A big **thank you** for purchasing my

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.

The Cute Kawaii GUI Pack can only be used under the terms of the **Asset Store License Agreement**, which you can find here.

Table of contents

Copyright & terms of use

Cute Kawaii GUI Pack can only be used under the terms of the **Asset Store License Agreement**, which you can find here.

The copyright of Cute Kawaii GUI Pack and all of its contents belongs to ©ricimi.

After purchasing Cute Kawaii GUI Pack, you have the right to use it only for the purposes of developing and publishing a game.

You are NOT allowed to redistribute or resell the Cute Kawaii GUI Pack or any of its contents for any purpose.

You are NOT allowed to use this product or any of its contents as part of the training data of an AI.

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

What is Cute Kawaii GUI Pack?

Cute Kawaii GUI Pack is a carefully crafted game UI pack, containing many elements to easily build professional game user interfaces in Unity in an adorable kawaii style.

All the elements included in the asset have been carefully designed, professionally organized and the shapes and colors 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.

Please note that game-specific features will require additional programming work on your end.

Unity version

The demo 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.

What is Kawaii?

In Japanese, the word kawaii means "Cute" and it embodies everything small, innocent and adorable.

Objects and characters are reduced to simple rounded shapes and lines and have cute little faces. The colors consist of soft pastels and rainbows.

Asset structure

Demo/Animations
Contains all the common animations.

Demo/Fonts
Contains the OPL fonts used in the asset.

Demo/Materials
Contains the materials used in the asset for the popup particle effects.

Prefabs/Common/1-Foundations
Contains the main backgrounds, shapes and labels.

Prefabs/Common/2-Components
Contains the common UI components like buttons, sliders, etc.

Common/Common/Prefabs/3-Layouts
Contains the common pre-built UI layouts like top and bottom bar, headlines, etc.

Common/Prefabs/4-Popups
Contains a common confirmation popup used throughout the project.

Demo/Prefabs
Contains all the prefab elements used to build the scenes, sorted by category.

Demo/Scenes
Contains the project scenes.

Demo/Scripts
Contains the complete C# source code of the asset.

Demo/Sprites
Contains all the images, icons, effects and backgrounds for the demo project

Icons
Contains .PNG icons and Avatars in different resolutions.

About the demo project

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

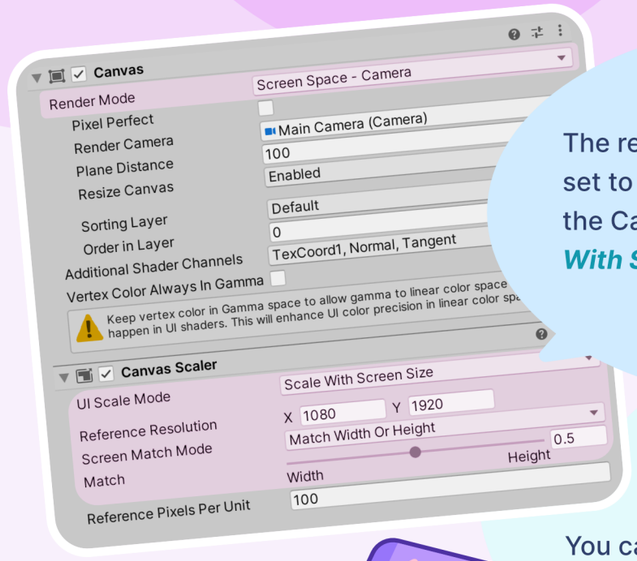
While the source code is not intended to be a universal framework, it can be a very useful reference when it comes to learning how to approach the implementation of a game UI using Unity's built-in UI system.

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.

Canvas

All the scenes in the demo project make use of Unity's Canvas to display their contents. The included demo has been designed for **portrait aspect ratios**.



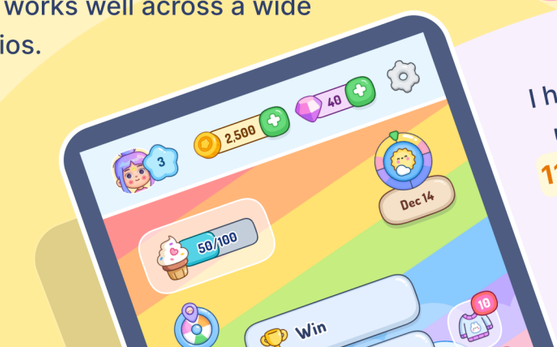
The render mode of the canvas is set to ***Screen Space – Camera*** and the Canvas Scaler is set to ***Scale With Screen Size*** scale mode.

You can find more details about this in the official **Unity documentation**.

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.

