Documentation - v1.0

Scroll down for more

Content

1. Frequently Asked Questions (FAQ)

• Does Shift UI support URP/HDRP rendering?

Yes. There are some scenes for URP/HDRP as well. The only thing that doesn't work with it is the blur materials.

• I'm stuck and need help, what can I do?

If you can't find a solution for your problem in this doc, **contact me!** I'd gladly help to solve your issue.

• I can't modify the element, all of the values are staying the same. Why?

Some elements are managed by **UI Manager**. Values in the manager are universal, which means it'll affect any object that has **'UI Manager'** component. If you don't want that, you can delete the component from specific elements. It'll take its own unique values as soon as deleting the manager component.

• What platforms can I build for?

Shift UI works in builds for all platforms that listed in Unity build window, but the demo scenes are working out of the box for only Desktop, WebGL and Console (mainly Xbox) platforms. The scenes can be used for other platforms with a bit tweak though.

• I'm getting errors, why and how can I fix it?

It could be about anything. Make sure to import **TextMesh Pro** from package manager and its essentials from Window > TextMesh Pro. If you're still having the issue, contact me with some details.

• Are you going to support and add new stuff to the package?

Of course! There'll be update support for a while, including major ones.

• I'm getting low performance. Why?

If you're getting low performance only on editor, make sure to disable **'Update Values'** on **UI Manager** while not playing with the values on it. However, if you get low performance on build as well, it might be because of old hardware. You can disable all of the **blur** objects in this case, it'll double your FPS up. Basically, just type **'Blur'** to **hierarchy**, and disable all of them.

2. Quick Start

First of all, thanks for purchasing the package! If you need some help to get started, this is the right place.

2.1. Demo scenes

There are a couple of demo scenes included in the package. If you wish, you can start to work with them.

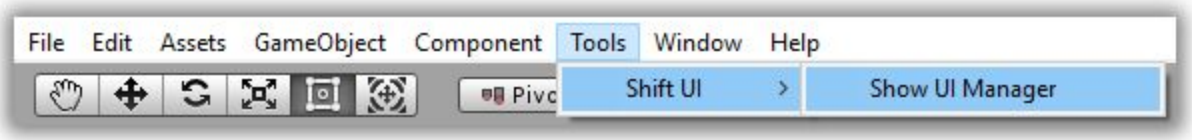
If you experience strange results like the image down below, don't worry, it's just a Unity bug. Disable and enable the **Canvas** object to fix this problem (or just hit play, it'll be fixed automatically). This is happening because of **Layout Group** and **Content Size Fitter**.

Note that there are lots of **Canvas Group** components in various panels. You'll have to set their alpha value to 1 in order to see them, and set 0 to make them invisible. So, basically, you have to maximize the alpha value while changing the content of a specific panel.

2.2. UI Manager

Do you want to change the appearance of the entire UI at the same time? Well, we got you covered. UI Manager is basically will change **every** single element which means you won't have to change stuff one by one.

You can open the window by clicking **Tools > Shift UI > Show UI Manager.** And you're all set! You can now expand the categories and start to change values.



Update Values

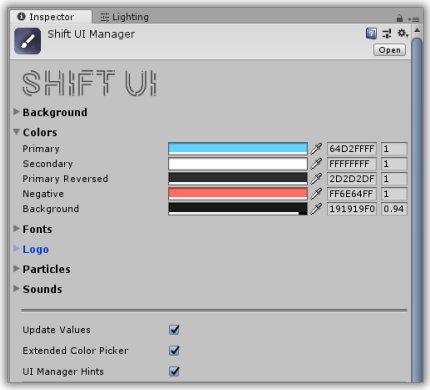
While this option is checked, UI Manager will be updating UI elements dynamically. If not, you won't be able to see any changes until you hit play (runtime). You can turn this off to gain more performance on editor, just don't forget to enable it while changing stuff. This feature is disabled in build mode, so it won't have any effect on builds.

Extended Color Picker

If you want to see more detailed color picker, enable this. This will be adding a hex code and an alpha slider right next to the color picker.

UI Manager Hints

If you want to see some tips about the manager, then you can enable this.



