A big **thank you** for purchasing our

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

If you have any support questions, please contact us **here**.
Please make sure to include your **Invoice number**.

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

ricimi
designs

Terms of use

The **Frozen GUI Pack** can only be used under the standard Unity Asset Store End User License Agreement. A Copy of the Asset Store EULA is available here.

Frozen GUI Pack Documentation

The **Frozen GUI Pack** is a customizable, mobile-friendly game UI pack containing many elements that can be used to create a complete game with a frozen look. While the sprites themselves do not depend on any specific Unity version, the accompanying demo project requires Unity 2019.4.0 or higher.

Asset structure

After importing the asset package into your Unity project, you will see all the resources provided live in the GUIPackFrozen folder. This folder is further subdivided into the following subfolders:

- **Sprites:** Contains the .psd files and exported .png files of all the graphics contained in this pack.
- **Demo:** Contains all the assets of an example project that makes use of all the sprites included 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!

- **Screenshots:** Contains the .psd files for the asset screenshots, which are useful as a reference.
- **Documentation:** Contains the documentation of this pack.

Some notes regarding 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.

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* UI 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 find more details about how this works in the official Unity documentation here.

Sorting layers in the demo project

The demo project contained in this GUI pack makes extensive use of custom sorting layers in Unity to define the order of the UI elements on the screen.

When generating the final package to submit to the Asset Store, Unity does not export this custom sorting layer information. The end result is that, after importing the GUI pack into a new or existing Unity project, you may see the rendering order is wrong (e.g., the background or the particles may be on top of everything else). This is only due to the fact that the custom sorting layer is not available.

In order to alleviate this situation, we have provided the original *TagManager.asset* file that contains the layer information in the *Demo/Settings* folder.

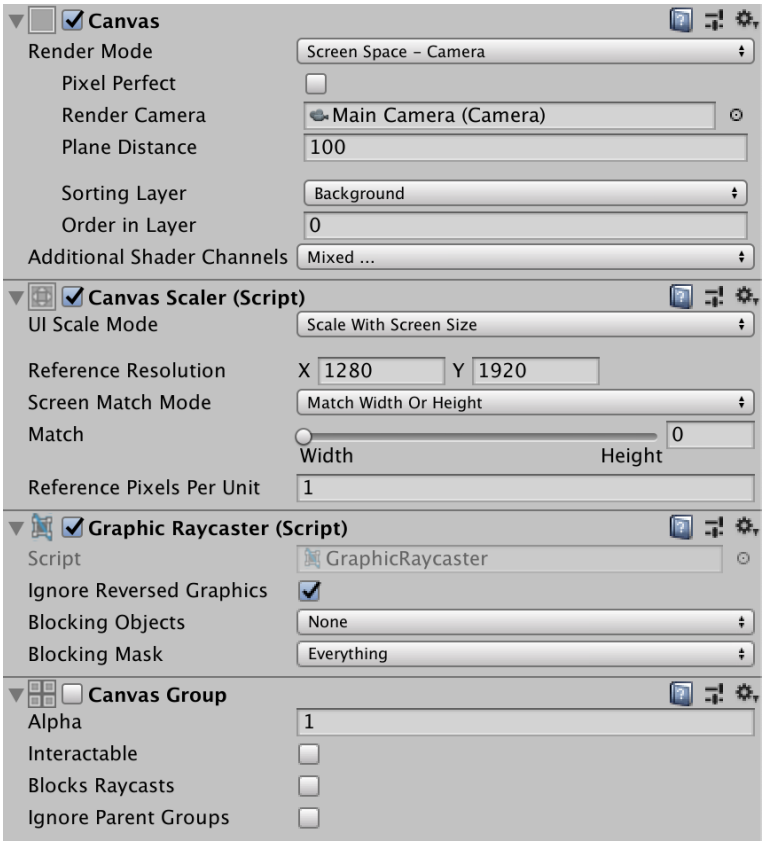
With Unity closed, you may go to your project path and replace the *TagManager.asset* contained in the *ProjectSettings* directory with the one provided by us. After re-launching your project, you will see the rendering order is as intended.

Important: if you have imported the Sweet Cookie GUI pack into an existing project that already defines its own set of custom layers, this will replace your layers with the ones in our GUI pack! Proceed with caution.

Reference resolution

Please note that we are using a reference resolution of 1920x1280 (landscape orientation) and 1280x1920 (portrait orientation), which works well across a wide range of aspect ratios. This is particularly useful for mobile development, where screen sizes vary wildly between devices.

