

A big **thank you** for purchasing



# Modular Game UI Kit

A multi-purpose UI system

I hope you find this pack useful to create a great game!

If you have any questions or need any help,  
please do not hesitate to reach me at [ricimi.com](http://ricimi.com).

The **Modular Game UI Kit** can only be used under the terms of the  
Asset Store License Agreement, which you can find [here](#).



# Table of contents

- 1. Terms of use & copyright**
- 2. What is Modular Game UI Kit?**

### **2.1 Unity version**

2.2 What is included?

- 3. Asset structure**

- 4. About the demo**

4.1 Canvas

4.2 Reference resolution

### **4.3 Inspector settings**

- 5. UI design language**

5.1 Color palette

5.2 Buttons

5.3 Buttons with icons

5.4 Backgrounds

5.5 Layout groups

- 6. UI components**

6.1 SceneTransition

6.2 Popup

6.3 PopupOpener

6.4 ModularPopup

6.5 ModularPopupOpener

6.6 TabMenu

6.7 SelectionSlider

6.8 TextSelectionSlider

6.9 SliderAmountText

6.10 SliderColorSwap

6.11 CircularProgressBar

6.12 Tooltip

6.13 Gradient

6.14 Useful tips

- 7. Contact**

## 1. Terms of use & copyright

Modular Game UI Kit can only be used under the terms of the Asset Store License Agreement, which you can find [here](#).

The copyright of Modular Game UI Kit and all of its contents belongs to ©ricimi.

After purchasing Modular Game UI Kit, you have the right to use it only for the purposes of developing and publishing a game.



**You are NOT allowed to redistribute or resale the Modular Game UI Kit or any of its contents for any purpose.**

**To redistribute or resale this product is NOT permitted under any circumstances.**

## 2. What is Modular Game UI Kit?

Modular Game UI Kit is a [premium multi-purpose game UI system](#) that contains everything you need to easily build professional, high-quality game user interfaces in Unity.

All the elements included in the asset have been carefully designed, professionally organized and are fully customizable. The complete C# source code is also available to you.

The asset contains a [complete demo](#) that you can use as a starting point for your own games.

### 2.1 Unity version

The project makes extensive use of the latest advancements in Unity's prefab system and therefore requires **Unity 2022.3.0 LTS** or higher.



Trying to use a Unity version lower than 2022.3.0 LTS will not work and will result in broken prefabs in your project.

## 2.2 What is included?

Modular Game UI Kit includes the following content:

- Professional UI design layouts for the following categories: character, character creation, clans, crafting, daily bonus, friends, home, in-game, inventory, login, membership, messages, notifications, profile, quests, ranking, rewards, settings, shop, coins & gems shop, skills, stage selection.
- UI components: buttons, toggle, switch, dropdown, input field, slider.
- Additional UI components: scene transition, popup, popup opener, modular popup, modular popup opener, tab menu, selection slider, text selection slider, slider amount text, slider color swap, circular progress bar, tooltip, gradient.
- Images & icon symbols: 2 background Images (1920px x 1080px), 1 character image (800px), 4 avatar images (512px x 512px), 4 building images (800px x 800px), 77 colored image icons (512px x 512px), 319 white icon symbols (256px x 256px).
- Animations: popups, effects, loading spinner.
- 1 handcrafted color palette.
- Complete C# scripts.

### 2.2.1 What is not included?

The 3D models used for the demo images are not included in the asset. They might be sold separately in the future.

### 3. Asset structure

- **Common**

Contains all the elements of the asset that are generally useful outside the context of the demo.

- **Common/Animations**

Contains all the common animations.

- **Common/Editor**

Contains the color palette.

- **Common/Fonts**

Contains the OPL fonts used in the asset. These are the Regular and SemiBold variants of the Inter font family.

- **Common/Materials**

Contains the “GrayTint” material used for the locked items.

- **Common/Prefabs/1-Foundations**

Contains the main background colors, shapes and fonts.

- **Common/Prefabs/2-Components**

Contains the common UI components like buttons, sliders, etc.

- **Common/Prefabs/3-LAYOUTS**

Contains the common pre-built UI layouts like tab menu, popup template, tooltips, etc.

- **Common/Prefabs/4-Popups**

Contains the common modular popups used throughout the project.

- **Common/Scenes**

Contains the start scene to start the project.

- **Common/Scripts**

Contains the complete C# source code of the asset.

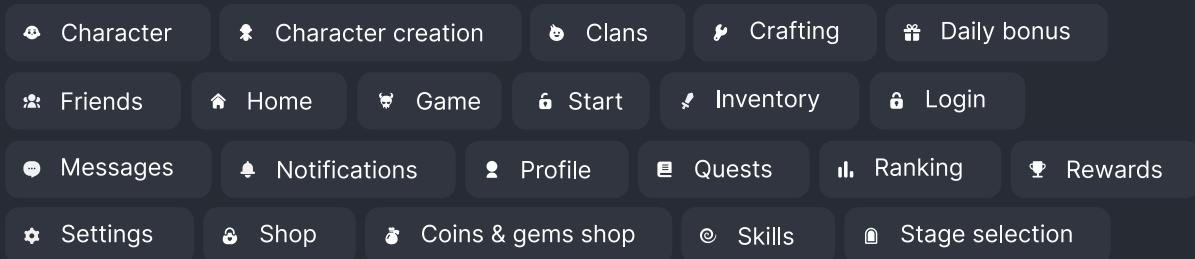
- **Common/Shader**

Contains the “GrayTint” shader used for the locked items.

- **Common/Sprites**

Contains all the images, icons, effects and backgrounds for the demo project.

- **Demo:** Contains all the pre-built, ready to use layouts for the most popular categories found in a game:



Each folder contains its prefabs.

- **Demo/Animations:**

Contains the animations for the demo.

- **Demo/Prefabs:**

Contains the prefabs for the different categories.

- **/Items**

Contains the individual items.

- **/Layouts:**

Contains the individual layout elements.

- **/Lists:**

Contains the lists for the tab menu.

- **/Panel:**

Contains the panel which used as a popup for the demo.

- **/Popups:**

Contains all the popups for the category.

- **Demo/Scenes/Main:**

Contains the used demo scenes in landscape orientation.

- **Demo/Scenes/Test:**

Contains scenes with the panels placed for quick testing in landscape orientation.

- **Demo/Scenes/Main-Portrait:**

Contains the used demo scenes in landscape orientation.

- **Demo/Scenes/Test-Portrait:**

Contains scenes with the panels placed for quick testing in landscape orientation.

## 4. About the demo

The kit contains a complete demo with **full C# source code** that you can use as a starting point for your own game UI.

While the source code is not intended to be a universal framework, it can be a very useful reference when it comes to learning how to approach the implementation of a game UI using Unity's built-in UI system.

Please note this is a graphical UI pack including helpful UI components, scripts, images and animations to use as a starting point for your own games.

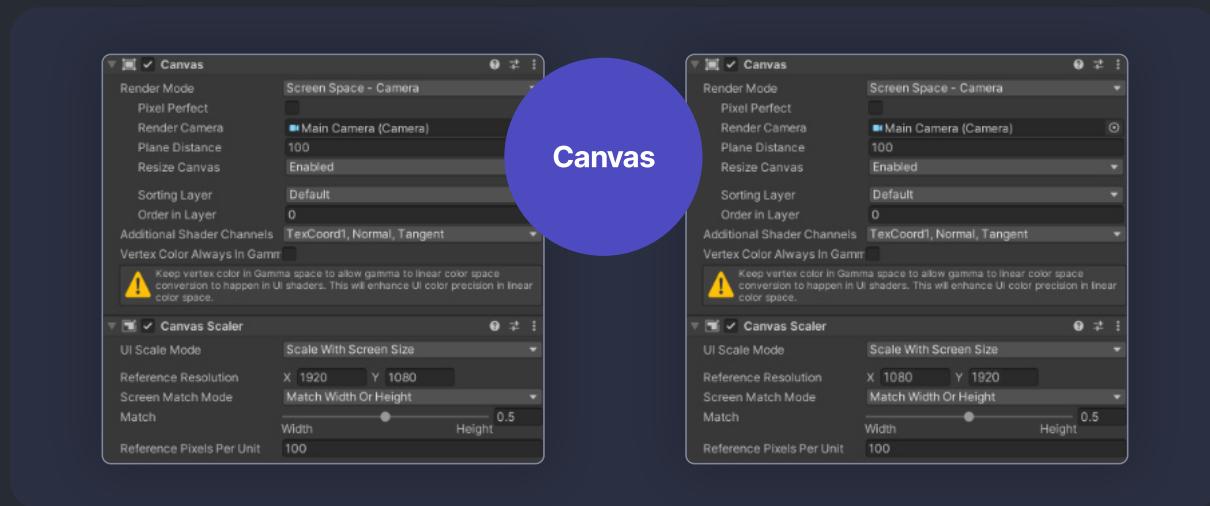
Game-specific features will require additional programming work on your end.

### 4.1 Canvas

All the scenes in the demo project make use of Unity's **Canvas** to display their contents. The render mode of the canvas is set to **Screen Space – Camera** and the **Canvas Scaler** is set to **Scale With Screen Size** scale mode.

