Guido Flat

This package brings a special freely colorizeable GUI asset into your application or game. If you are bored with the original and simple Unity3D User Interface, just change it with this new GUI. It’s easy!

It’s designed for Unity3D’s Canvas system and it’s fully compatible with Unity 2017. It supports all platforms, including mobiles or consoles as well!

Unlike the other Guido packages (Guido Blue Elegant and Guido Mr Black) the UI elements of this package doesn’t have any color scheme. Their colors can be set freely.
To help you to make it quick and easy there are 4 different scripts in the package.
You can control them from code (runtime) or directly from the inspector in edit mode as well.

Inspector usage

The **ColorizeImage**: Add an empty GameObject to your scene and add the script as new component. It has a list to add the Image objects to be colorized called “Images To Colorize”. Now add the drop link of the Image objects to be colorized to the list. Once you are done to add your elements just enable “Auto Update” checkbox in the inspector above the list. Now you can change the color of the given elements just by changing the color of the ColorizeImage component in Inspector. Thats all.
The **ColorizeText** is working the same way but on Text objects.

Working with the **ColorizeImageInChildren** is a bit different. It can be used to change ALL the Image object's color under the GameObject of the script in the hierarchy. There are lists to exlcude some Images to protect their original color. Make your hierarchy with a parent object and add the script to the parent. Now drag and drop the links of the components to be excluded first! It's important to do this BEFORE enabling "Auto Update" in inspector. After enabling the flag, all the Images under the parent (except the excluded ones) will be updated with the selected color.
The **ColorizeTextInChildren** is working the same way but on Text objects.

To get familiar with inspector usage, please check the provided demo scene flat_inspector. The scene contains some buttons in different shapes. The "Canvas/Buttons" element of the scene has got a ColorizeImageInChildren and a ColorizeTextInChildren component. Both has got some elements on their exclude list. Feel free to modify the lists to test the functions. (they are based on the button shape for default)
There is an empty GameObject called "Canvas/TextRounded" with a ColorizeText component. This one sets all the Rounded button type's text color based on its Texts To Colorize drop link list.

Usage from code

Follow the steps of the inspector usage, add the scripts as components and set the corresponding lists but do not set the "Auto Update" flag, leave it inactive. In your controlling class make a reference to the used Colorizers and just modify their color on the fly to trigger the color change. The setter in the color member will do the rest. Please check the provided demo scene

“flat”, where the controlling class called “Demo” is attached to the Canvas object. The Canvas has got a ColorizeImageInChildren component as well with a pretty long exclude list. Right under the Canvas there is 3 empty GameObject with a ColorizeText component. There are references to the 3 ColorizeText and the ColorizeImageInChildren as well and its SetUI() function passes the color to the 4 Colorizer where the setter updates the colours of the elements based on their lists and the hierarchy.
That’s all.