A big **thank you** for purchasing my

CarToOn

NEW

GUI Pack

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 **Cartoon GUI Pack** can only be used under the
Unity Asset Store License Agreement which you can find here.

ricimi
ricimi.com

Table of contents

1. Copyright license & terms of use

The **Cartoon GUI Pack** can only be used under the **Unity Asset Store License Agreement** which you can find **here**.

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

After purchasing the **Cartoon 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 Cartoon 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 Cartoon GUI Pack?

The **Cartoon GUI Pack** is a customizable, mobile-friendly game UI pack containing a collection of UI elements, icons and demos that can be used to create a complete game UI in a professional and playful cartoon style.

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.

1.2 TextMesh Pro Version

The accompanying demo project requires **TextMesh Pro 3.0.6** 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 UI.

Included are:

- Buttons
- Dropdown
- Scrollbar
- Input Fields
- Progress Bar (Animations only)
- Slider
- Switches
- Toggles
- Demo scenes
- Popup prefabs
- Icons
- Background landscape images
- C# scripts
- UI components
- Animations
- Sound effects
- Color palette
- .PSD 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 "**GUIPackCartoon**" 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 the color palette for the demo.

Demo/Fonts
Contains OPL fonts. The pack uses two font families called "LilitaOne" (logo and text) and "BPreplay" (text).

Demo/Materials
Contains the materials used for the particle effects (used for the "End Game Popups").

Demo/Resources
/Popups
Contains all popup prefabs.

/UI Elements
Contains all UI element prefabs such as buttons, slider, etc.

Demo/Scenes
Contains the Home, Level, UI Elements and Icons scenes.

Demo/Scripts
Contains the full C# source code of this pack.

Demo/Shaders
Contains a "Gray Tint" shader used for the demo project.

Demo/Sounds
Contains all sound used for the demo project with credits.

Demo/Sprites
Contains all shapes, icons and images used for the demo project.

Sources
Contains the original art source files and Mockups in .PSD format.

5. UI Elements

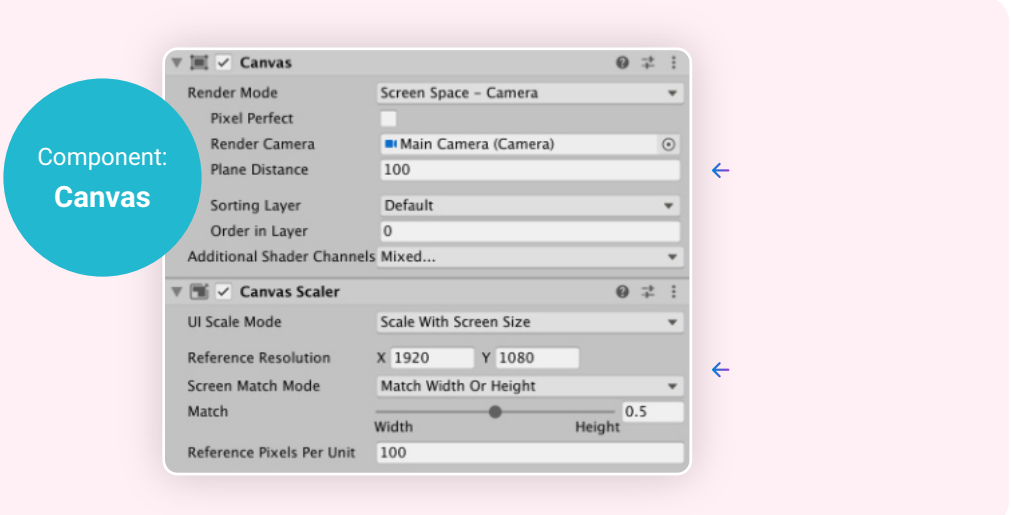
4.1. Unity's built-in UI system

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.