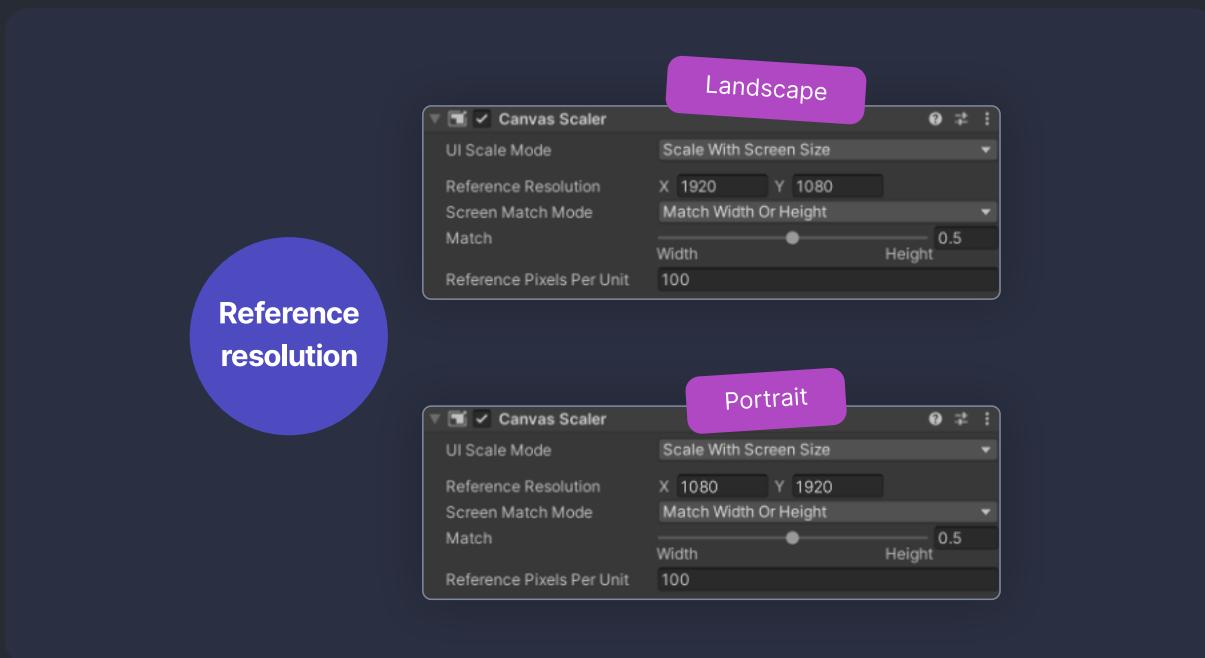
This, together with extensive use of anchors when positioning UI elements, makes it possible to automatically scale the UI across multiple resolutions. This is particularly useful for mobile development, where screen sizes vary wildly between devices. You can find more details about this in the official [Unity documentation](#).

The included demo has been designed for [landscape](#) and [portrait](#) aspect ratios.



## 4.2 Reference resolution

The asset uses a reference resolution of **1920x1280**, which works well across a wide range of aspect ratios.



## 4.3 Inspector settings

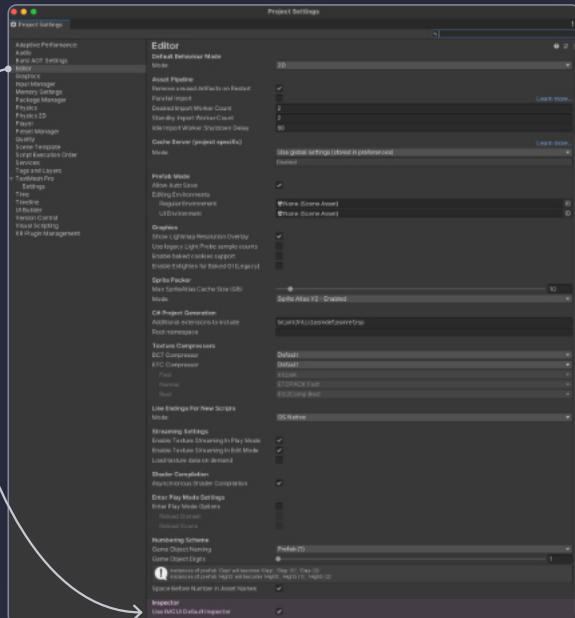


At the time of this writing, **Unity 2022.3.0 LTS** contains a few bugs on the Inspector that seem to be caused by the recent rewrite of the editor code with UI Toolkit.

Until these issues get fixed in a future release of the engine, we highly recommend reverting to the IMGUI-based Inspector so that your experience with the asset is optimal.



Please make sure to enable the “**Use IMGUI Default Inspector**” option in **Edit/Project Settings/Editor**.



Space Before Number in Asset Names

Inspector

Use IMGUI Default Inspector



## 5. UI design language

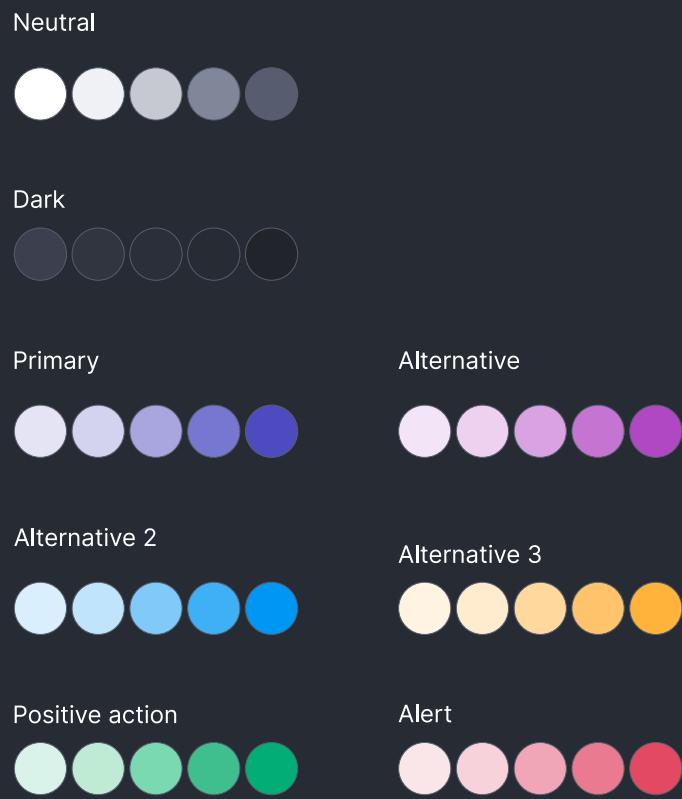
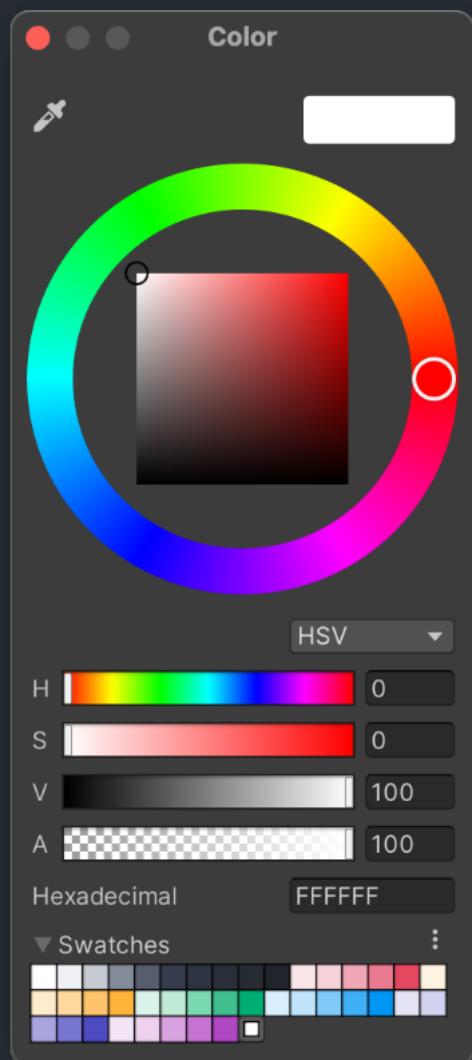
The kit uses a unique **UI design language** that is fully customizable.

You can change the background/button shapes and their colors very easily and your changes will be applied to the whole project.

### 5.1 Color palette

The kit provides a handmade, carefully crafted **color palette** with 40 colors that combine among themselves in a harmonious way.

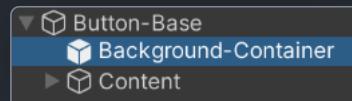
This makes it very easy for you to choose from a wide gamut of chromatic options that are known to work well out of the box when changing the look and feel of your game.



## 5.2 Buttons

The buttons are sorted in primary, alternative and secondary buttons.

-  You can change the **button colors** inside the **base button** folder:  
Common/Prefabs/2-Components/Base-Button-With-Icon/.
-  You can change the shape of the buttons (including IconButton) changing  
the Button-Base prefab's **Background-Container**.



