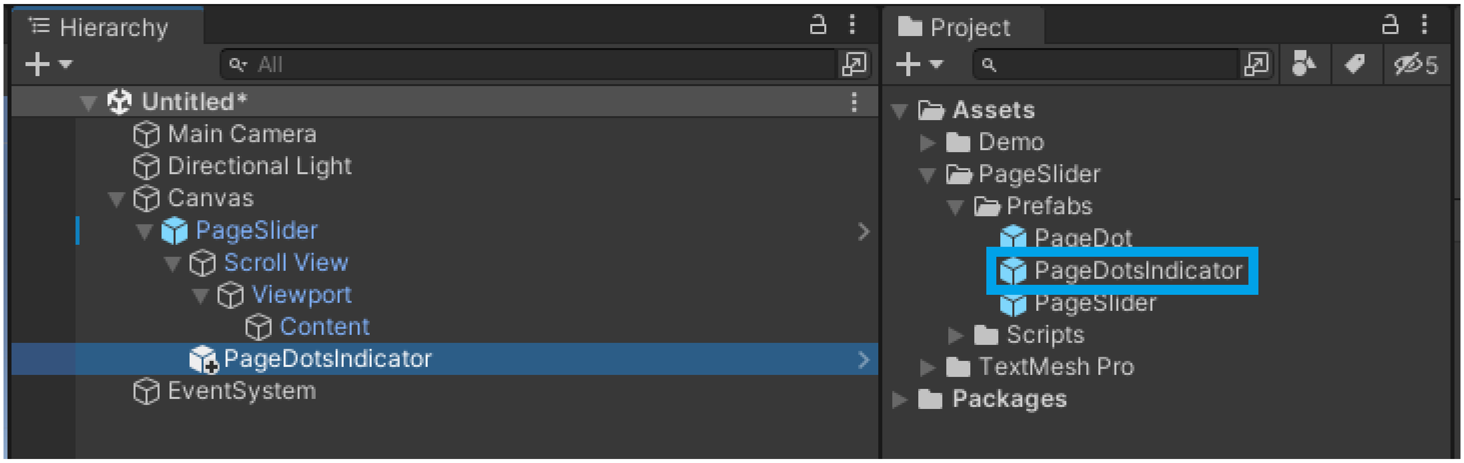
1. Add the Page Slider

1. Create a new `Canvas` GameObject.
2. In the Project window, locate the `PageSlider` prefab within your project's folder structure: `PageSlider/Prefabs/`.
3. Drag the `PageSlider` prefab from the Project window and make it a child of the `Canvas` in the Hierarchy.