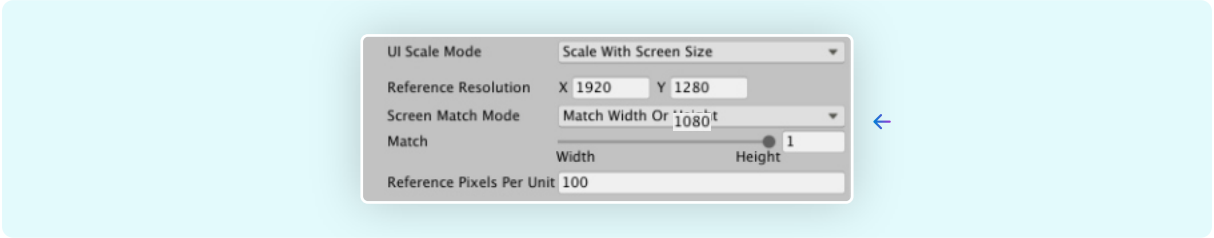
You can also use our pre-built canvas prefab and then choose (right click) “Unpack Prefab” to quickly create a new scene.

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 **1920x1080**, 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.

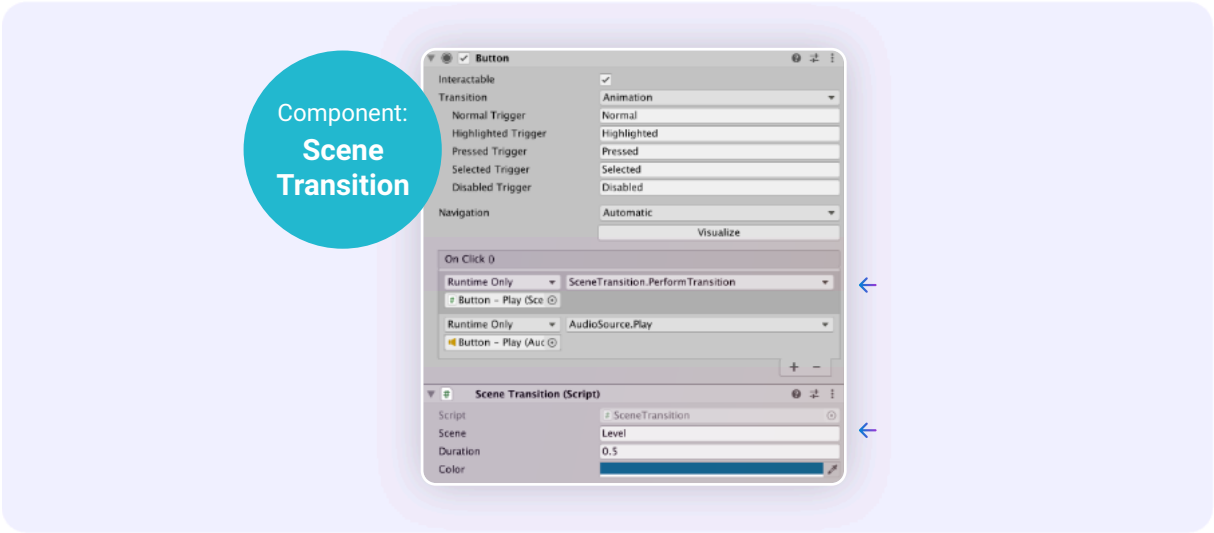
The pack provides also a useful **ColorSwapper** component.

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.