**Primary:** The most commonly used type of button.



**Alternative:** An alternative primary button that uses a different color.



**Alternative 2:** An alternative primary button that uses a different color.



**Positive:** A button for positive actions (e.g., "Accept", "Claim reward", etc.).



**Secondary:** A button used for actions that might be important but not as much as the *primary* action.



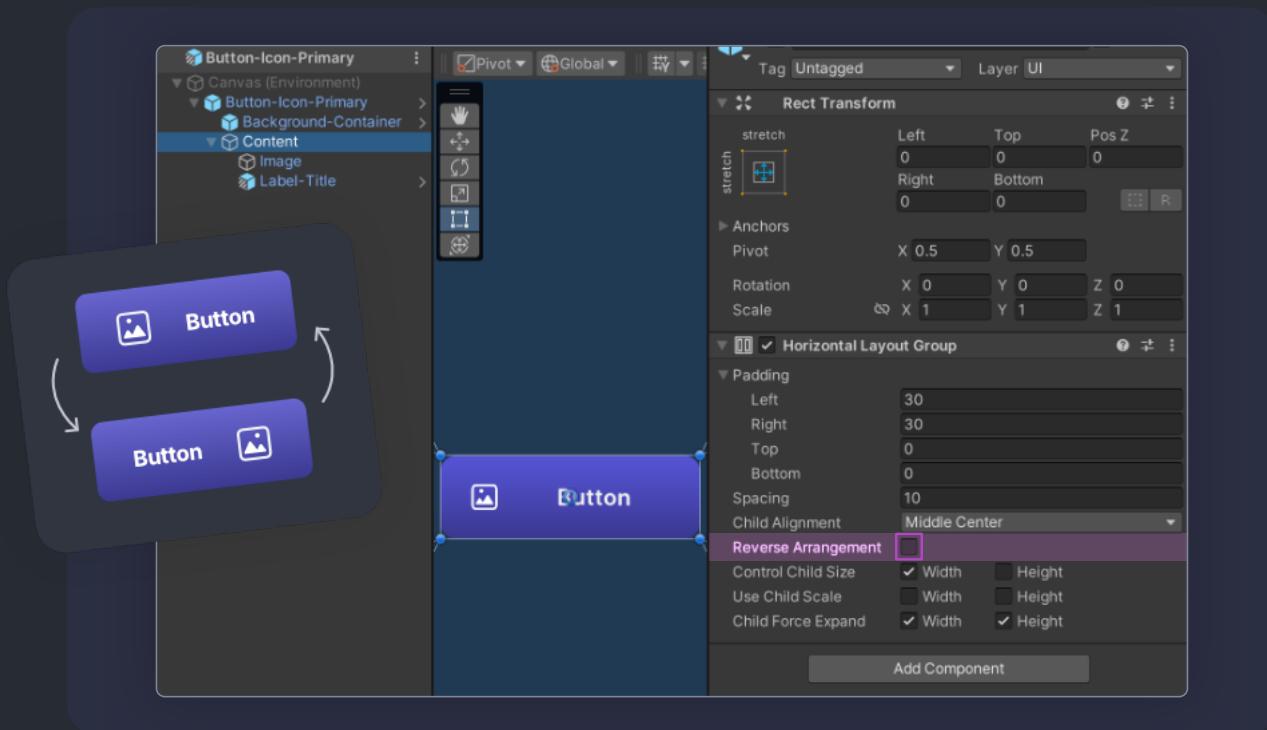
**Alert:** Useful to grab the attention of the user to an important matter.

Prefab variant of the *secondary* button.

## 5.3 Buttons with icons

The buttons use a layout group to allow easily changing the positions of their content (such as icons).

For example, if you want to change the icon position from left to right, just toggle the **Reverse Arrangement** checkbox.



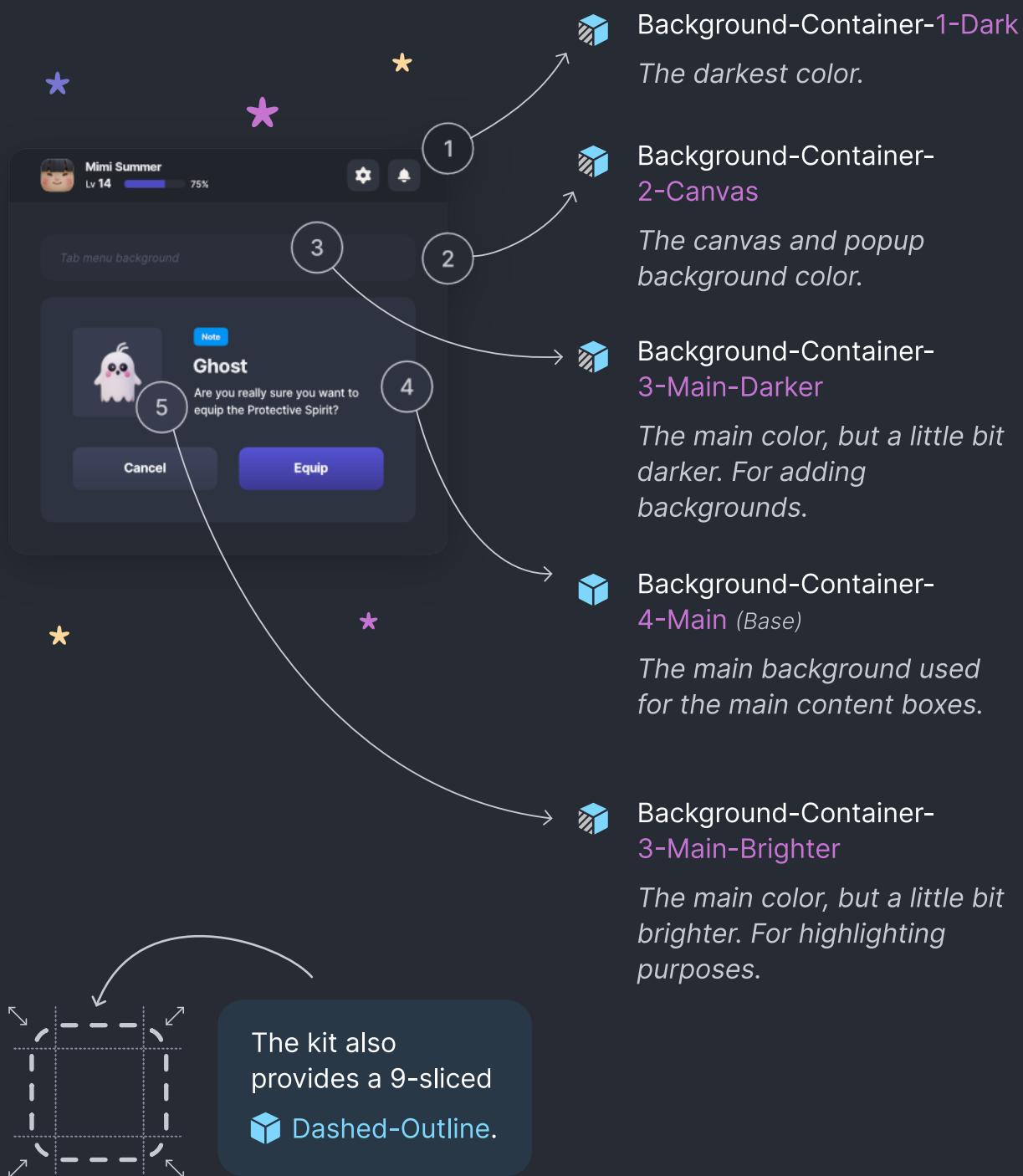
## 5.4 Backgrounds

The background elements are ordered by numbers. They are repeated throughout the project via prefabs.

You can change the background colors or the shapes of the boxes for the whole project by editing these prefabs.



You can change the **background colors** inside the folder:  
Common/Prefabs/1-Foundations.

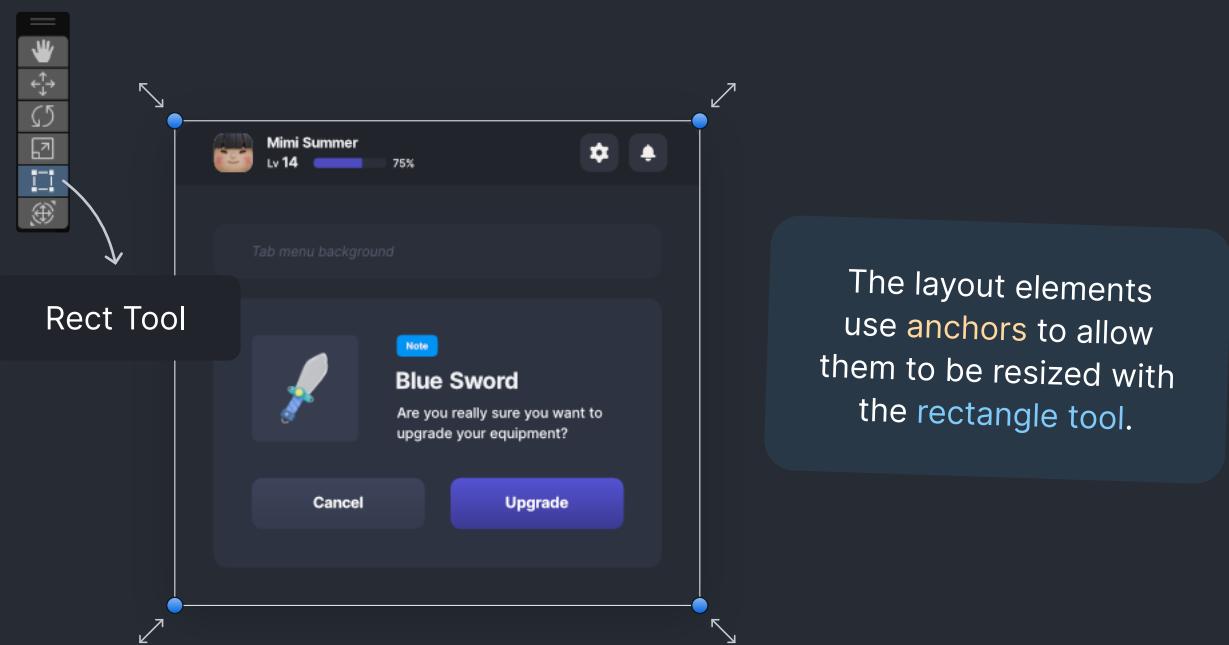
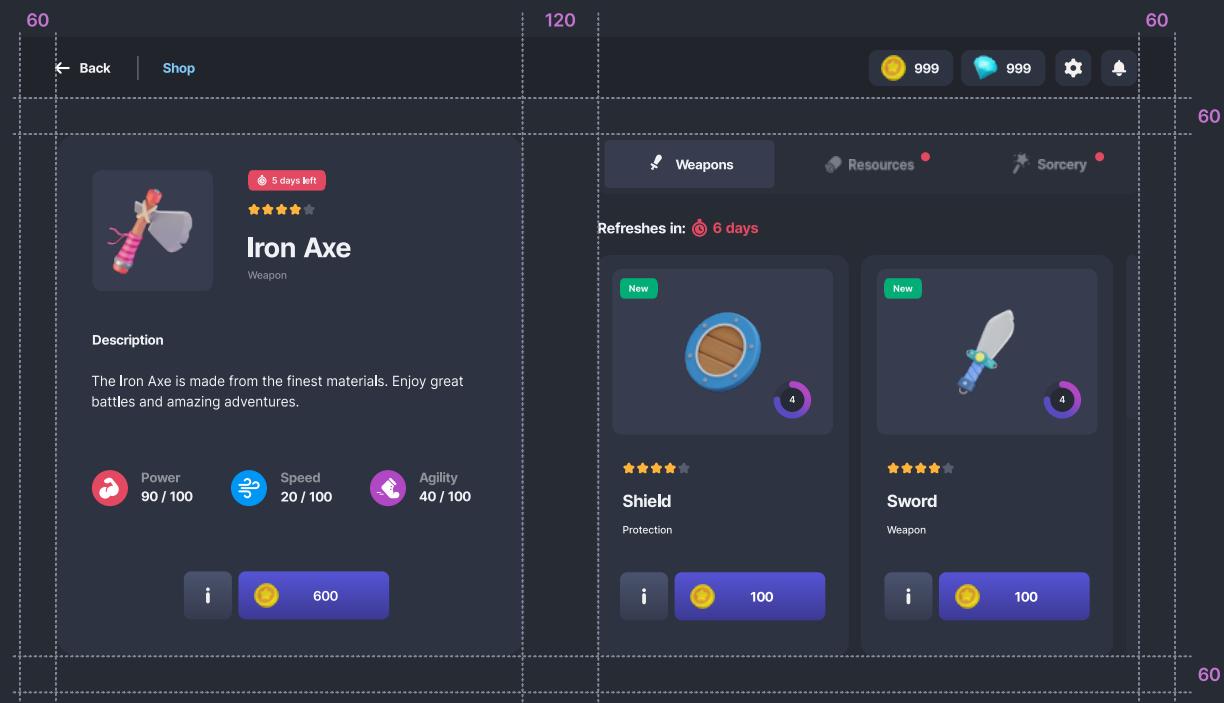