Reference resolution

The asset uses a reference resolution of **1920×1280**, which works well across a wide range of aspect ratios.

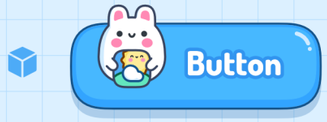


I have tested the demo using a resolution of **1125 × 2436** (iPhone X) and **1080 × 1920** (HD portrait) in the game tab.

Buttons

You can find the Buttons used throughout the project inside folder: *Demo/Prefabs/2-Components/Buttons*.

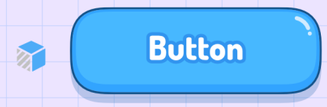
The buttons are sorted in:



Button-Color: These buttons consist of a big sized icon on the left side.

Button-Color-Icon: Button with a size fitted icon. These buttons use a layout group (Content) 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.



Button-Color-Icon-Only: These buttons only consist of a label.

Button-Color-Label-Only: These buttons only consist of an icon.

Shapes

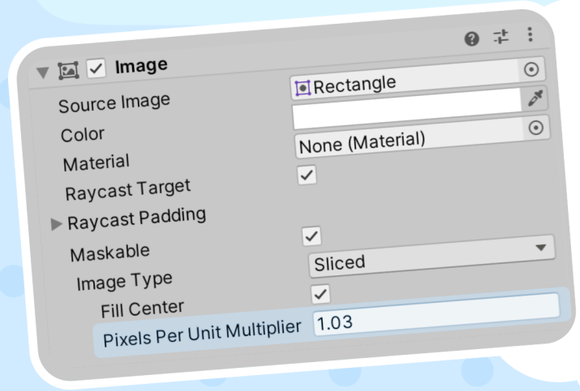
You can find the shapes used throughout the project for the containers inside folder *Demo/Prefabs/Common/1-Foundations/Shapes*. They consist of **different layers** and are therefore **completely customizable**. You can adjust the shape and color, as well as the outline size.

The shapes are sorted in:

Rectangles-Outline:
A simple rectangle only with an outline.

Rectangles-Outline-Shadow:
A rectangle with an outline, depth shading and Shadow.

Rectangle-Outline-Transparency:
A rectangle only with an outline with a transparent background.

