

A big **thank you** for purchasing our



Clean & Minimalist GUI Pack

UI elements, demos and icons for games.



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

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

Please make sure to include your invoice number.

The Clean & Minimalist GUI Pack can only be used under the Unity Asset Store License Agreement which you can find [here](#).



Table of contents

- 1. Copyright & Terms of use**
- 2. What is Clean & Minimalist GUI Pack?**
 - 1.1. Unity Version
- 3. What is included?**
- 4. Asset structure**
- 5. UI Elements**
 - 4.1. Unity's built-in UI system
 - 4.2. Canvas
 - 4.3. Reference resolution
- 6. UI Components**
 - 5.1. Scene Transition
 - 5.2. Popup Opener
 - 5.3. Clean Button
 - 5.4. Tooltip
 - 5.5. Gradient
 - 5.6. Sprite Swapper
 - 5.7. URL Opener
- 7. Contact**

1. Copyright & terms of use

The **Clean & Minimalist GUI Pack** can only be used under the [Unity Asset Store License Agreement](#) which you can find [here](#).

The copyright of the **Clean & Minimalist GUI Pack** and all of its contents belongs to © **ricimi**.

After purchasing the **Clean & Minimalist 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 resale the Clean & Minimalist GUI Pack or any of its contents for any purpose. To distribute or resale this product is NOT permitted under any circumstances and is strictly prohibited.

Thank you for respecting my work.

Documentation

2. What is Clean & Minimalist GUI Pack?

The **Clean & Minimalist GUI Pack** is a customizable, mobile-friendly game UI pack containing a collection of UI elements and demos that can be used to create a complete game with a nice clean and minimalist look.

1.1 Unity Version

While the sprites themselves do not depend on any specific Unity version, the accompanying demo project requires **Unity 2022.3.0** or higher.

3. What is included?

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.

Included are:

- Buttons
- Dropdown
- Scrollbars
- Input Fields
- Modal Windows
- Progress Bars (Animations only)
- Slider
- Switches
- Toggles
- Tooltips
- Demo scenes
- Popup prefabs
- Icons
- Game controller icons
- Background images
- C# scripts
- UI components
- Gradients
- Sound effects
- Animations
- Color Palettes
- .PSD and .SVG Source icons
- .PSD Source Mockups

4. Asset structure

After importing the asset package into your Unity project, you will see all the resources provided live in the **"GUIPack-Clean&Minimalist"** folder.

This folder is further subdivided into the following subfolders:

Demo

Contains all the assets and prefabs of the example demo that makes use of all the sprites included in the pack via Unity's UI system.

Please note the demo and its accompanying source code are only intended as an example showcasing what you can build with this game UI pack. Feel free to use it as a starting point and extend it as you see fit for your own game.

Demo/Animations

Contains all Animations.

Demo/Editor

Contains all color palettes. The **Main Color Palettes** for each theme as well as a **Complete Color Palette** for choosing any advanced colors (very useful).

Demo/Fonts

Contains OPL fonts. The pack uses two Google font families called "Roboto" and "Roboto Condensed".

Demo/Gradients

Contains Gradient Materials. You can use these pre-made gradients or the gradient component to choose your own colors.

Demo/Prefabs

Contains all UI elements as prefabs.

Demo/Resources/Popups

Contains all pre-built popup prefabs in Light and Dark theme.

Demo/Scenes

Demo/ Contains the main "Demo" menu scene.

Themes/ Contains the "Light" and "Dark: theme scenes.

Demo/Scripts

Contains the full C# source code of this pack.

Demo/Shader

Contains the shader used for the gradient materials.

Demo/Sounds

Contains the sound effects used for the demo project.

Demo/Sprites

Contains all shapes and icons used for the demo project.

Sources

Contains the original art source files of all icons as vectorial .PSD and SVG as well as Mockups in .PSD format.