## 5.5 Layout groups

For the **landscape** orientation, the demo is using a **horizontal layout group** for the panel layouts.

For the **portrait** orientation, the demo is using a **vertical layout group** for the panel layouts.

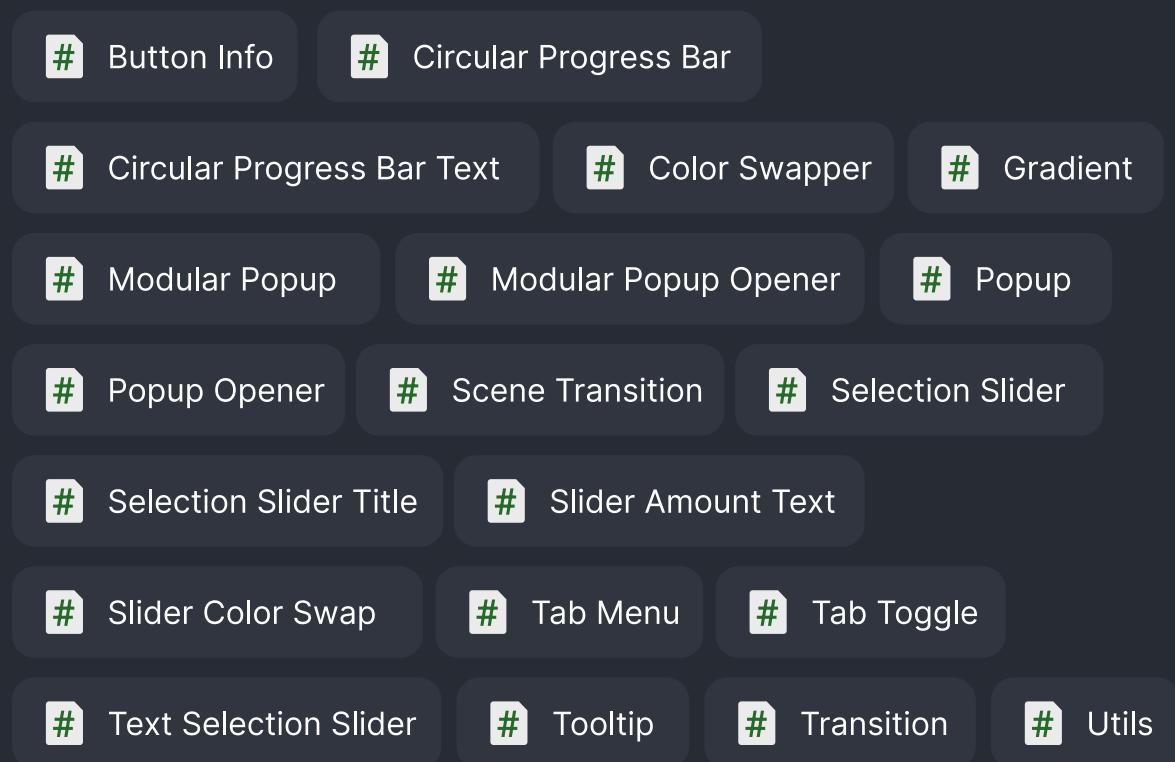
The portrait variations are named **X-Portrait** (where X is the prefab name).



## 6. UI components

The demo makes extensive use of **Unity's built-in UI features**, but also provides some useful extensions added on top of them.

**The pack provides the following components:**

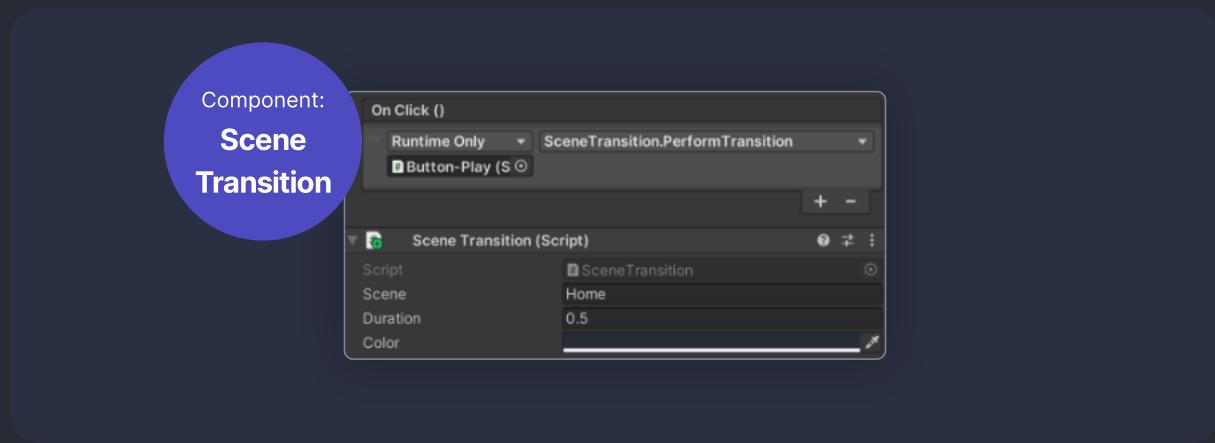


## 6.1 SceneTransition

This component provides functionality to **transition from one scene to another**. Using it is very simple.

You can choose the **background color** as well as the **duration time**. Consider the following example, where we have a “Play now” button in the “Start” scene that should transition to the “Home” scene when clicked.

To do this, we only need to add a **SceneTransition** component to this button game object.



The **SceneTransition** component carries out the logic needed to smoothly fade out from the current scene into the new one. You can specify the **destination scene name** and the **duration** and **color** of the transition.

Note how the very same button calls the **SceneTransition's PerformTransition** method in order to start the transition when clicked.

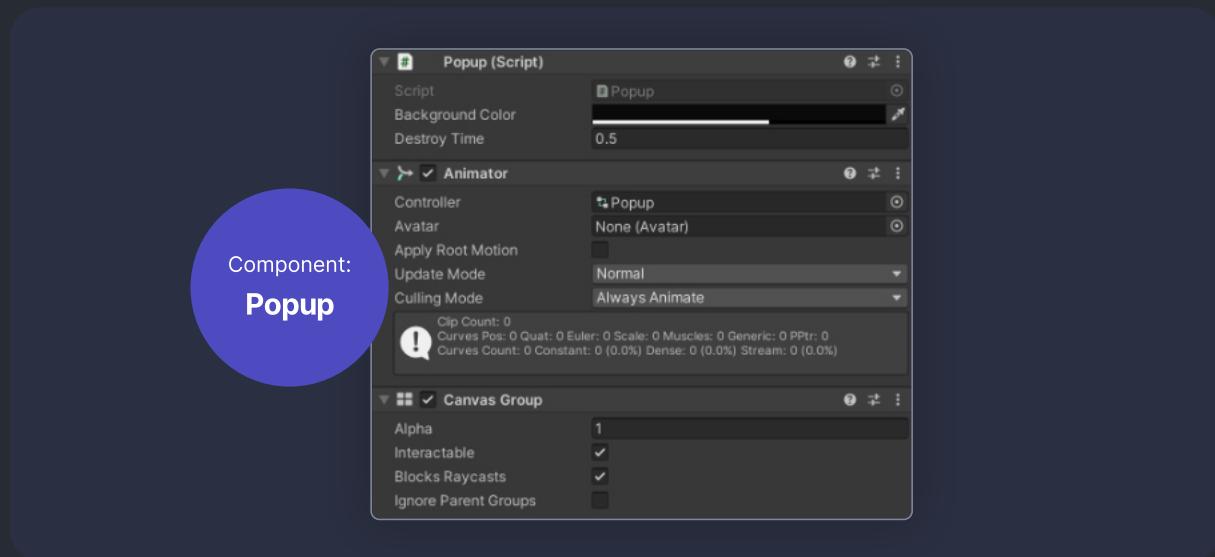
You can make this call in code, too.

 The scenes you want to open must be added to the **Build Settings**.

## 6.2 Popup

This component provides functionality to **open a popup** and darkening the background behind it.