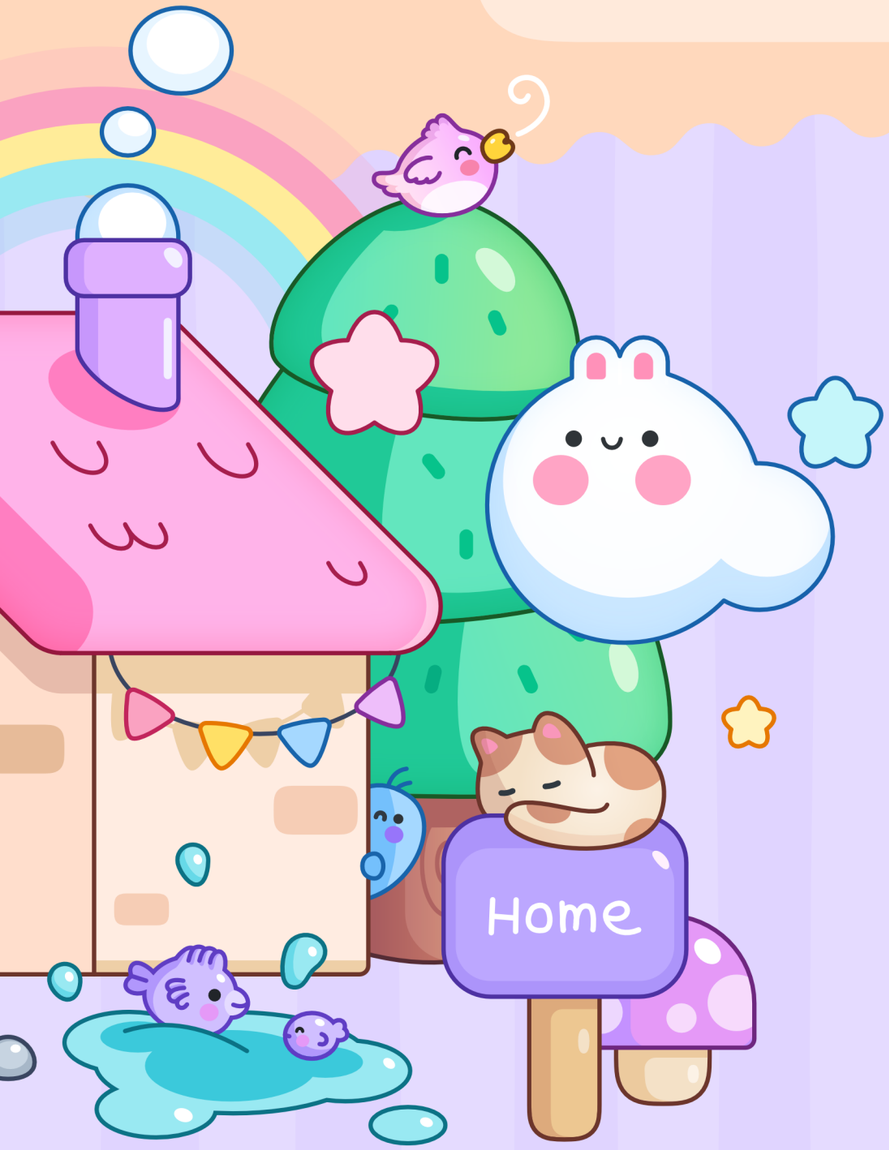


You can adjust the shape by changing the Pixel Per Unit Multiplier inside the Image component setting *(Note that the settings of the interior images are slightly larger)*.

Components

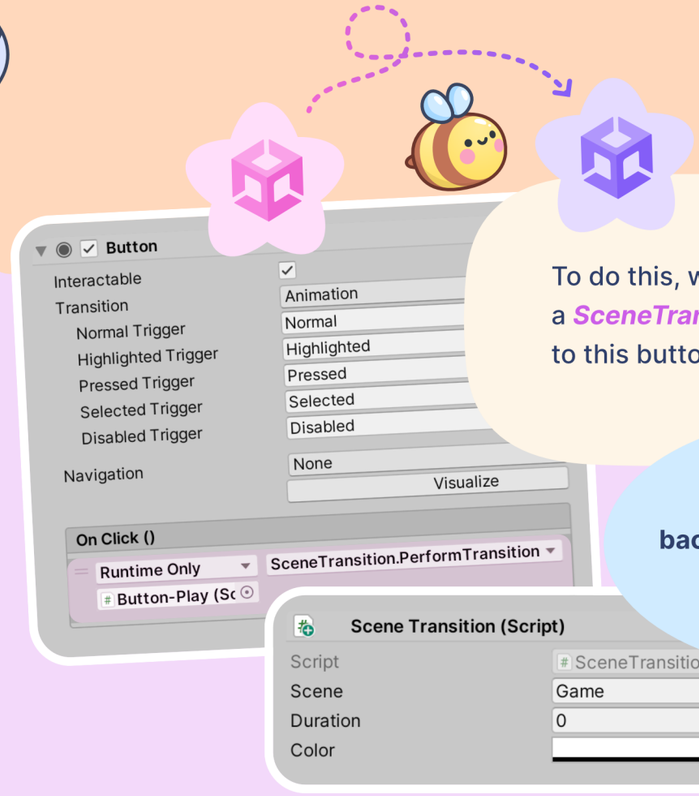
The demo project makes extensive use of Unity built-in UI features, but also provides some useful extensions.

The demo project uses the ***SceneTransition*** component and the ***PopupOpener*** component.



The SceneTransition component

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



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

You can choose the **background color** as well as the **duration time**.

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, duration and color of the transition.

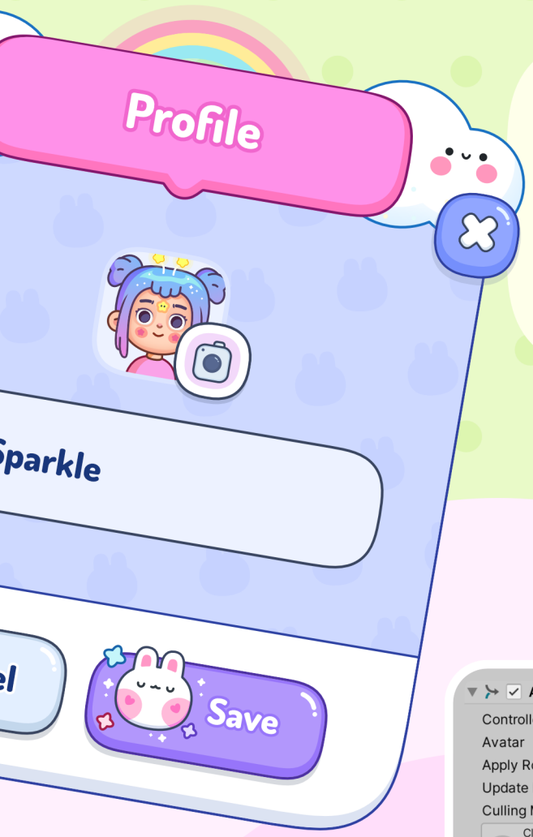
Note how the very same button calls the *SceneTransition's* **PerformTransition** method in order to start the transition when clicked. You can make this call in code, too.