5. UI Elements

4.1. Unity's built-in UI system

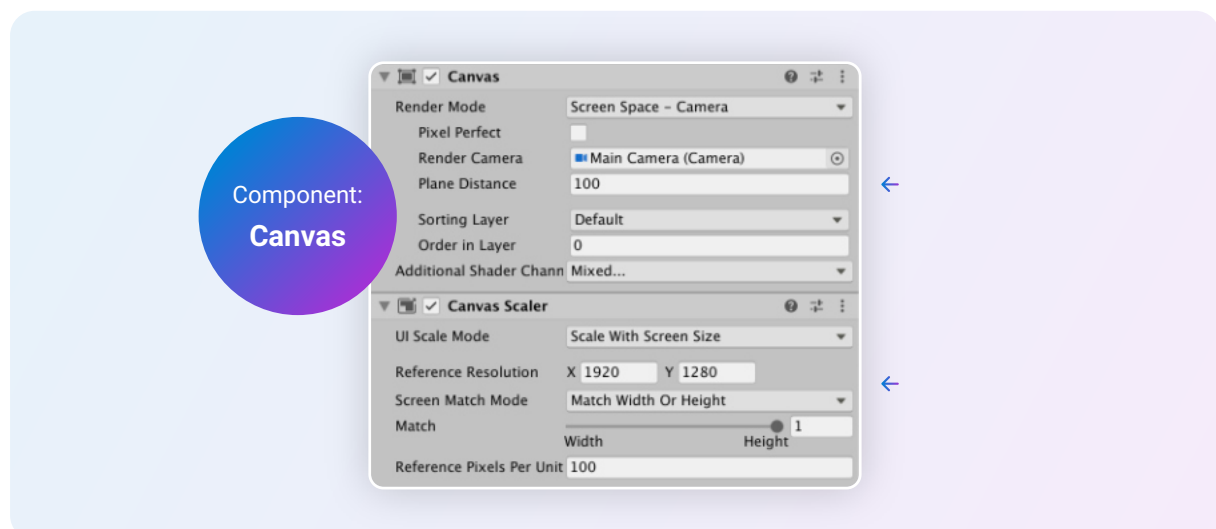
While the source code is not intended to be a universal framework, it can be a very useful reference when it comes to learning how to approach the implementation of a game UI using Unity's built-in UI system.

4.2. Canvas

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

This, together with extensive use of anchors when positioning UI elements, makes it possible to automatically scale the UI across multiple resolutions.

Please note we have optimized the demo for panoramic aspect ratios.



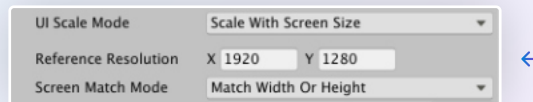
You can also use our pre-built [canvas prefab](#) and then choose (right click) “Unpack Prefab” to quickly create a new scene.

You can find more details about “Designing UI for Multiple Resolutions” in the official Unity documentation [here](#).

4.3. Reference resolution

We are using a reference resolution of **1920x1280**, which works well across a wide range of aspect ratios.

This is particularly useful for mobile development, where screen sizes vary wildly between devices.



You can find more details about “Designing UI for Multiple Resolutions” in the [official Unity documentation](#).

6. UI extensions

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

Two of the most notable ones are the **SceneTransition** component and the **PopupOpener** component.

The pack provides also a **Tooltip**, a **Clean Button** a **Gradient** component, a **SpriteSwapper** and a **URLOpener** component.

5.1. The SceneTransition component

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” button in the Home scene that should transition to the Game 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.

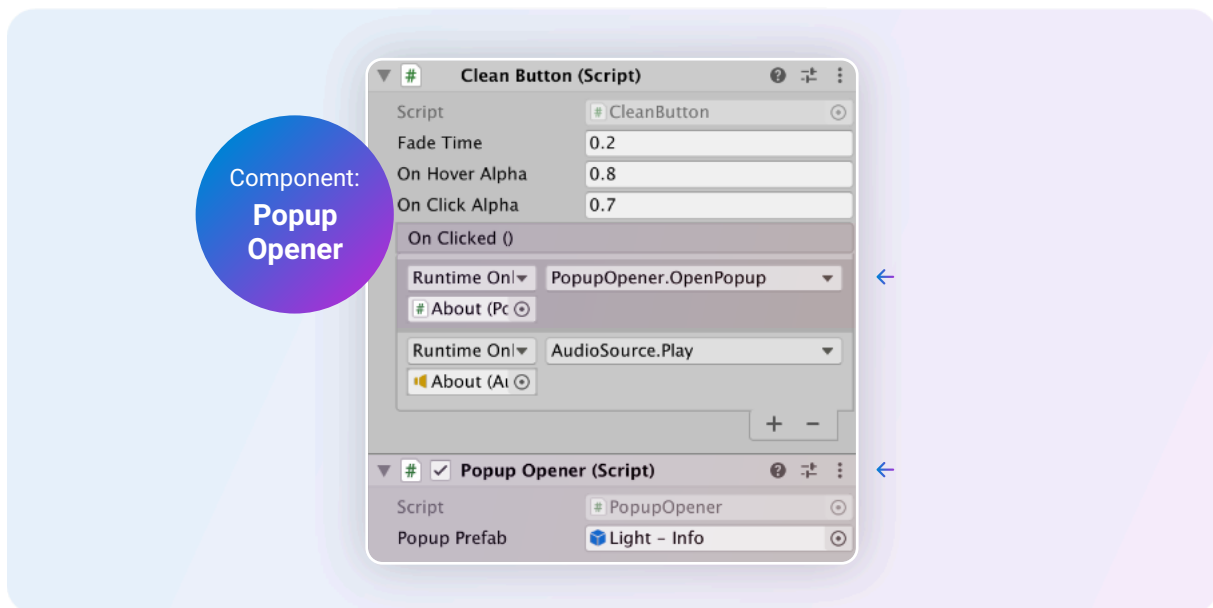
You can make this call in code, too.