Using it is, again, very simple. We just need to add a **Popup** component to the popup prefab we want to open.



You can choose the **background color** as well as the **destroy time**. The destroy time is the time for the popup game object to be destroyed (in seconds).

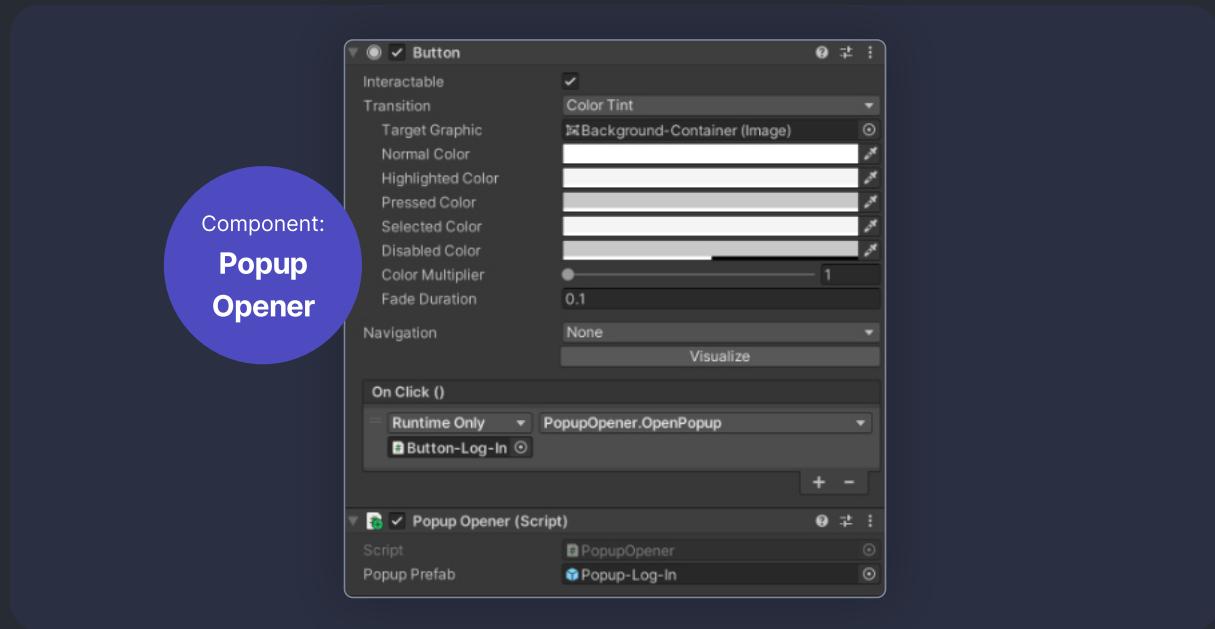
This is useful if you have a closing animation.

We also need to add the **Animator** component for the popup animation.

Because the popup animation uses an alpha transition, we need to add the **Canvas Group** component as well.

## 6.3 PopupOpener

To open a popup, we need to add a **PopupOpener** component to a game object (generally a button). The PopupOpener component carries out the logic needed to open a popup in the current scene by invoking its OpenPopup method.

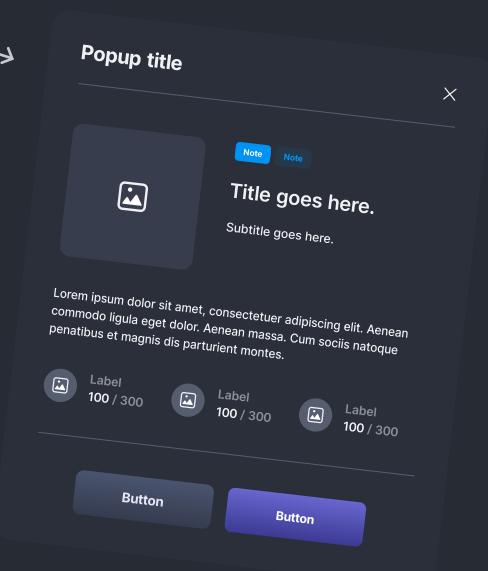


 The asset includes a popup template to help you quickly create a new popup with some basic contents.

You can find it in the folder: **Common/Prefabs/3-LAYOUTS/Popup:**

 **Popup-Template--Unpack-Prefab**

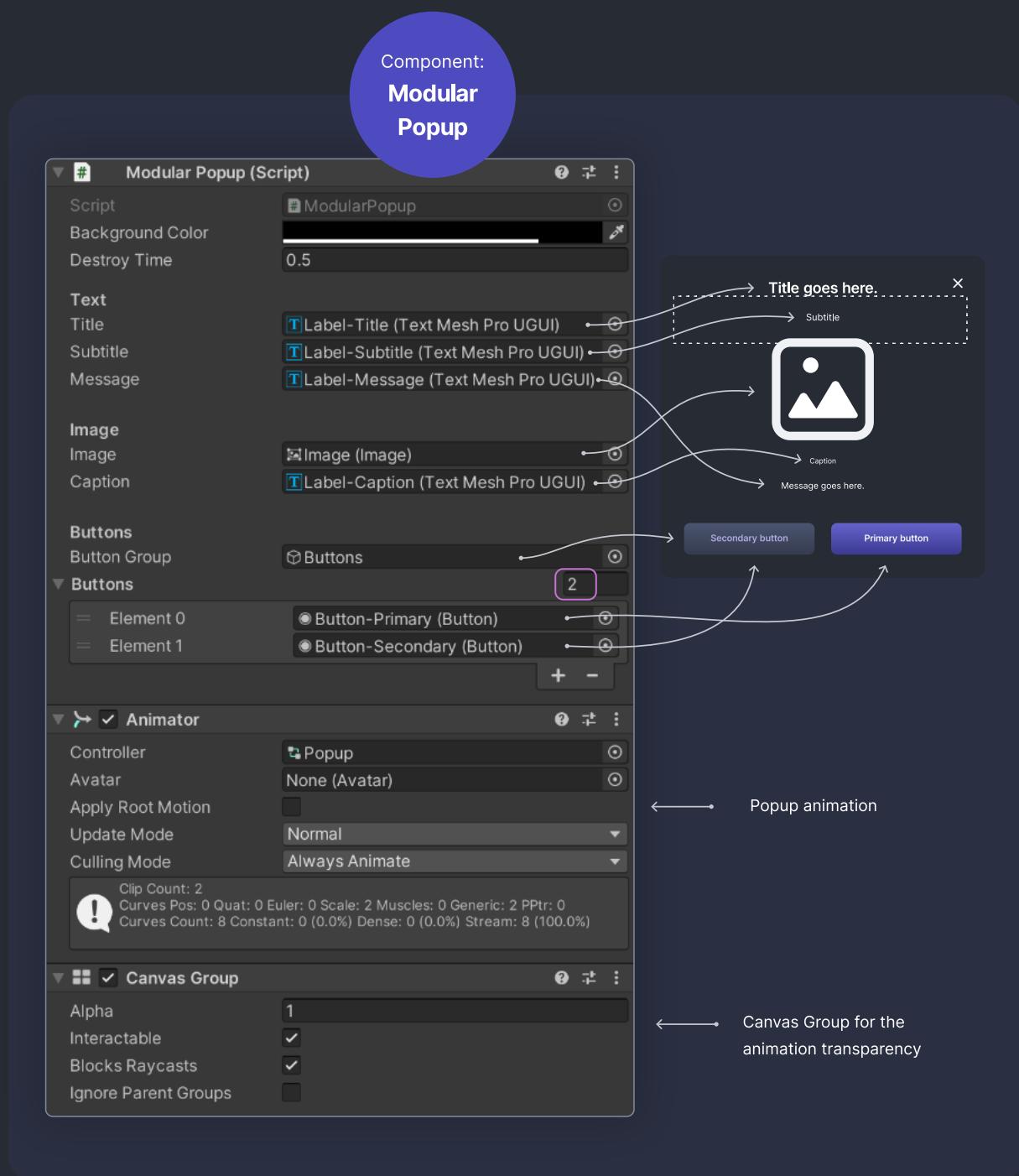
Just drag it to your canvas and  
unpack it with right click: **Prefab >**  
**Unpack**



## 6.4 ModularPopup

The **ModularPopup** component is a generic and modular type of popup that is used across the whole demo to represent many different types of popups in a cohesive way.

This popup is structured in such a way that it can be used with or without text, with or without an images, and with or without any number of buttons, all while automatically adjusting the final layout to always be correct.



## 6.5 ModularPopupOpener

The **ModularPopupOpener** component is used to open a ModularPopup. Every property is optional, so you can leave the ones you do not need empty and the popup prefab will automatically adjust its layout as appropriate.

This component has the following properties:

- **Popup Prefab:**  
The popup prefab to open.
- **Title:**  
The title text.
- **Subtitle:**  
The subtitle text.
- **Message:**  
The body text.
- **Image:**  
The image to show.
- **Tint color:**  
The tint color for the image.
- **Caption:**  
The image caption.
- **Buttons:**  
The buttons to show.  
Every button has an optional label and a click callback.
- **Close Popup When Clicked:**  
If enabled, the popup will be automatically closed upon clicking this button.
- **Ignore Button Clicked Event:**  
This should be generally left disabled. It should only be enabled when the modular popup prefab that you want to open already defines its own 'on clicked' events for its buttons, and you do not want to override them.