The home scene canvas settings example (portrait orientation):

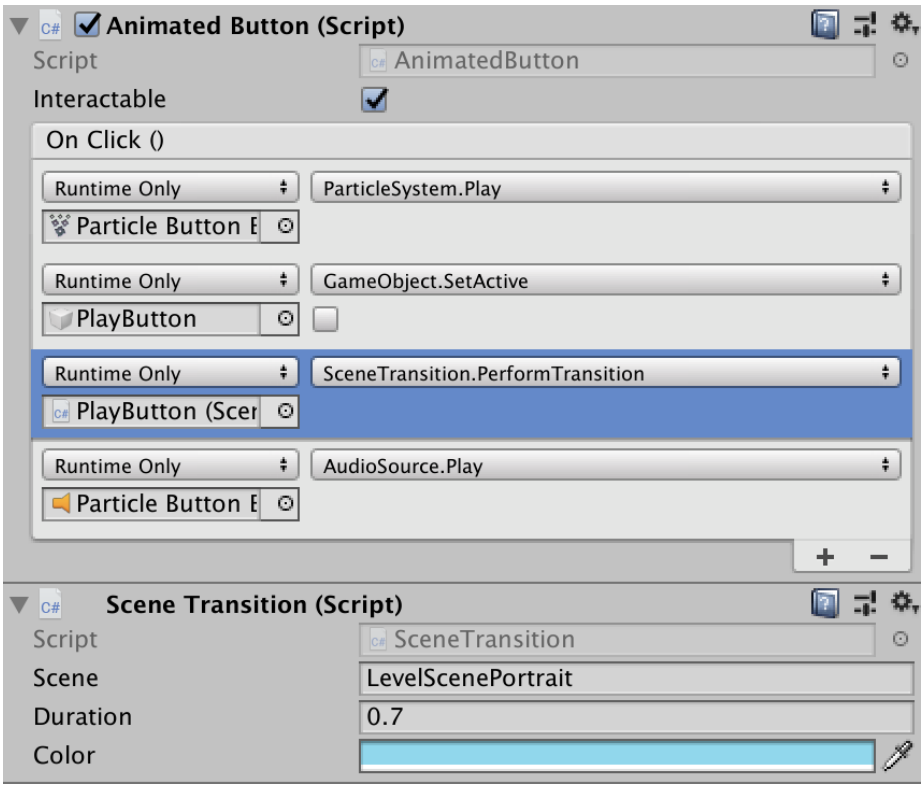


The demo project makes extensive use of Unity built-in UI features, but also provides some useful extensions. Two of the most notable ones are the ***SceneTransition*** component and the ***PopupOpener*** component.

The *SceneTransition* component

The ***SceneTransition*** component provides functionality to transition from one scene to another. Using it is very simple. Consider the following example, where we have a *Play* button in the home scene that should transition to the levels scene when clicked:

We only need to add a ***SceneTransition*** **component** to the *PlayButton* game object.



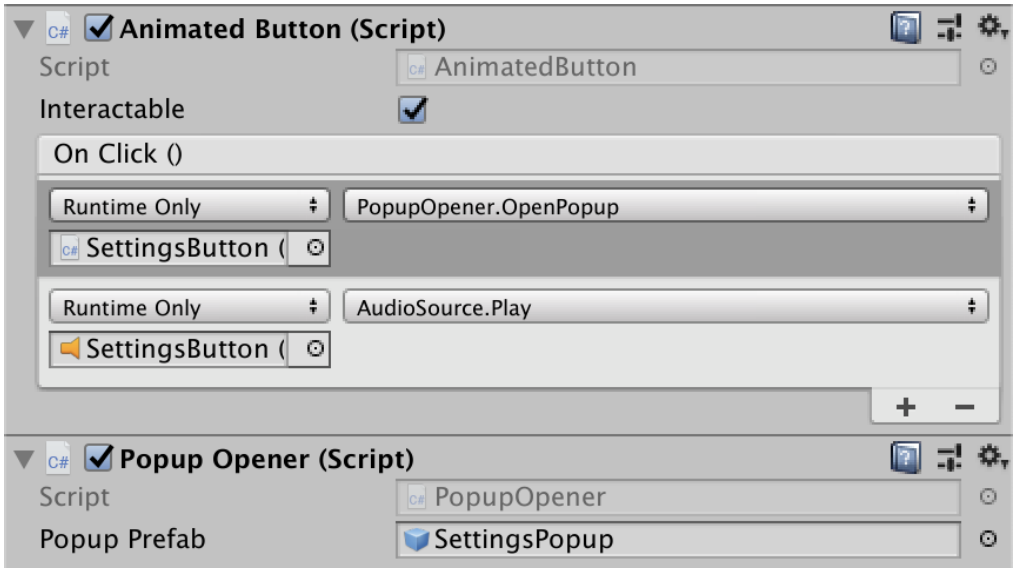
The *SceneTransition* component carries out the logic needed to smoothly fade out from the current scene into the new one. You can specify the destination scene name and the duration and color of the transition.

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

The *PopupOpener* component

The ***PopupOpener* component** provides functionality to open a popup and darkening the background behind it. Using it is, again, very simple. Consider the following example, where we have a *Settings* button in the home scene that should open a settings popup when clicked:

We only need to add a *PopupOpener* component to the *SettingsButton* game object.



The *PopupOpener* component carries out the logic needed to open a popup in the current scene. You need to specify the popup prefab to open (a game object that contains a *Popup* component).

Note how the very same *SettingsButton* calls the *PopupOpener*'s *OpenPopup* method in order to open the popup when clicked. You can make this call in code, too.

Thank you for supporting us by buying this pack. You're helping us to create **more assets** and **exciting products** for game developers.

Frozen GUI Pack

ricimi.com

© gamevanilla

www.gamevanilla.com

Follow us:

@ricimiart @gamevanilla