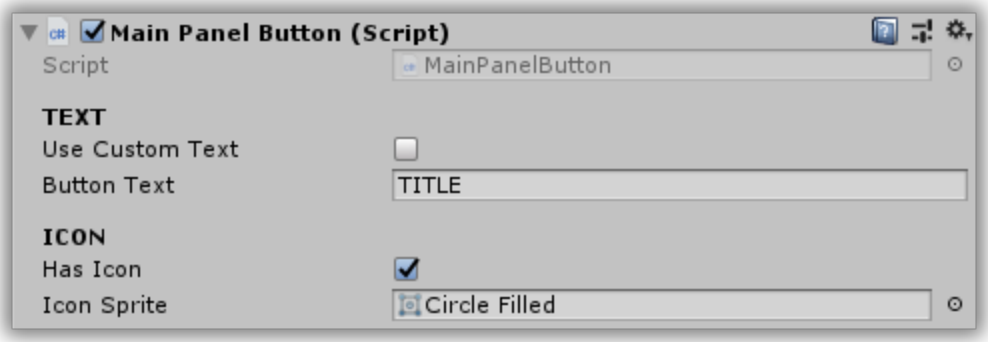
Note that UI Manager values are universal and will affect any object that contains UI Manager component.

3. UI Elements

There are lots of UI elements in Shift. While some of them are custom made, the others are developed through the default Unity UI elements. In this section, you'll learn more about the custom ones. If you wish, you can learn more about the default ones from the official Unity tutorials / documentation.

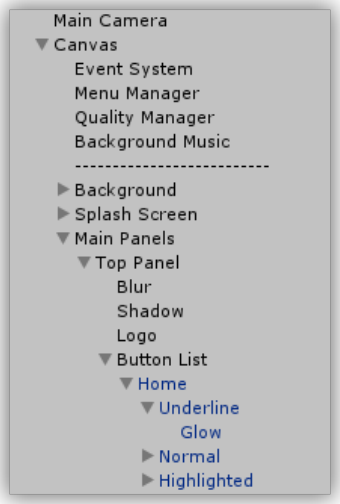
3.1. Changing content

In most cases, you can change the content from inspector. For example, instead of trying to find the text object, you can simply change it on inspector. This will save some time while changing complex elements. Here's an example:



3.2. Editing UI elements

Every UI object was made with native Unity UI, so you can customize or change them as much as you want. All of the objects are regularly categorized, so, if you want to change a specific thing, just search for it. After you're done, you can **'Apply'** prefab changes to change all of the connected objects.



3.3 Custom UI elements

There are some custom elements that are not included in the default Unity UI. In this section, you can learn more things about them.

3.3.1. Horizontal selector

Basically, think this thing as a dropdown, but the navigation is managed by only with next and previous controls. In my opinion, fits better than dropdown in most cases.

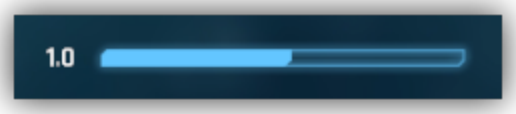
Items
You can add horizontal selector items to this list. If you wish, you can add functions to each item as well.

Saving
You can save the last selected value by checking this box. Note that every selector should has its own unique **Selector Tag** value.

Usage
It can be used with OnClick or you can call it from your script. **HorizontalSelector.ForwardClick();** or **HorizontalSelector.PreviousClick();** is what you're looking for.

3.3.2. Slider

Shift UI is using the native slider that comes with Unity UI, but we've added some new cool features into it.



You can find these new features from **Slider Manager** which is attached to the slider object. You can also save the last selected value by checking **Enable Value** box. Note that every slider should has its own unique **Slider Tag** value in order to save correctly.

3.3.3. Switch

Basically, think this thing as a toggle, but ‘modernized’.

Events
You can add On / Off events and invoke them at start from **Switch Manager**.

Saving
You can save the last selected value by checking **Save Value** box. Note that every switch should has its own unique **Switch Tag** value in order to save correctly.

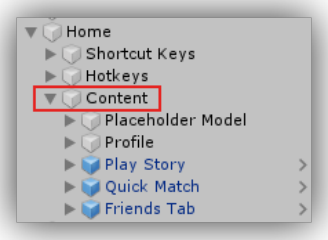
4. Panels & Windows

There are lots of UI elements in Shift. While some of them are custom made, the others are developed through the default Unity UI elements. In this section, you’ll learn more about the custom ones. If you wish, you can learn more about the default ones from the official Unity tutorials / documentation.

4.1. Editing panels

Wanna add your own content to an existing panel? Well, it’s pretty easy.