**Component: Modular Popup Opener**

**Button**

Interactable: Color Tint  
 Transition: Background-Container (Image)  
 Normal Color:   
 Highlighted Color:   
 Pressed Color:   
 Selected Color:   
 Disabled Color:   
 Color Multiplier: 1  
 Fade Duration: 0.1  
 Navigation: Automatic  
 On Click ():  
 = Runtime Only ModularPopupOpener.OpenPopup  
 Ingame-Inventory-1

**Modular Popup Opener (Script)**

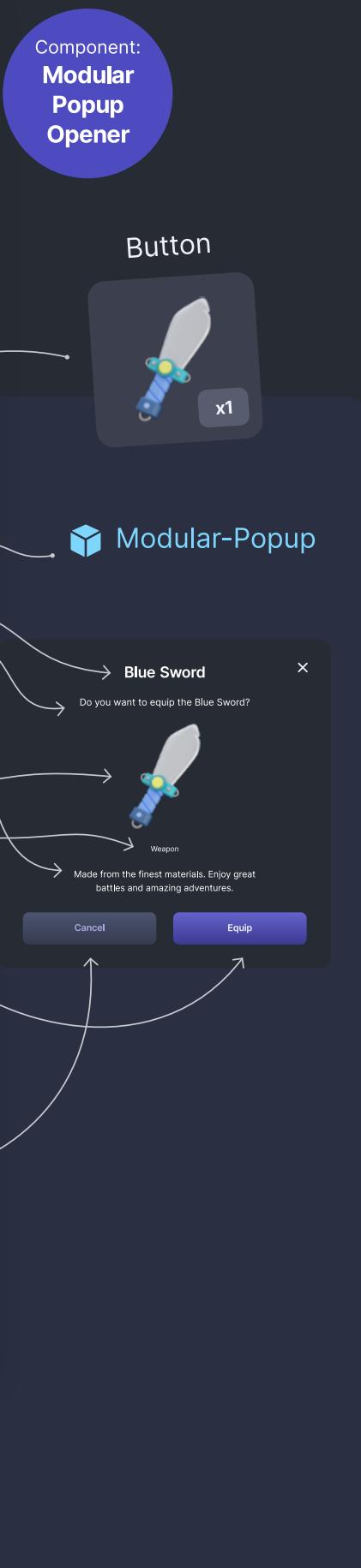
Script: ModularPopupOpener  
 Popup Prefab: Popup-Modular

**Text**: Blue Sword  
**Title**: Do you want to equip the Blue Sword?  
**Subtitle**: Made from the finest materials. Enjoy great battles and amazing adventures.

**Image**: Sword  
**Tint Color**:   
**Caption**: Weapon

**Buttons**: 2

- Equip**: Label: Equip  
 Close Popup When Clicked  
 Ignore Button Clicked Event  
 On Clicked Event ():  
 List is Empty
- Cancel**: Label: Cancel  
 Close Popup When Clicked  
 Ignore Button Clicked Event  
 On Clicked Event ():  
 List is Empty

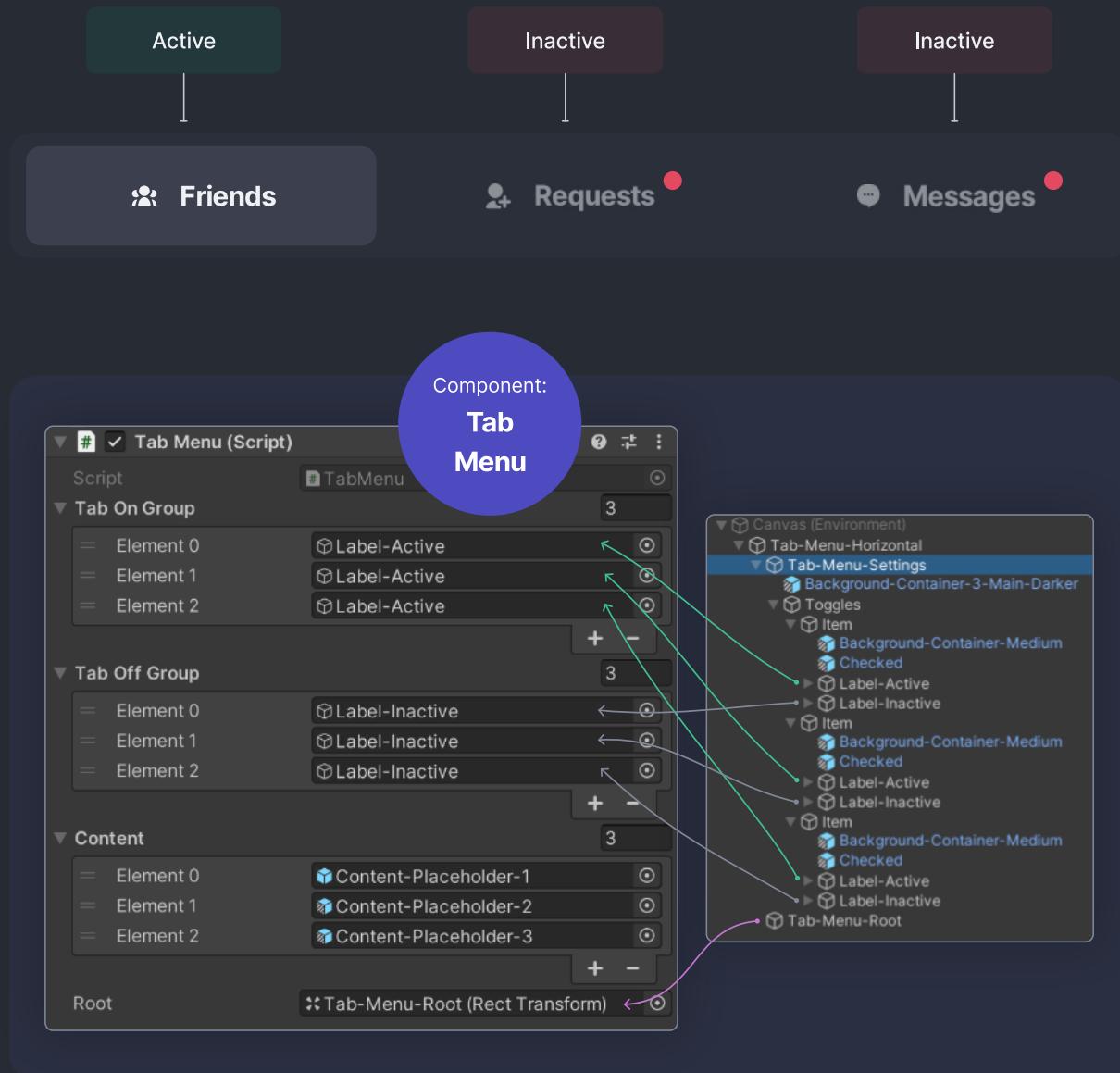


## 6.6 TabMenu

The **TabMenu** component provides a tabbed menu that allows you to switch between different, independent tabs of content.

This component has the following properties:

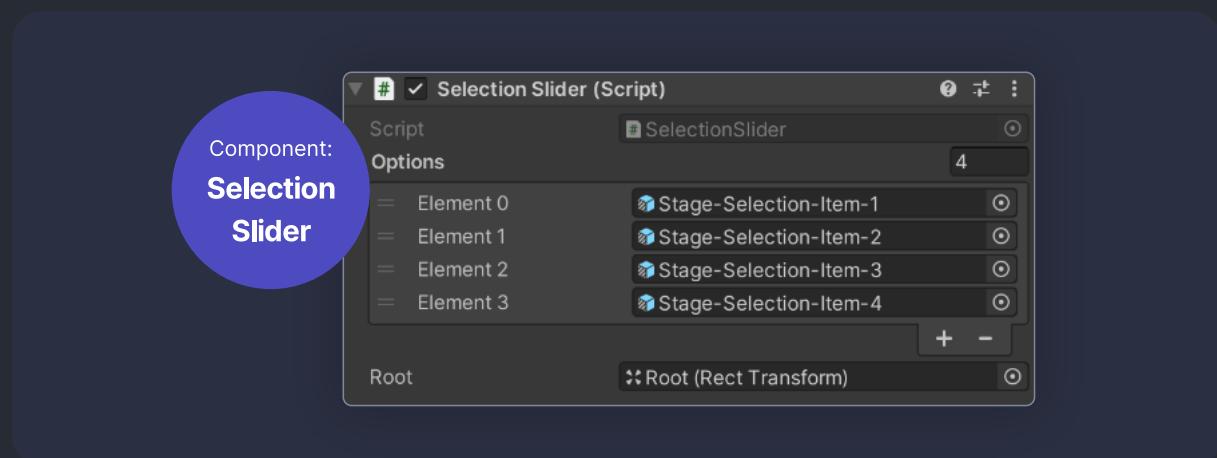
- **Tab On Group:** The active label game objects.
- **Tab Off Group:** The inactive label game objects.
- **Content:** The prefabs used as the content in every tab.
- **Root:** The (optional) parent game object to which to attach the content of every tab.



## 6.7 SelectionSlider

The **SelectionSlider** component allows you to scroll through different prefabs using buttons.

You can make use of the **Root** using a **GameObject** to load the prefabs in specific positions or sizes.

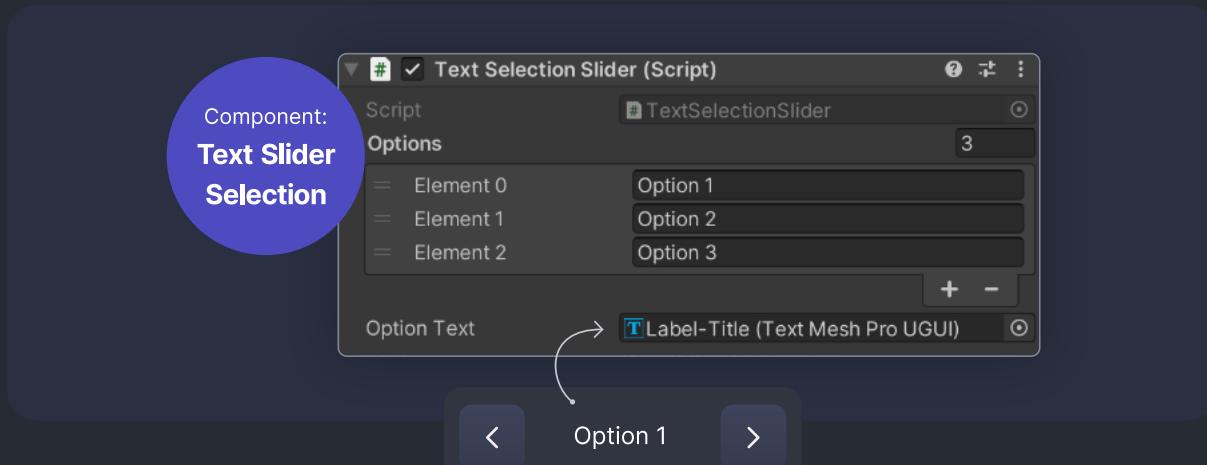


## 6.8 TextSelectionSlider

The `TextSelectionSlider` component is a specialized selection slider that allows you to scroll between different text-based options.