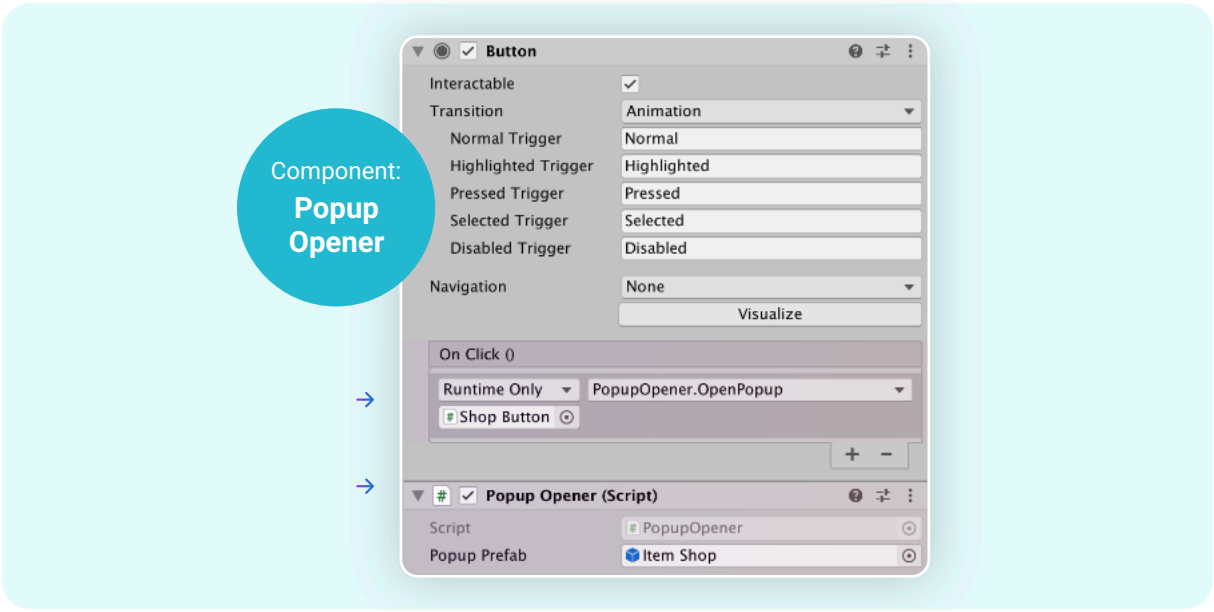
You can make this call in code, too.

5.2. The PopupOpener component

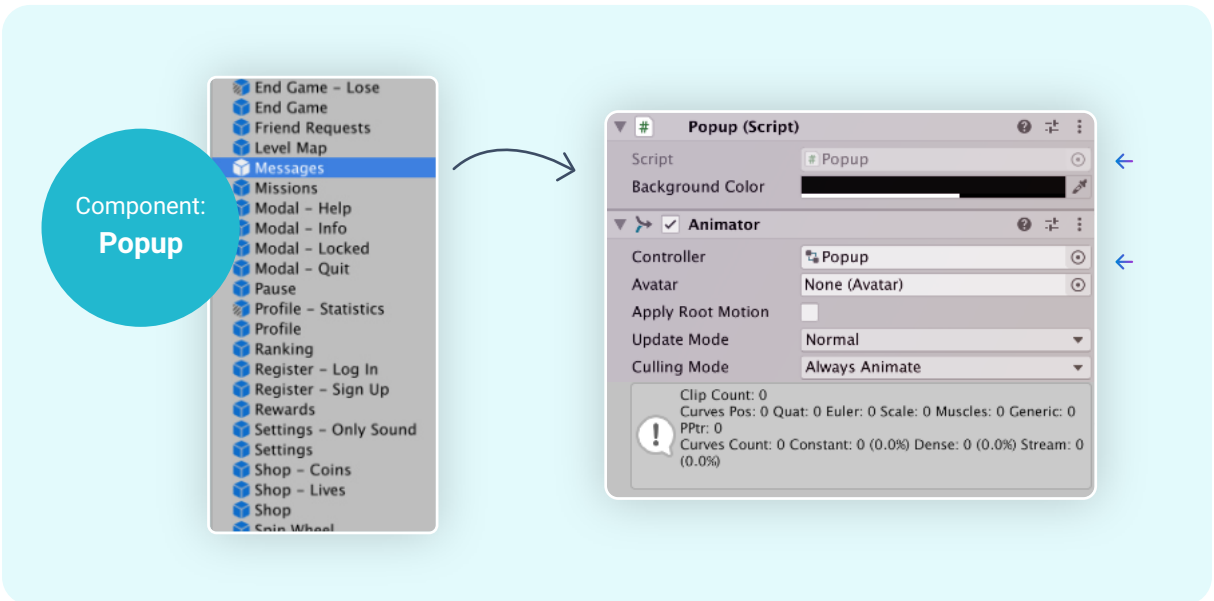
This component provides functionality to open a popup and darkening the background behind it. You can choose the background color as well as the duration time.