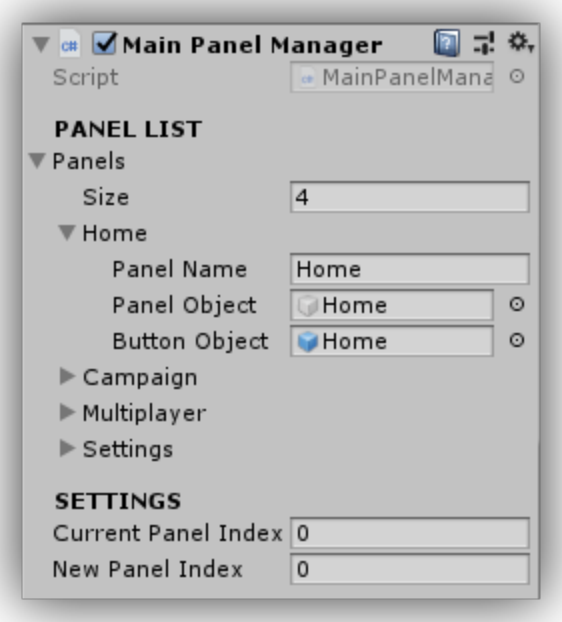
To add your own content to an existing panel, just drag your stuff under **[Panel Name] > Content object**. As long as your stuff is a child of **Content**, it’ll be automatically animated. Simple as that.



You can also change **Key Shortcut List** and **Key Helpers**. Key Shortcut List shows the button shortcuts at the bottom corner. Key Helpers object basically manages the hotkeys for shortcuts.

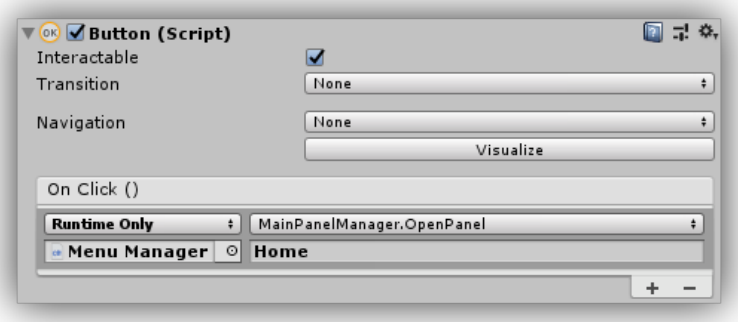
4.2. Adding panels

Duplicate an existing panel to create a new one. You'll also need a button to open the panel, so duplicate one of the existing buttons on the list and set the panel index. If you don't want to use a button, then you can give it a 'Dummy' (aka blank) object. After duplicating the objects, you can now assign the new one into **Main Panel Manager**.



Create a new **'Panels'** list item, and drag your panel & button to the fields. After doing these steps, you can now call your panel via button (OnClick) or from your script.

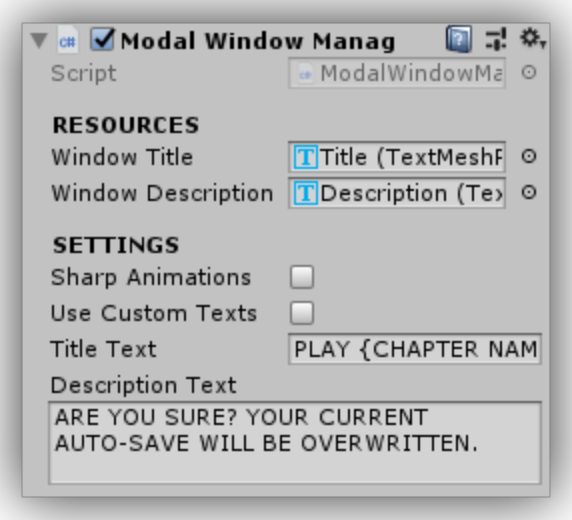
OnClick example:
Menu Manager > MainPanelManager.OpenPanel() > Your Panel String



Script example:
PanelManagerVariable.OpenPanel("Your Panel String");

4.3. Modal Windows

In order to change Modal Window content, you just need to change some values from **Modal Window Manager** (which is attached to the modal object).



In runtime, your window will be changing depending on the values. If you want to change those values within the main object, you can enable **'Use Custom Texts'**. You can call the window via OnClick or your script. You just need to call **ModalObject.ModalWindowIn()** or **ModalObject.ModalWindowOut()**.

5. Animations

Shift UI is using Unity Mecanim system, which is highly customizable. You can change or tweak the animations easily.