This component has the following properties:

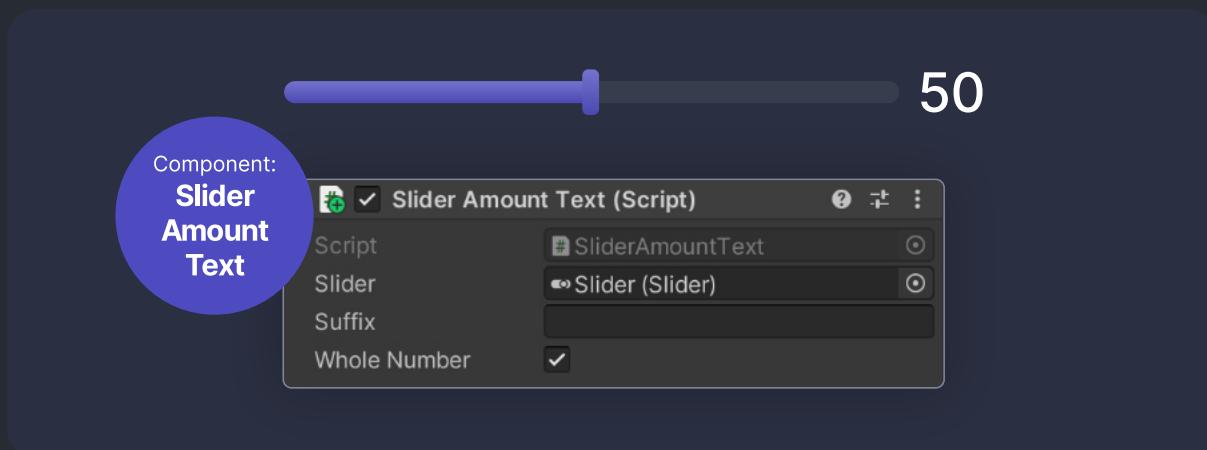
- **Options:** A list containing all the available options. Every option is a string.
- **Option Text:** The associated TextMesh Pro label displaying the currently selected option.



## 6.9 SliderAmountText

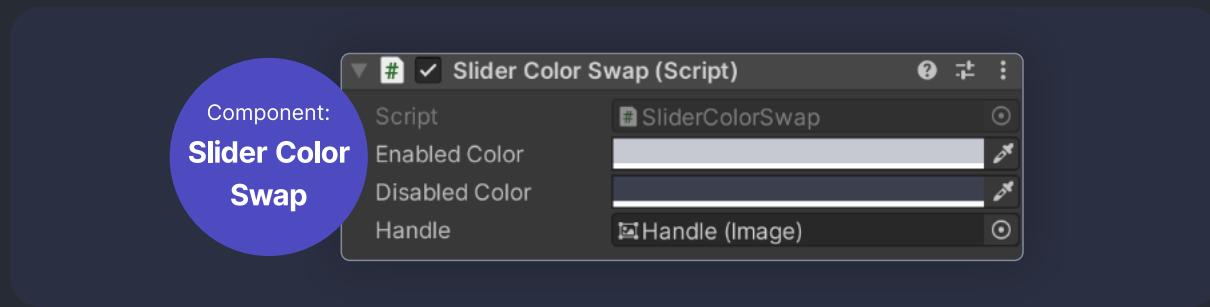
The `SliderAmountText` component allows you to link a text label with the current progress value of a slider.

You can specify the `suffix` to use (if any) and whether the number should be whole or not.

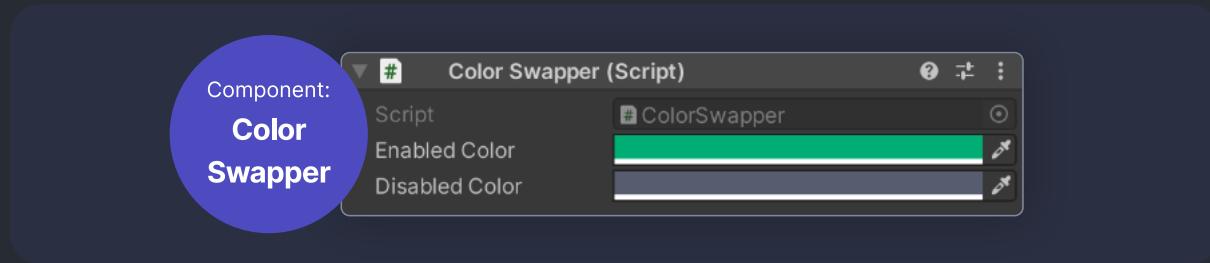


## 6.10 SliderColorSwap

You can easily swap the handle color of any slider using this component.



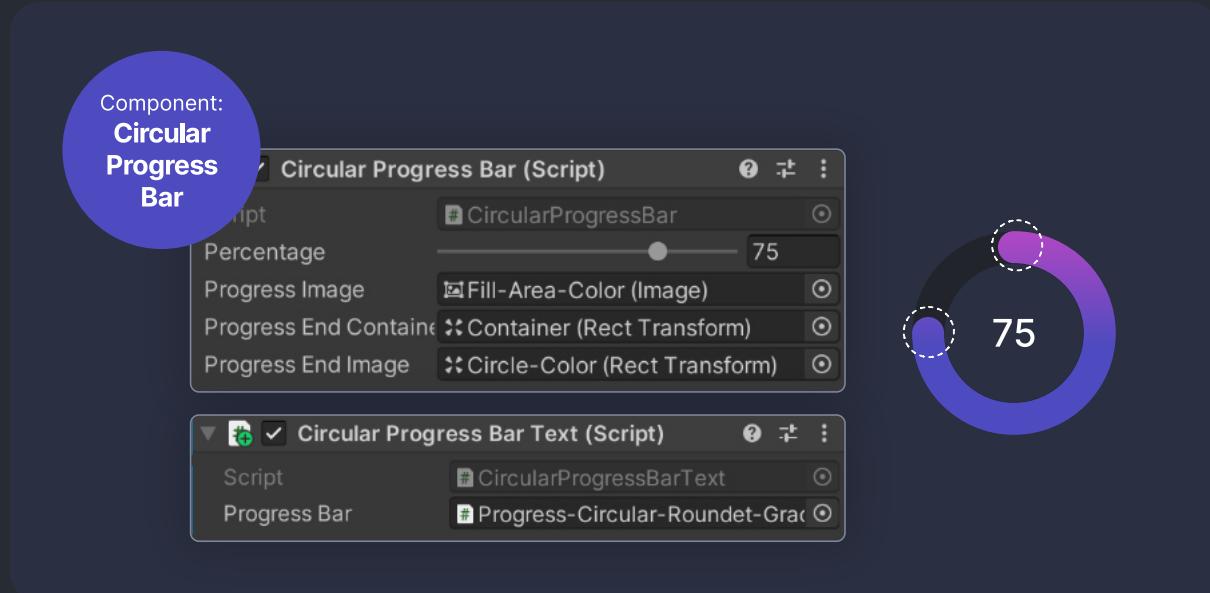
You can easily swap colors using the [ColorSwapper](#) component.



## 6.11 CircularProgressBar

The [CircularProgressBar](#) component provides a progress bar with rounded corners.

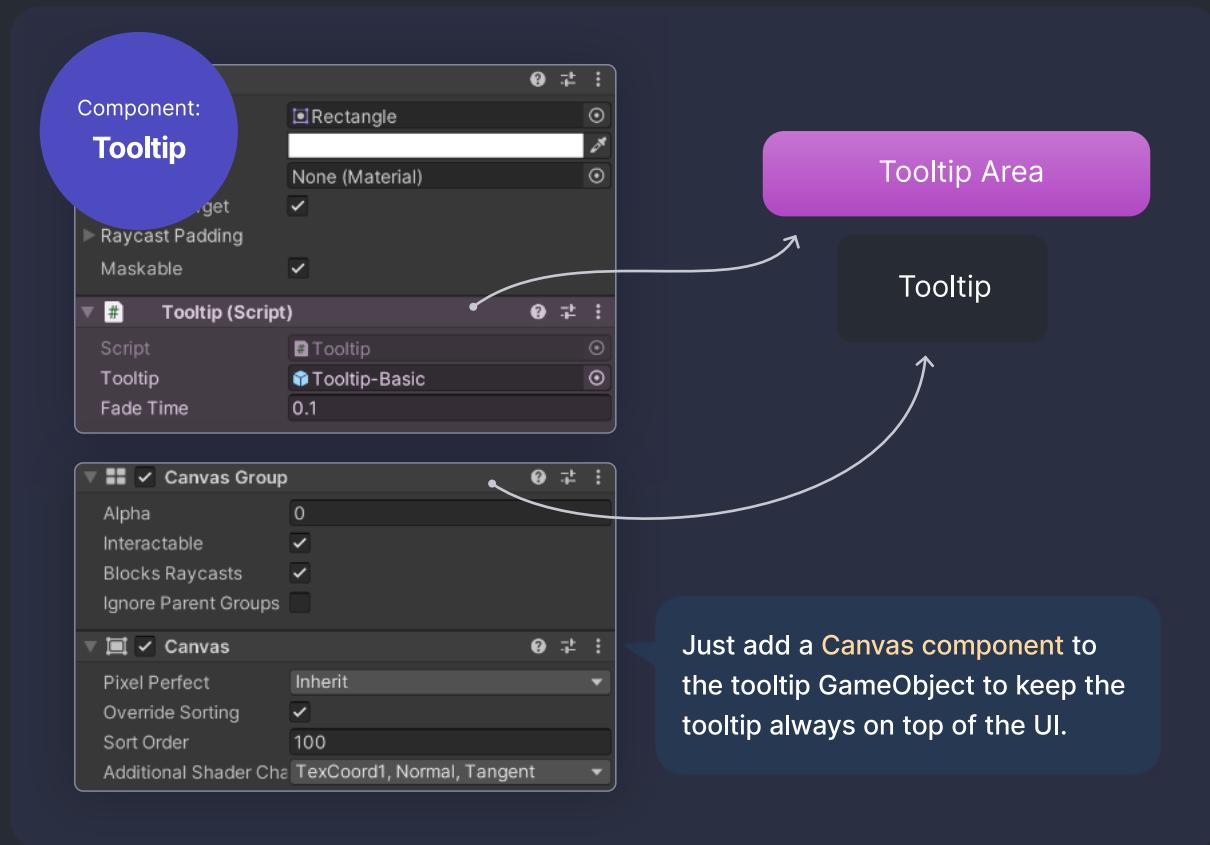
The [CircularProgressBarText](#) component allows you to link a text label with the current progress value of a slider.