5.2. The PopupOpener component

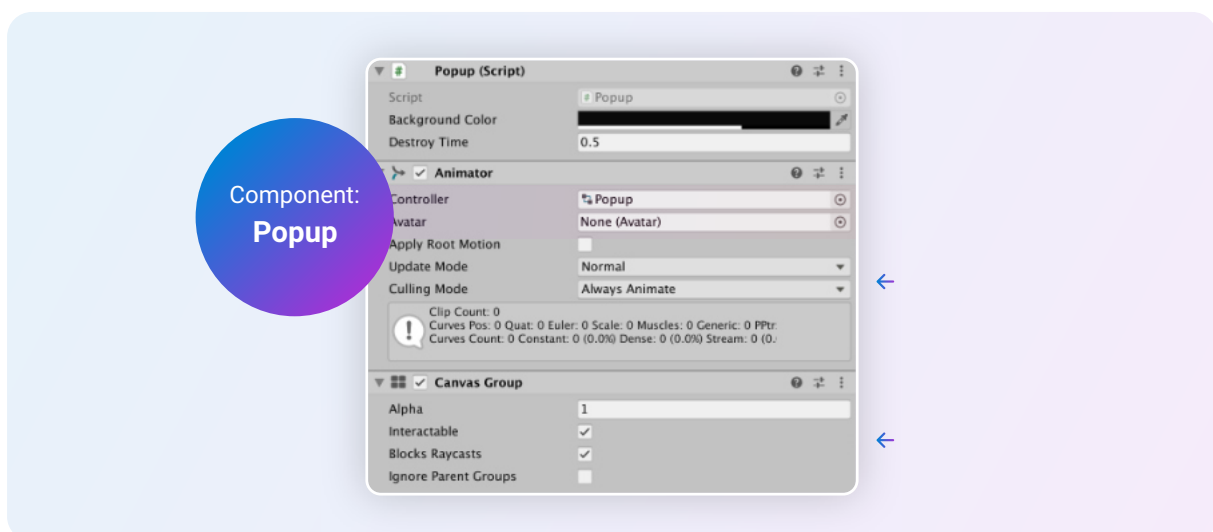
This component provides functionality to open a popup and darkening the background behind it. You can choose the background color as well as the duration time.

Using it is, again, very simple.

We just need to add a **PopupOpener component** to a Button game object. The PopupOpener component carries out the logic needed to open a popup in the current scene.



We also need to add a Popup component to the **Popup Prefab** we want to open (a game object that contains a Popup component).



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.