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

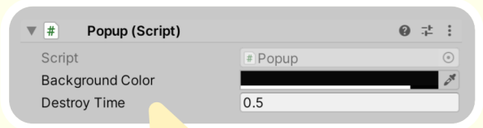
The Popup component

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

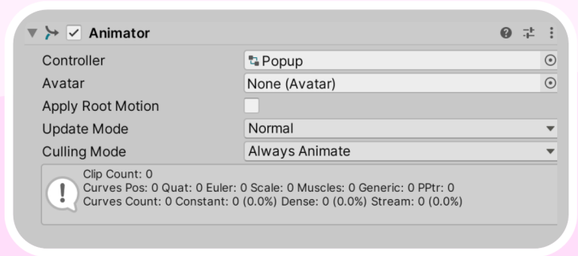
Using it is, again, very simple. We just need to add a *Popup* component to the **popup prefab** we want to open.



You can choose the **background color** as well as the **destroy time**.



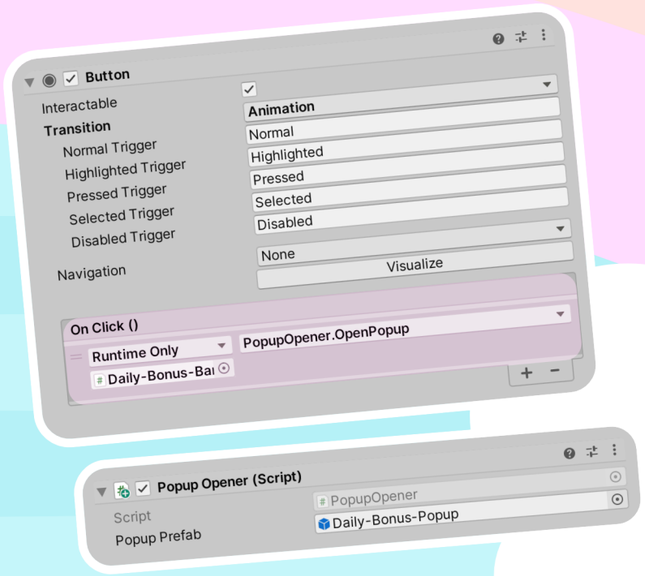
The destroy time is the time for the popup game object to be destroyed (in seconds). This is useful if you have a closing animation.



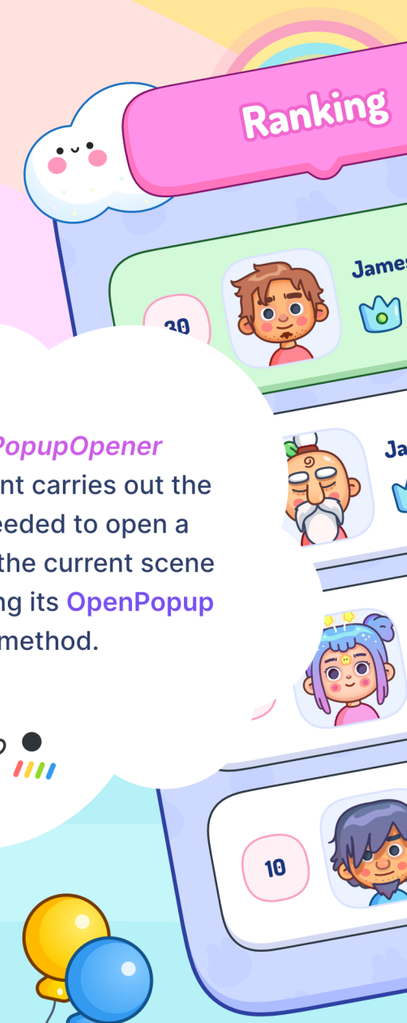
We also need to add the *Animator* component for the **popup animation**.
Because the popup animation uses an **alpha transition**, we need to add the *Canvas Group* component as well.

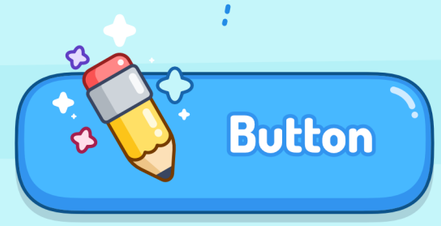
The Popup Opener component

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.





Contact

If you have any questions, please contact me at ricimi.com.

I am always happy to help.

Please make sure to include your **invoice number**.

Thank you

ricimi
ricimi.com