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.

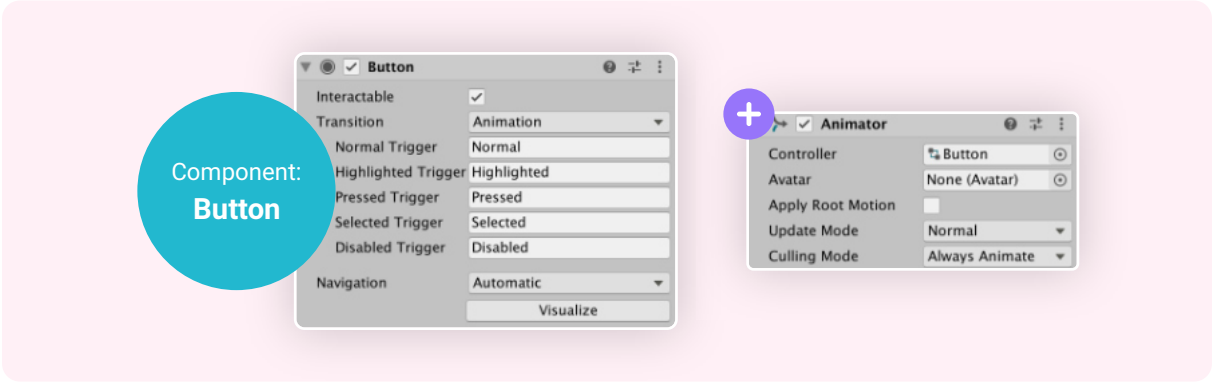
We also need to add the Animator component for the popup animation.

(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.

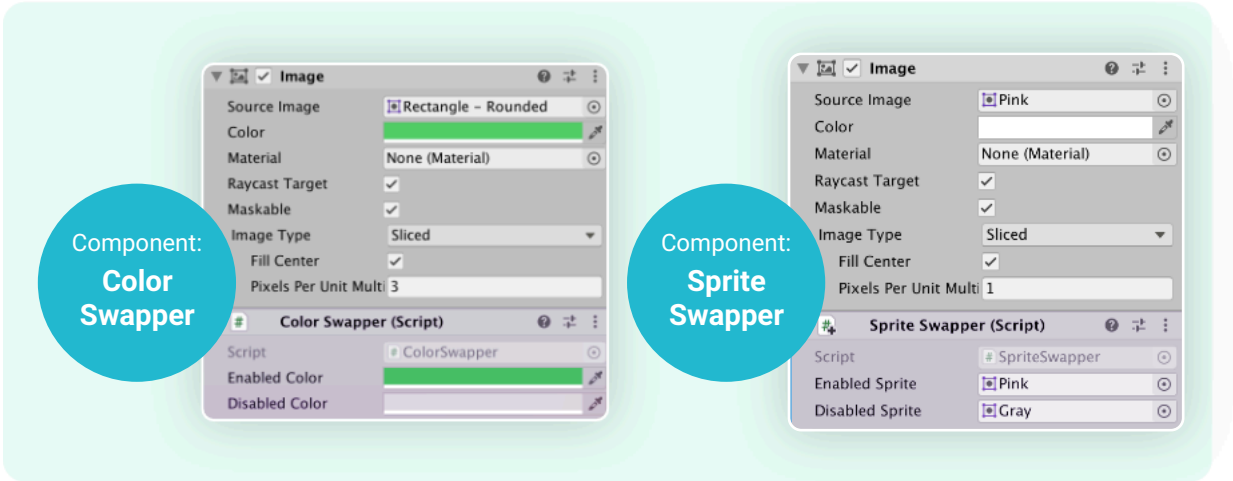
5.3. The Button component

This is Unity's standard button component. I used "Color Tint" or "Animation".