5.1. Editing animations

You can manage the animations from **'Animator'** tab (Window > Animation > Animator). Now, you can click an object that has animations and you'll see the animations states. That's it, select an animation and change or edit them as you want! You can learn more info about animator on official Unity docs.

6. Inputs

This package supports touch, keyboard/mouse and gamepad input events.

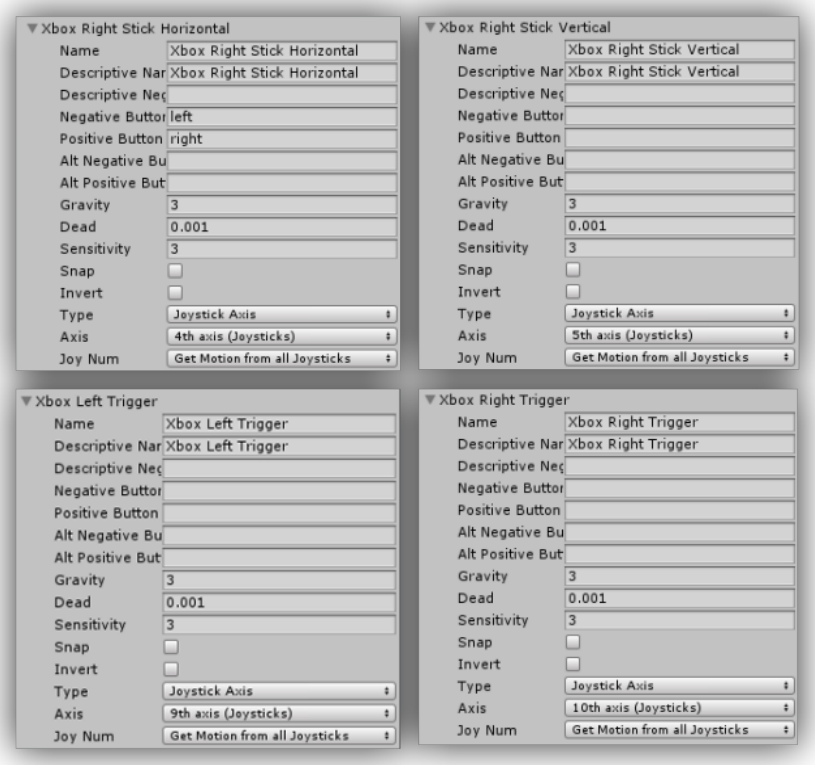
There's a component called **Gamepad Checker** which you can find it on **Menu Manager** object in the scene. Keyboard & Gamepad objects in the list will be disabled or enabled depending on the controller. For example, if gamepad is plugged, then keyboard objects will be disabled.

6.1. Key shortcut system

You can invoke a specific event when a key is pressed. You can find or add those events under **[Panel Name] > Hotkeys**. You can also add/duplicate shortcut keys to make them visible to the user (at the bottom).

6.2. Gamepad support

Gamepad support is managed by key shortcut system and virtual cursor. If you wish, you can enable the regular navigation by changing 'Navigation' to 'Automatic' on UI objects. In order to use all of the gamepad features, **you have to add these 4 new inputs**:



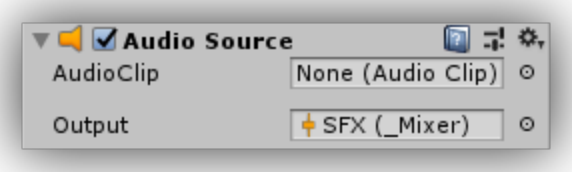
You can also download InputManager.asset file and replace with yours from here.

7. Quality Manager

Shift UI has an out of the box graphics/quality and audio manager system. You don't have to waste your time with creating them from scratch.

7.1. Audio manager

Audio settings are managed by an **Audio Mixer**. There are 3 states; Master, SFX and Music. You can assign one of the states to 'Output' field of your audio source and it'll take the values from the mixer.



7.2. Graphics manager

Most of the graphics settings are managed via events. For example, you can call **QualityManager.VsyncSet(0);** in order to turn vsync off. If you're using one of the demo scene, you don't have to do extra stuff because everything is ready to use out of the box.

Resolution selector items are dynamically generated from **Quality Manager**, you just have to assign the selector to the 'Resolution Selector' field on Quality Manager, and add a new 'Click Event' which is 'SetResolution'.

8. Contact & Licence

You can contact me or get the latest updates via:

Discord

E-mail

Website

YouTube

If you have any problems, questions, suggestions or feedback, please feel free to contact me.

Licence