



User Manual

Table of contents

1. [Introduction - What is Gamestrap UI](#)
2. [Installation/setup](#)
 - [Download Gamstrap UI from the asset store](#)
 - [Updating to new versions of Gamestrap](#)
 - [Support channels](#)
3. [Assets description \(glossary\)](#)
 - [Gamestrap Theme Asset](#)
 - [Gamestrap Modifiers](#)
 - [UI Effects](#)
 - [Text Override](#)
4. [Graphic assets](#)
 - [Fonts](#)
 - [Icons](#)
 - [9-sliced images](#)
5. [Asset folder Structure](#)
6. [Gamestrap UI Window \(toolkit\)](#)
 - [Introduction \(what is and does\)](#)
 - [UI Description](#)
7. [Element Panel](#)
 - [Filter properties](#)
 - [Type](#)
 - [Tag](#)
 - [Regex](#)
 - [Nested Elements](#)
 - [General List management](#)
 - [Colors](#)
 - [Modifier](#)
 - [Nested Elements List](#)
8. [Examples](#)
 - [Complete Project](#)
 - [Example Wire](#)

1. Introduction - What is gamestrap UI

Gamestrap UI 5.0 helps you create faster UI through the unity UI system. It provides you with a set of tools, assets and examples to help you make your own designs in less time.

This document is intended as a manual for new users and basic documentation for existing users of Gamestrap UI.

2. Installation/setup

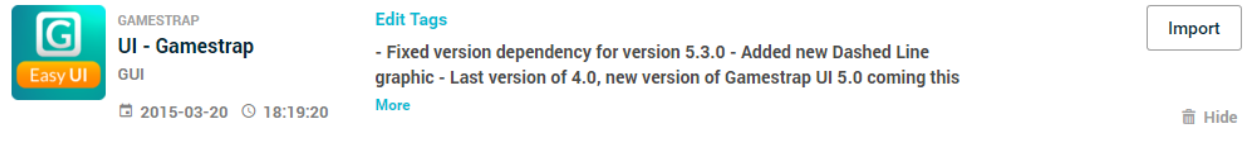
Gamestrap UI can be purchased and downloaded from the unity asset store at the following link <https://assetstore.unity.com/packages/tools/gui/ui-gamestrap-28599>

Download Gamestrap UI from the asset store

Once you purchase Gamestrap UI you can import the asset through the Unity Editor into any open project. This will download and import the necessary files for the asset to work into your current Unity project.

1. Open the Asset store window in the Editor (if it's not already open). This can be done by going to Window>General>Asset Store or just Window>Asset Store for some versions of the editor. It will bring the asset store window.
2. From here click on the "My Assets" button, which is the 5th from the left in the Asset Store Window.

3. Search for Gamestrap UI and click on import. If this is the first time you import the asset you need to first click on download and then on import.



4. Select which files to import and which to pass on. This is useful if you don't want to add examples or prefabs that you are not going to use.

Note: If this is the first time you download Gamestrap UI or you are not sure which files you need, we recommend to just download them all to avoid conflicts of missing files. For more details on what each folder contains, and where you can find certain files, refer to the [Asset description](#) and the [Asset Folder Structure](#) sections.

Updating to new versions of Gamestrap

If a new version of Gamestrap is released and you want to download the new update you will have to reimport the asset following the same steps for [Downloading Gamestrap UI from the asset store](#).

IMPORTANT: Upgrading to Gamestrap UI 5.0 from previous versions will lose compatibility with old instances of some of our Game Objects, and requires for settings like color palettes to be created in the new systems. Be mindful of this when upgrading your version of Gamestrap UI that is currently being used in a project. We suggest backing up project files before doing an upgrade.

Support channels

For any Gamestrap UI related inquiries you can contact us at www.gamestrap.info or write us an email to contact@gamestrap.info

3. Assets description (glossary)

Gamestrap Theme Asset

The Gamestrap theme asset is the pivotal point for all Gamestrap UI operations. A Gamestrap theme is a custom scriptable object supported by Gamestrap UI. This asset can store multiple element configurations and is used with the [Gamestrap UI Window](#) to edit and apply such settings to specific objects in the current Scene, well go into more detail on what this settings are, and how they affect objects in further sections of this document.

You can create new instances of Gamestrap Theme through the Asset unity menu (**Asset>Create>Gamestrap>Theme**) or by right clicking on the project window (**Create>Gamestrap>Theme**).

Multiple instances of this asset can be created, in order to separate different configurations. This is especially useful in projects with different aesthetic “themes”, to manage different color palettes easily or just for general project organization.