5.4. The Color & Sprite Swapper component

You can easily swap any color or sprite using this components. You just have to add the component to the desired image and then call it from the button component.

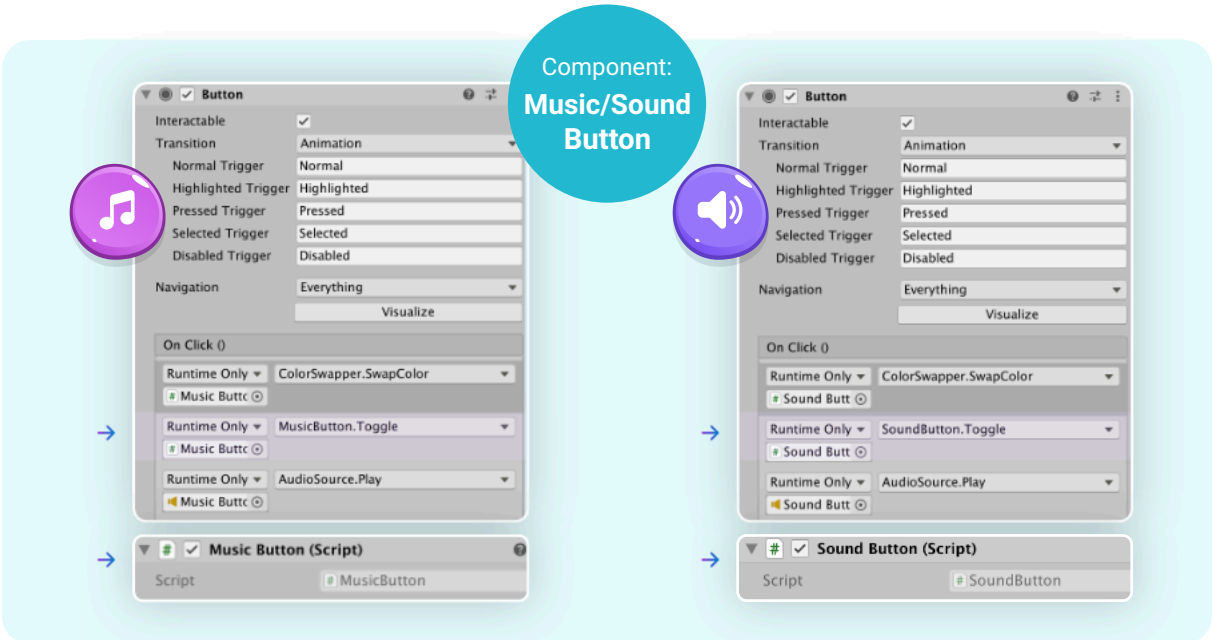


5.5. The Sound & Music Button / Manager component

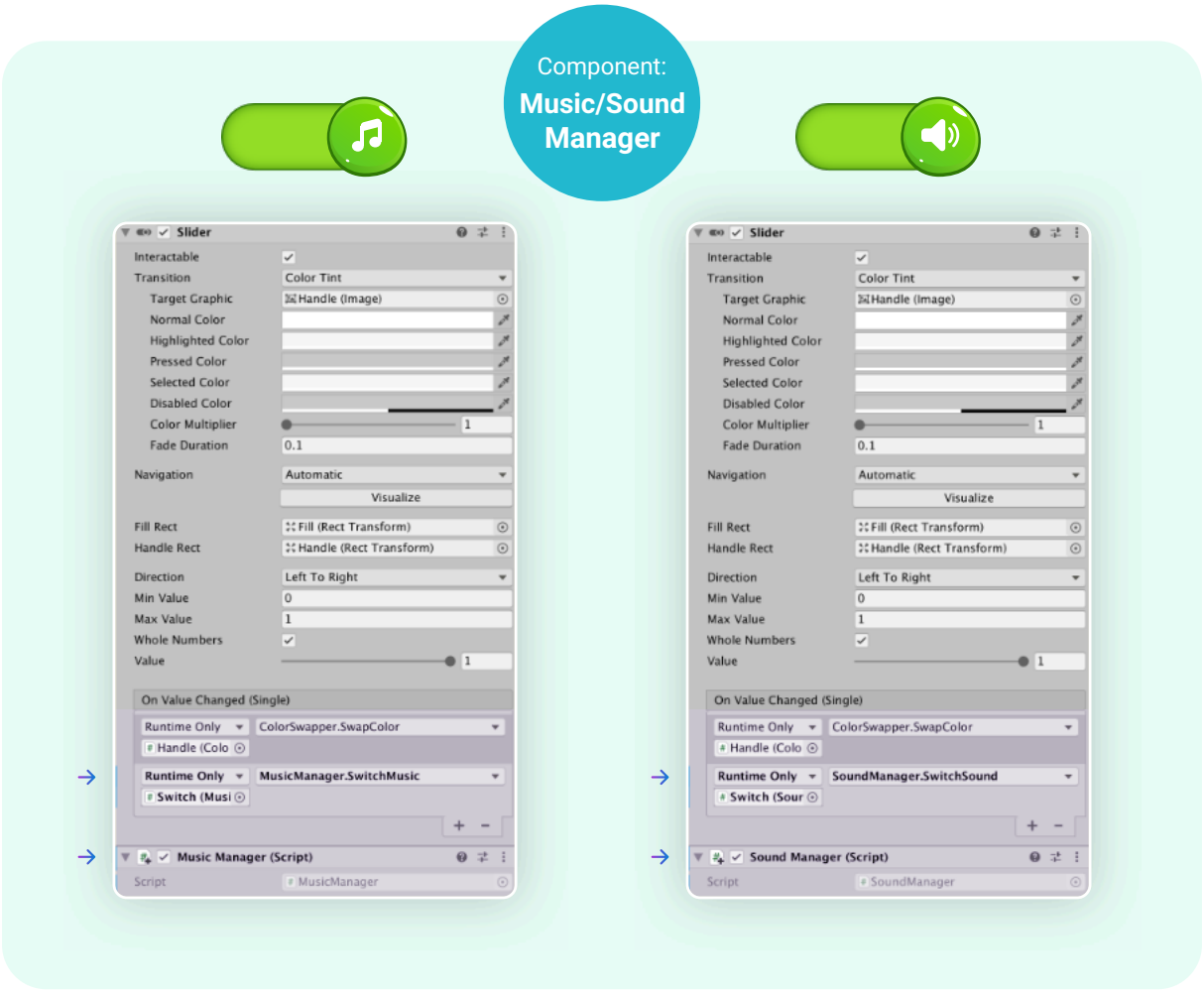
You can easily manage sound effects and background music. First, you need to add an empty “Game Object”, call it InitialPlayerPrefs and add the Initial Player Prefs component. Add it to all scenes you want the music to play.

For the **background music** you need to add an empty Game Object and call it BackgroundMusic. Then you just have to add an “Audio Source” component with your Background Music file. *Choose “Play on awake” and “Loop” if you want the music to start right away to loop this song.*

For the **buttons** you need to add the Music and Sound Button component, as well as the Color Swapper component.



For the music and sound **switches** you need to add the sound manager component instead of the sound button component.



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. 🙂

ricimi
ricimi.com

gamevanilla
gamevanilla.com

@ricimi @ricimi.art