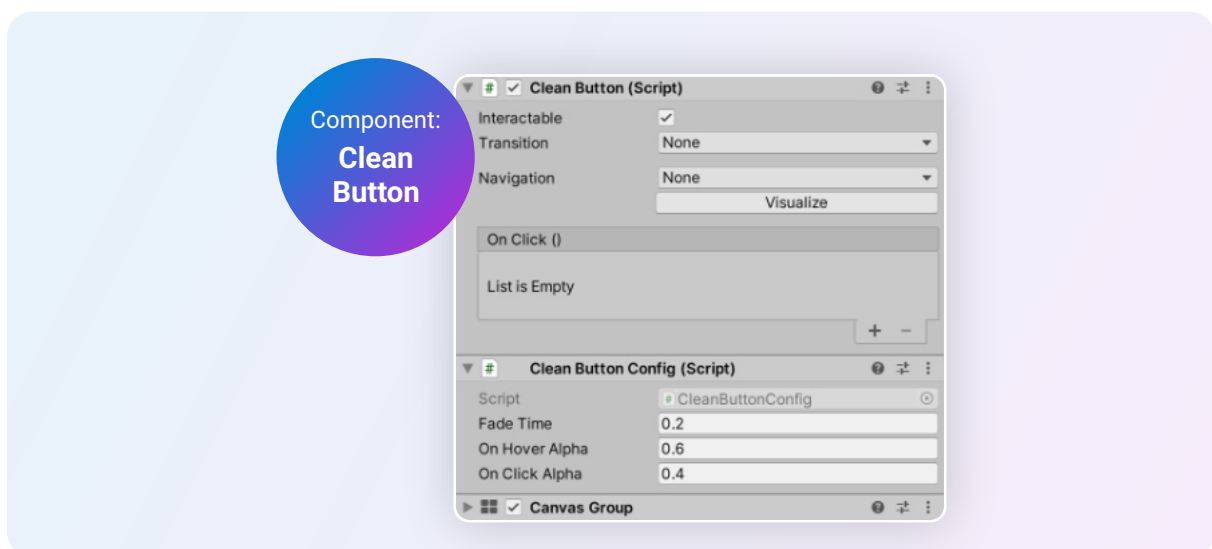
We also need to add the [Animator component](#) for the popup animation.

(Because the popup animation uses the [Canvas Group Alpha](#) we need to add the [Canvas Group component](#) as well)

Now a Button calls the PopupOpener's OpenPopup method in order to open the popup when clicked. You can make this call in code, too.

5.3. The Clean Button component

This component derives from Unity's built-in Button and provides a custom fading behavior when the user hovers over the button or clicks it.



You can specify the configuration of the fading in the attached CleanButtonConfig component:

Fade time: The duration of the fade in seconds.

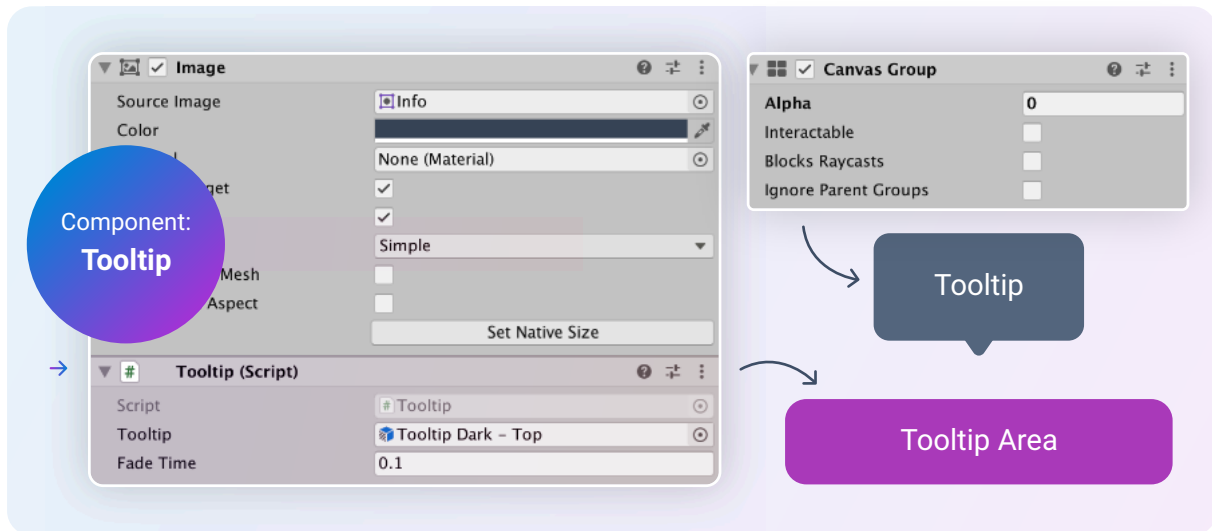
On Hover Alpha: The transparency level when hovering over the button (between 0.0f and 1.0f).

On Click Alpha: The transparency level when clicking the button (between 0.0f and 1.0f).

5.4. The Tooltip component

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 Alpha set to 0, like you can see in the image below.