Gamestrap Modifiers

Gamestrap Modifiers are scriptable objects with a predefined effect that can be applied through Gamestrap UI. Multiple instances of the same modifiers can be created to store multiple configurations for the same effect. By default gamestrap includes the following Modifiers:

UI Effects

The category of modifiers that are only used in UI Game Objects.

Gradient

Adds a [gradient](#) effect vertically between two colors. “Top” and “Bottom” color variables must be set to the respective colors. This effect is added on top of the base color by multiplying its values, meaning that the gradient effect will also have the tint of the base color of the UI Object the effect was applied to.



Mirror

This creates a mirror image of the applied object. This is done by creating a copy of the UI object, flipping it downwards, and moving it down. The “Scale” and “Offset” modify the vertical scale and position of the reflection.

On top of the base mirror effect, a gradient effect similar to the [Gradient Modifier](#) is added, and affects only the reflection. This is especially useful for making the reflection fade by using the alpha of the gradients.



Radial Gradient

This is similar to the regular [Gradient Modifier](#), but instead of applying it vertically, this applies the gradient following a defined radius. For this, a “Center Position” is set, and this marks the center of the radial gradient. The center has the most amount of “Center Color” applied, and the gradient then is applied in the radius set in the “Radius” variable, transitioning from the set center color to the outer color.

This gradient (like the regular [Gradient Modifier](#)) multiplies the gradient and the base UI Object's color.

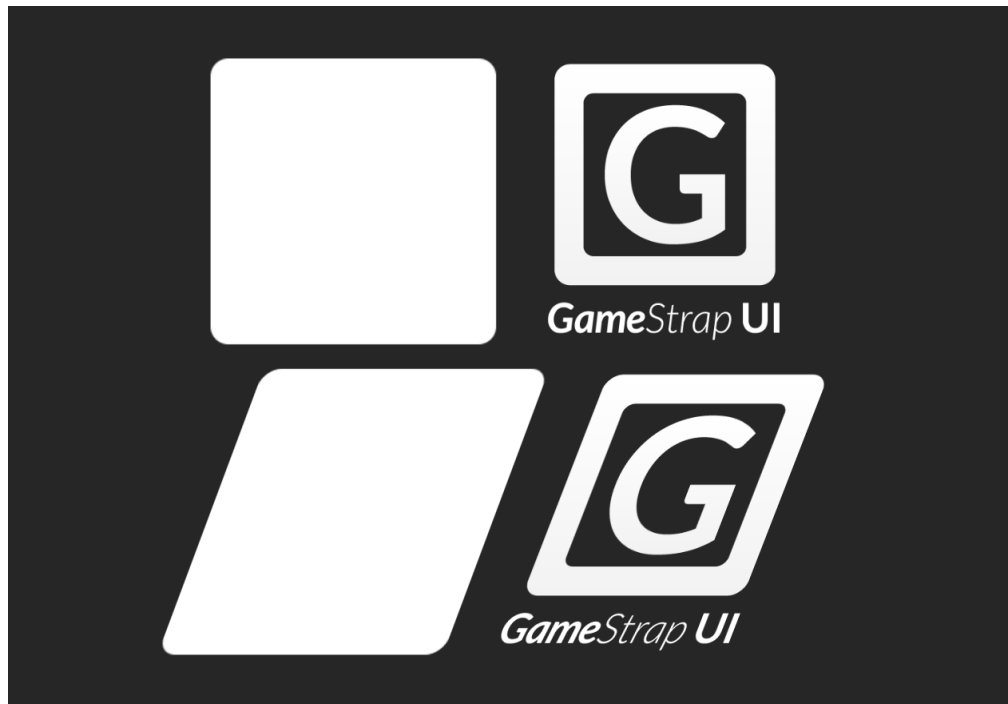


Shadow

This modifier allows Gamestrap UI to add and manage the default unity's shadow effect. By using this instead of the base unity component, the effect can be added, modified and deleted through the [Gamestrap UI Window](#). For more information on the shadow effect, refer to the unity documentation on the [shadow component](#)

Skew

This applies a distortion effect similar to the skew effect on other image editing software. This effect is controlled by the "skew" variable, adding more distortion or tilt to the right of the UI object with greater values and to the left with negative values.



The skew Modifier also includes a perspective effect, that can distort the UI object replicating a shift on perspective. As the skew, this can be controlled with the “Perspective” variable, adding more distortion with greater values.

Note: The perspective effect is suggested for 9-Slice and text UI elements. Other UI elements may not produce good results because of meshes with low poly counts.



Text Override

This modifier allows for easy application of settings to text components. The modifier stores a configuration that is then applied to the text component of the UI element, overwriting whatever settings were there.

Note: This does nothing if no text component is attached to the target UI Object.