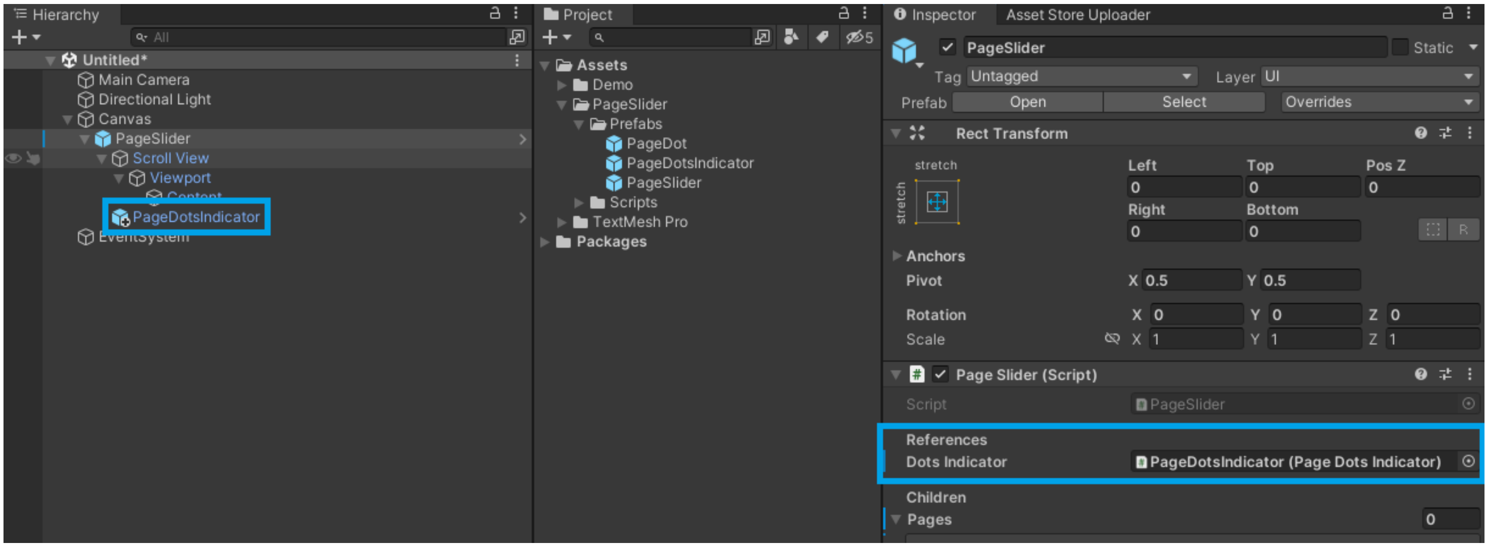


2. (Optional) Add Page Dots Indicator

1. In the Project window, locate the `PageDotsIndicator` prefab within your project's folder structure: `PageSlider/Prefabs/`.
2. Drag the `PageDotsIndicator` prefab and make it a child of the `PageSlider` in the Hierarchy.



3. Select the `PageSlider` GameObject.
4. In the Inspector window, assign the `PageDotsIndicator` to the `Dots Indicator` field.



PageSlider

Tag Untagged Layer UI

Prefab

	Left	Top	Pos Z
	0	0	0
	Right	Bottom	
	0	0	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Script PageSlider

References

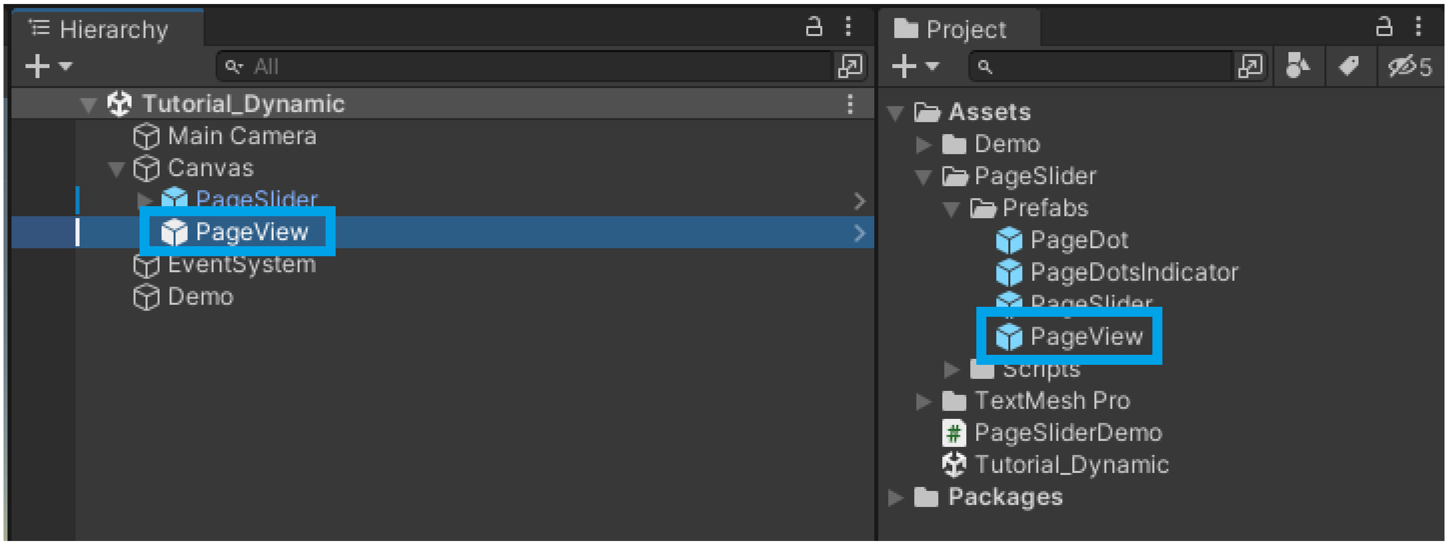
Dots Indicator PageDotsIndicator (Page Dots Indicator)