## 6.12 Tooltip

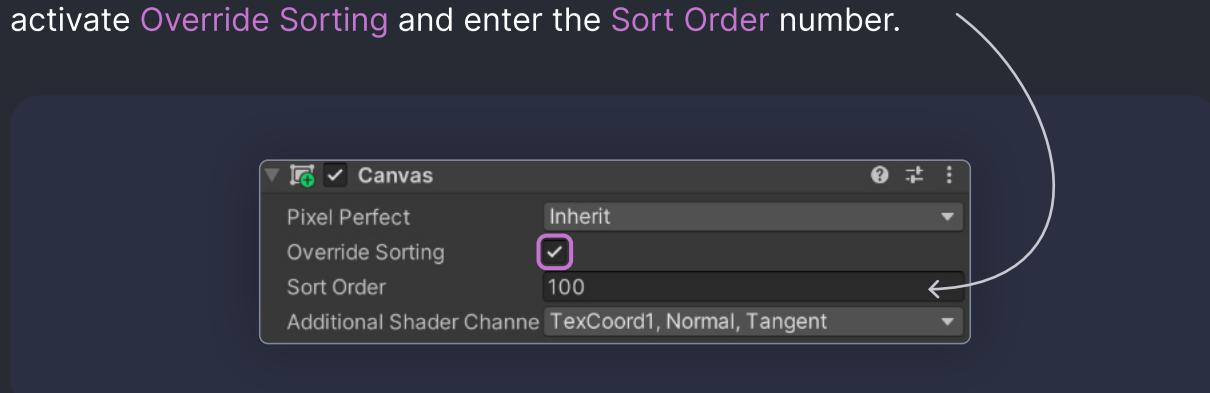
You can easily add a tooltip to any game object using the Tooltip component. You can choose the [fade time](#) (the duration for the fade-in time in seconds).

The tooltip itself needs a [canvas group](#) with the [Alpha](#) set to **0**, like you can see in the image below.



You can find our pre-made tooltip prefabs with a size fitter component (the size of the window fits automatically to its content) inside the folder [Common/Prefs/Prefs/3-LAYOUTS/Tooltips/](#), but you can literally choose any object as a tooltip.

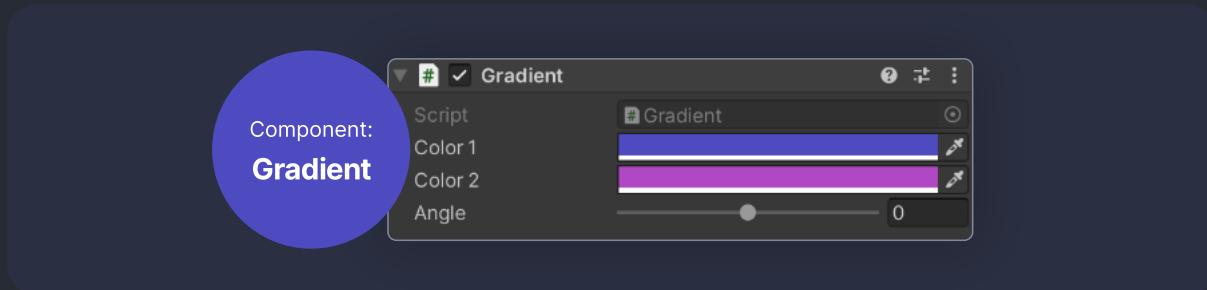
If you need to keep the tooltip on top of the UI, just add a [Canvas](#) component, activate [Override Sorting](#) and enter the [Sort Order](#) number.



## 6.13 Gradient

This is a very nice component, because it adds a modern and special style and also additional color options to your design.

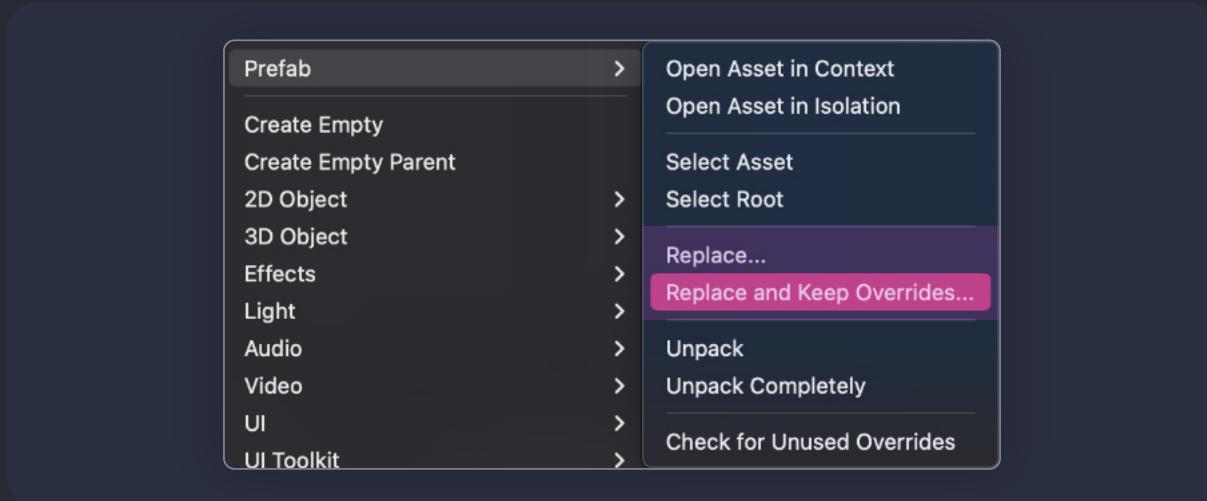
It can be used with any image. Using it with the modular color palette is very easy and quick, especially to try out different color combinations.



## 6.14 Useful tips

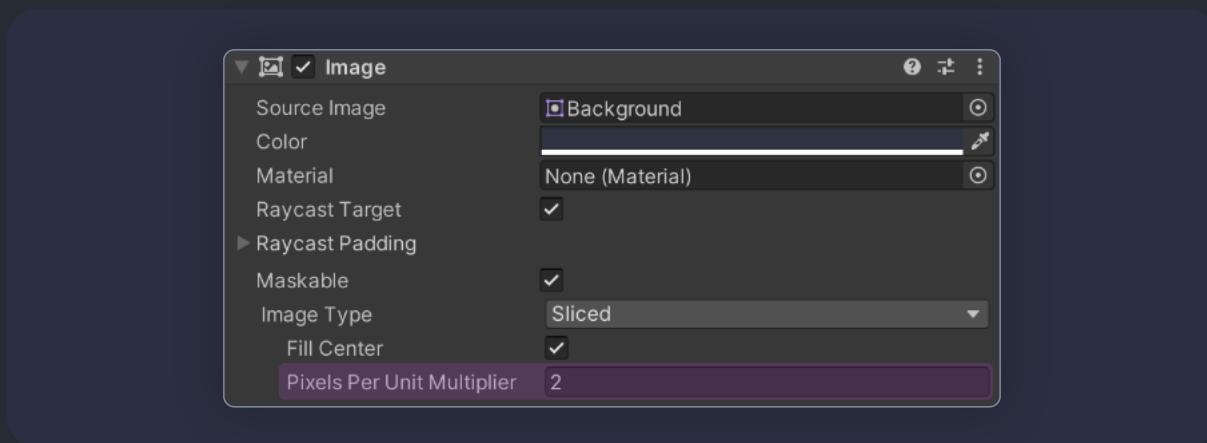
### Replacing Prefabs

You can easily **replace a prefab** by selecting it and right-clicking over it, selecting one of the following options: **Prefab/Replace ...** or **Prefab/Replace and Keep Overrides ...**



### Rounded corner radius

You can easily set the **corner radius** for the buttons or backgrounds by changing the **Pixels per unit Multiplier** amount.



### Changing font color

You can easily change the **font color** by changing the **Label Prefab** located inside the folder: **Common/Prefabs/1-Foundations/Label/**. This will change to font color for the whole project.

## 7. Contact

If you have any questions or need any help, please do not hesitate to reach me at [ricimi.com](http://ricimi.com). Please make sure to include your invoice number.

I am always happy to help. 😊



♥  
Thank you



# Modular Game UI Kit

A multi-purpose UI system



Copyright © ricimi - All rights reserved.



@ricimi



@ricimi.art