4. Graphic assets

Included with Gamestrap UI are a set of graphic assets that help kickstart any UI project with some basic UI prefabs. All the included icons and images are on .psd format, making them easy to edit right away with other 3rd-party software.

Fonts

Fonts are used in all text related UI components. Unity allows fonts to be easily added, for more information on this, refer to [unity's documentation on Fonts](#).

We include the following fonts to complement the ones that come with unity:

- Basis33
- Bebas
- Bigfish
- Fira
- Lato
- Monospace
- OpenSans
- PixelFJpt1
- Polaris
- Ubuntu

Icons

Small sprites that are commonly used in menus. We include an assortment of commonly used icons. These can be found in the following directory: **Gamestrap>UI>Graphics**

9-sliced images

These are images ready to be used as are or as 9-slice sprites. For more information on 9-slice sprites, their use, or how to set a image to be 9-slice, refer to the [unity documentation](#).

5. Asset folder Structure

Gamestrap

- /Color:** Scripts related to the management of color and colored objects.

- /Editor:** Scripts for the unity editor.

- /UI:** The main files of Gamestrap UI. Scripts and game objects for use in UI.

- /Editor:** Scripts for the unity editor.

- /Effects:** Contains [Gamestrap's Modifiers](#)

- /Examples:** Prefabs, scripts, and scenes to learn Gamestrap UI features.

- /Fonts:** The [fonts](#) that are included with Gamestrap UI

- /Gizmos:** Icons for the Unity editor integration.

- /Graphics:** 2D image assets ready to be used as [icons](#) or [9-slice images](#)

- /Helper Scripts:** Commonly used scripts useful for most UI implementations

- /Scripts:** Source code and used by the asset.

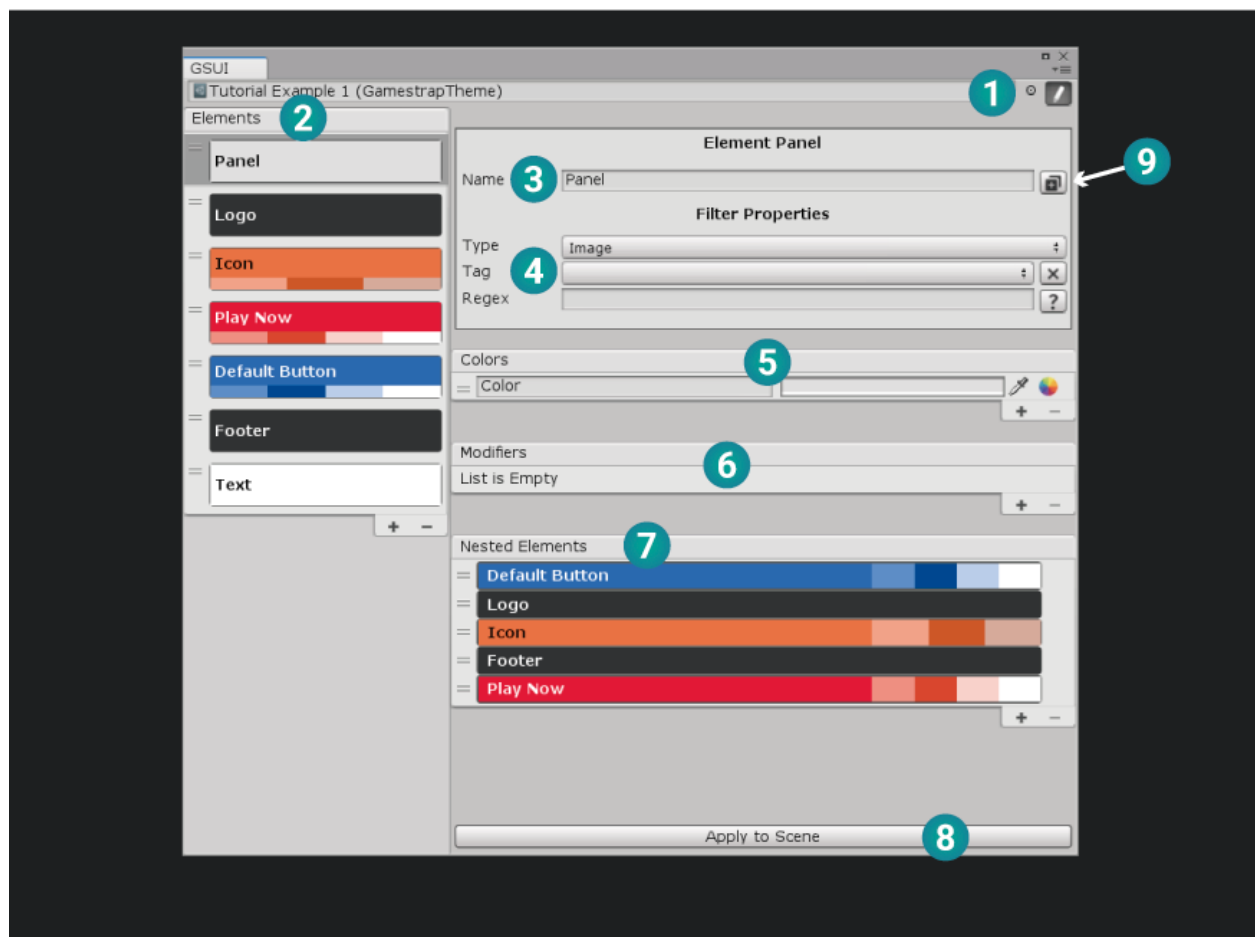
- /Showcase:** Scenes to showcase Gamestrap UI features.