Children

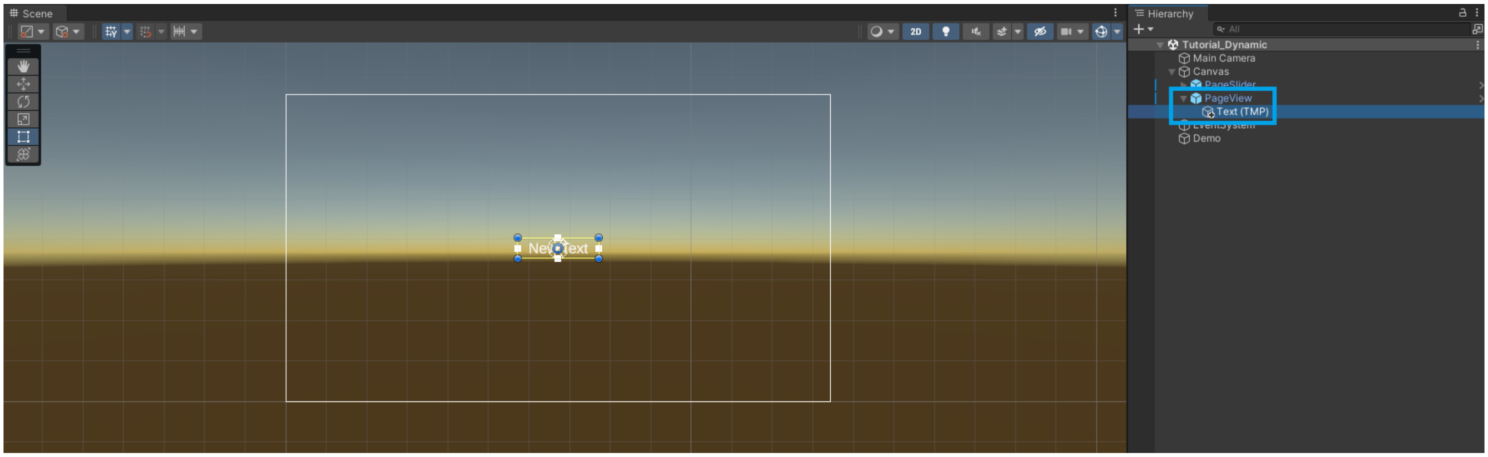
Pages 0

3. Create a Page View

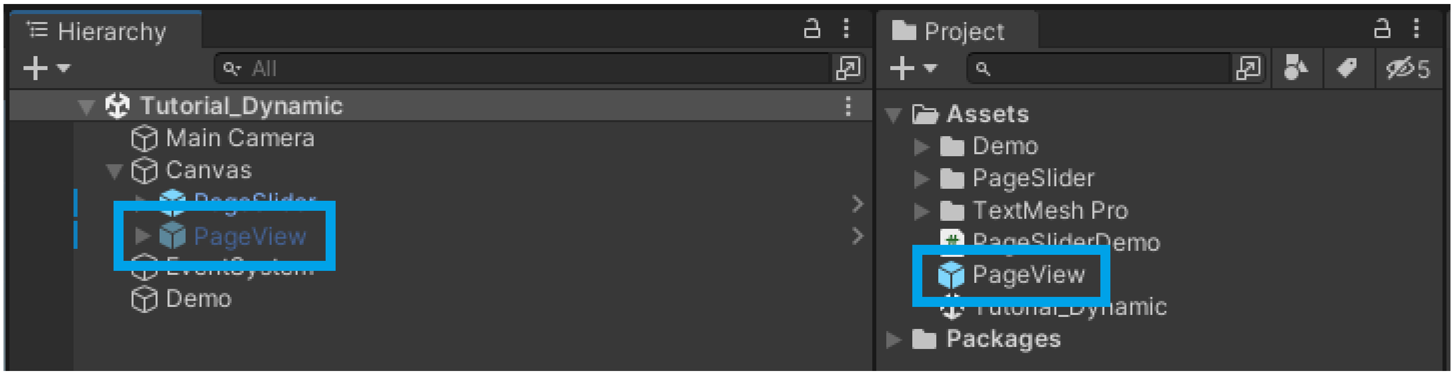
1. Select the `Canvas` GameObject.
2. In the Project window, locate the `PageView` prefab within your project's folder structure: `PageSlider/Prefabs/`.
3. Drag the `PageView` prefab from the Project window and make it a child of the `Canvas` in the Hierarchy.



4. Add your desired UI elements to configure the `PageView` layout. In this example, I'm adding a single Label, but the layout can be as complex as you need.