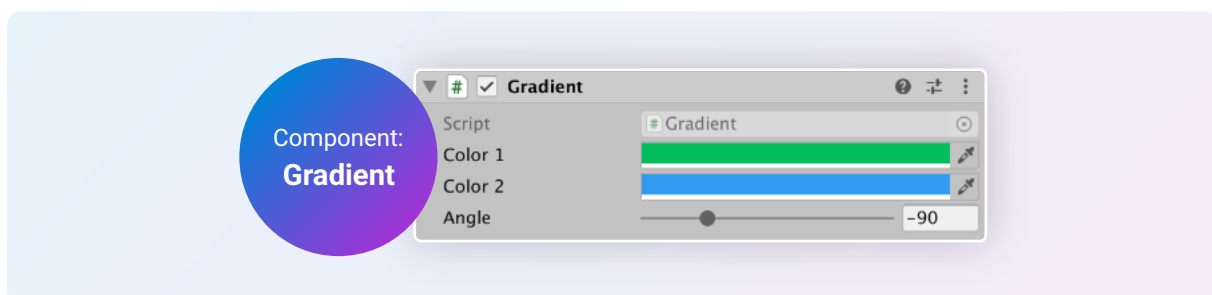


You can find our pre-made tooltip prefabs with a size fitter component (the size of the window fits automatically to its content) inside the folder [prefabs/UI Elements/Tooltip](#), but you can literally choose any object as a tooltip.

5.5. The Gradient component

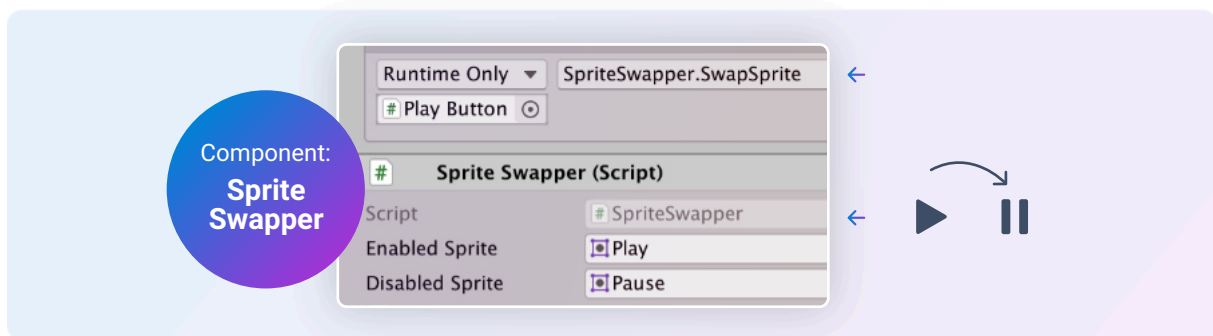
This is a very nice component, because it adds a modern and special style and also additional color options to your design.

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



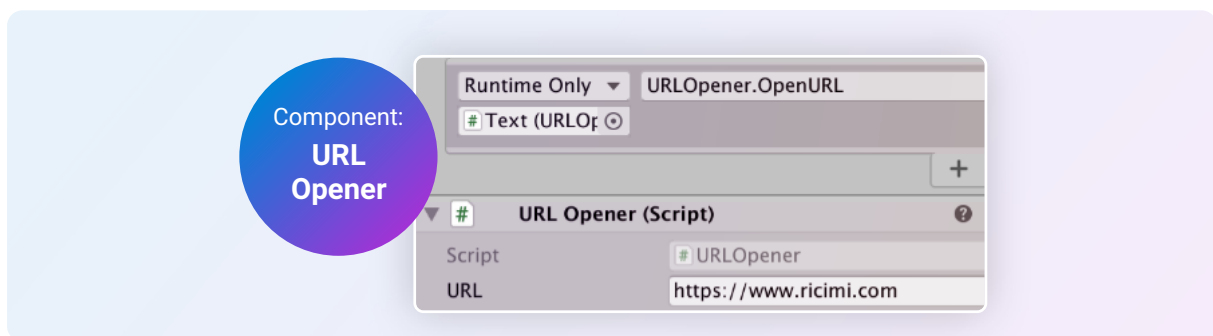
5.6. The Sprite Swapper component

You can easily swap any sprite using this component. The sprites have to be the same size.



5.7. The URL Opener

This component opens a URL from Unity.



7. Contact

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

Please make sure to include your invoice number.

I am always happy to help. 😊





Clean & Minimalist GUI Pack

UI elements, demos and icons for games.



Copyright © ricimi - All rights reserved.



@ricimi



@ricimi.art