6. Gamestrap UI Window (toolkit)

Introduction (what is and does)

The Gamestrap UI window allows to edit and apply [Gamestrap modifiers](#) and [element](#) configurations. This window edits the contents of a theme, therefore at least one theme instance is required for it to work. Check the [Gamestrap Theme asset](#) section for more information on how to create a theme instance in your project.

UI Description



1. **Theme Selection:** Selects what theme is currently being edited.
2. **Elements List:** Lists of all the created elements in the selected theme. Select a Element to begin editing it.
3. **Name:** This sets the name of the element configuration. This is the name that appears in the element list on the left of the window.
4. **Filter Properties:** Options to [configure filters](#) of the selected element.
5. **Colors list:** List of applied [Color configurations](#) of the selected element.
6. **Modifier list:** List of applied [Modifiers](#) of the selected element.
7. **Nested elements list:** List of the elements set to act as nested elements for the current selection.
8. **Apply to scene button:** Apply the selected configurations to all elements that fit the filters of the selected element.
9. **Duplicate button:** Create an identical copy of the selected element and add it to the element list.

7. Element Panel

Filter properties

These properties filter which game objects in the scene will be affected by the gamestrap element configuration. This, for example, allows to only target a type of game object like only buttons, or game objects with "label" in their name, or any other filter that you may think of.

Type

This filters objects by what UI component they have, selecting only objects with the specified Type. This list contains the

common UI components from unity, with the addition of the “Nothing”, “Everything”, and the “custom” options.

The Nothing and Everything are self explanatory, selecting all objects without any of the listed UI components, or selecting every kind of object, respectively.

The custom option is intended for integrations of custom, user created, components. This selects all objects with components that implement “ISearchableComponent”.

Tag

This filters objects that have the selected Tag applied to them.

For more information on Tags, how to apply them to objects and how to create new tags refer to the [unity documentation on tags](#).

Regex

This filters objects that have a specific name. This is done by using regular expressions to make specific selections on the name of the objects. For more information on Regular expressions, refer to the [c# reference page on regular expressions](#). This page can also be accessed from the Gamestrap window by pressing the help button next to the regex field.

Nested Elements

Nested elements allow to assign other element configurations to children of the element. These nested elements will still use their respective filters, but will only look for child objects of the selected element.

General List management

Most lists in the Gamestrap UI share the same behavior of unity's lists. Individual elements can be rearranged by dragging them from the icon on the left. More entries can be added with the plus icon on the bottom of the list, and with the minus the selected element can be removed from the list.

Colors

This list the different colors to be used by the element. They go in key-value pairs, where key is the name of the color and the value is the actual color. Note that these items will change when changing the element type, as it must match the names used by such components to work. additional colors can be added, but for the unity components to work, the color's key name must match those of the component (like 'pressed', 'normal', etc).

For convenience, on the right of the color picker is a button to open the color helper tool, this is a custom tool from gamestrap that helps selecting colors according to different color theory rules.

Modifier

This lists all the applied modifiers on the element. For more information refer to the [Gamestrap modifiers](#) section on this document.

Nested Elements List

This lists all the elements to be Nested. For more information refer to the [Nested Elements](#) section on this document

8. Examples

Complete Project

A fully working UI project showing a way to implement the Unity UI with scene transitions and static levels

To run this example you need to add the following scenes to the

build settings

- Gamestrap/Examples/Complete Project/Screens/Gameplay/**Gameplay.scene**
- Gamestrap/Examples/Complete Project/Screens/Levels/**Levels.scene**
- Gamestrap/Examples/Complete Project/Screens/MainMenu/**MainMenu.scene**

Example Wire

A UI layout showing how it's possible to implement multiple themes to a UI