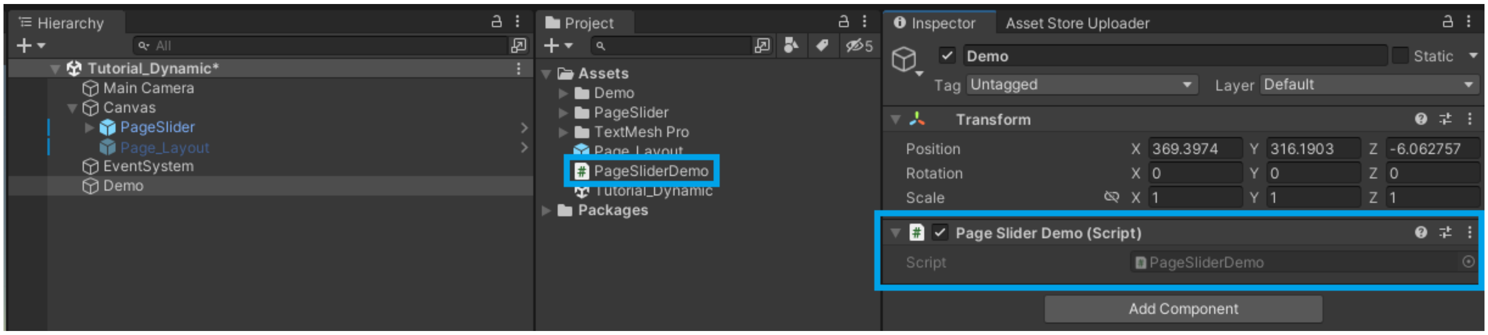


1. Create a new prefab for the `PageView` layout. The Page Slider will instantiate this prefab at runtime to generate each page.
2. Disable or remove the page layout from the Scene.



4. Add Pages at runtime

1. Create a new C# script (or use an existing one).



☑ Demo

Tag Untagged Layer Default

Position X 369.3974 Y 316.1903 Z -6.062757
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Script PageSliderDemo

1. Create a reference to the `PageSlider` and the `PageView` prefab.
2. Instantiate the pages that you want and configure their properties.
3. Add the pages to the `PageSlider` using the AddPage method. Please note it accepts a `RectTransform` component.

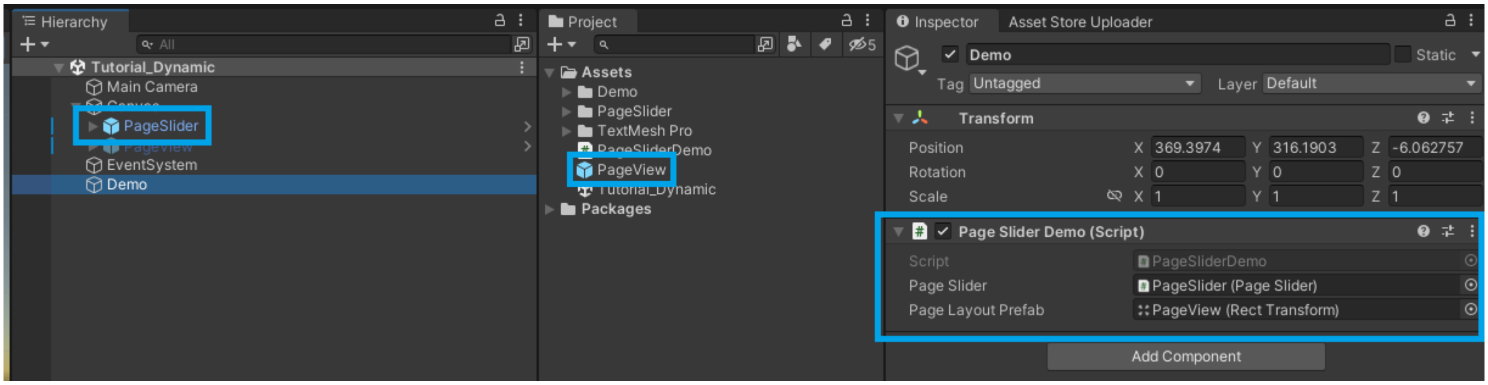
```
using TMPro;
using TS.PageSlider;
using UnityEngine;

public class PageSliderDemo : MonoBehaviour
{
    public PageSlider _pageSlider;
    public PageView _pageView;

    void Start()
    {
        for (int i = 0; i < 3; i++)
        {
            var page = Instantiate(_pageView);
            page.GetComponentInChildren<TextMeshProUGUI>().text = i.ToString();

            _pageSlider.AddPage((RectTransform)page.transform);
        }
    }
}
```

1. Assign the `PageSlider` and the `PageView` prefab references.



Demo — Static
Tag: Untagged — Layer: Default

Transform

	X	Y	Z
Position	369.3974	316.1903	-6.062757
Rotation	0	0	0
Scale	1	1	1

Page Slider Demo (Script)

Script	PageSliderDemo
Page Slider	PageSlider (Page Slider)
Page Layout Prefab	PageView (Rect Transform)

6. Run the project and the pages will be added dynamically to the `PageSlider`.

Note: For more complex cases you should create a custom class for the page layout. Check the `Demo_Dynamic` and `Demo_Lazy` available on the GitHub repository.

Helpful links

- Understand the main concepts by reading the getting started guide
- For a more in-depth explanation, check out this video.
- Explore the API Reference for more details.
- Modify the source code available on GitHub.
- Install directly from the Unity Asset Store.
- Get in touch with me at tomazsaraiva.com.