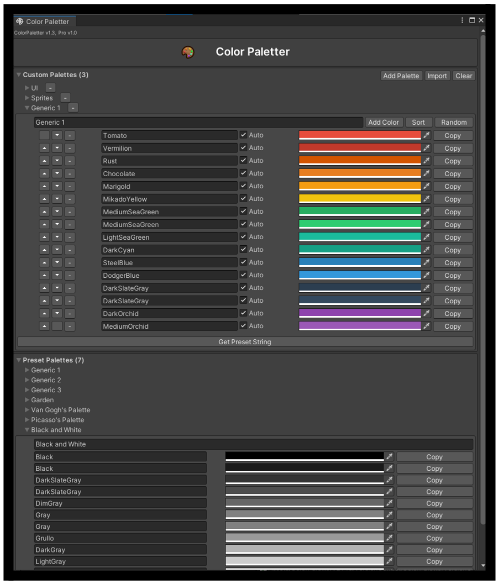
Welcome to the

ColorPaletterV2!

v1.4.0

[Pro version documentation]

The Color Paletter is a Unity package (made in version 2021.3.12f1) that allows you to create and manage custom color palettes easily. It provides a simple and intuitive interface for creating, organizing, and accessing color palettes in your Unity projects.



Generic 1

Tomato	✓ Auto	🟥	Copy
Vermilion	✓ Auto	🟥	Copy
Rust	✓ Auto	🟧	Copy
Chocolate	✓ Auto	🟧	Copy
Marigold	✓ Auto	🟧	Copy
MikadoYellow	✓ Auto	🟨	Copy
MediumSeaGreen	✓ Auto	🟩	Copy
MediumSeaGreen	✓ Auto	🟩	Copy
LightSeaGreen	✓ Auto	🟩	Copy
DarkCyan	✓ Auto	🟩	Copy
SteelBlue	✓ Auto	🟦	Copy
DodgerBlue	✓ Auto	🟦	Copy
DarkSlateGray	✓ Auto	⬛	Copy
DarkSlateGray	✓ Auto	⬛	Copy
DarkOrchid	✓ Auto	🟪	Copy
MediumOrchid	✓ Auto	🟪	Copy

Black and White

Black	⬛	Copy
Black	⬛	Copy
DarkSlateGray	⬛	Copy
DarkSlateGray	⬛	Copy
DimGray	⬛	Copy
Gray	⬜	Copy
Gray	⬜	Copy
Grullo	⬜	Copy
DarkGray	⬜	Copy
LightGray	⬜	Copy

Features:

- Create custom color palettes: Design and organize your own palettes to suit your project's needs
- Create random color palettes: Generate random color palettes of any length, from 5 to an infinite amount
- Accessible through code: Use Color Paletter's class name to access and utilize color palettes programmatically in your scripts
- Simple to use: Color Paletter's user-friendly interface makes it easy to create and manage color palettes without any hassle

Editor Usage:

Palette Creation:

- To access the Color Paletter Window, go to “Window>ColorPaletterV2>Color Paletter” in the top bar
- Create a new palette by clicking "Add Palette"
- Expand the custom palettes section and open your new palette
- Give the new palette a name of your choice and click "Add Color"
- Set a color
- Toggle "Auto" if you'd like the name to be automatically generated based on the inputted color

Importing/Exporting Palettes:

- On the palette you'd like to export, click "Get Preset String"
- To import, click "Import" and paste in your preset string. Simple!

    - To export multiple, I suggest just creating a list of palette strings in some sort of document :)
    - You can also import preset palettes

Palette Randomization:

- Have at least one color in the list
- Press "Random". This will generate 5 new carefully selected colors for your Palette
- If you want more colors to be generated, add some new entries, then click "Random"

Code Usage (Under the "ColorPaletterV2" namespace)

(there are other public static methods, but you shouldn't really need to mess with them at all):

- ColorPaletter.GetColor(string colorName)
    - **Returns** first palette color with the inputted name, null if not found

      ```
      Color myColor = ColorPaletter.GetColor("My Color").color;
      myColor = new Color(0,0,0,0);
      ```

    -
- ColorPaletter.GetPaletteByName(string paletteName)
    - **Returns** palette by the given name, null if not found

      ```
      ColorPalette palette = ColorPaletter.GetPaletteByName("My
      Palette");
      ```

- ColorPaletter.SetColor(ColorPalette palette, string targetColorName, Color newColor)
    - Changes the desired color from a given palette (an alternative to the previous method)

      ```
      ColorPaletter.SetColor(ColorPaletter.GetPaletteByName("My
      Palette"), "My Color", new Color(0,0,0,0));
      ColorPaletter.SetColor(ColorPaletter.GetCustomPalettes()[index]
      , "My Color", new Color(0,0,0,0));
      ```

    - **Overloads**:
        - SetColor(string paletteName, string targetColorName, Color newColor)

          ```
          ColorPaletter.GetColorFromPalette("My Palette", "My
          Color", new Color(0,0,0,0)););
          ```

- ColorPaletter.GetColorFromPalette(string paletteName, string colorName)
  - **Returns** first found palette color from specified palette, null if not found

    ```
    ColorPaletter.GetColorFromPalette("My Palette", "My Color");
    ```

  - **Overloads**:
    - (ColorPalette colorPalette, string colorName)

      ```
      ColorPaletter.GetColorFromPalette(myPalette, "My Color");
      ```

- ColorPaletter.AddNewCustomPalette(string paletteName)
  - Add a new custom palette with the provided name
  - **Returns** newly created palette for further use

    ```
    ColorPalette newPalette =
    ColorPaletter.AddNewCustomPalette("New Palette");
    ```

  - **Overloads**:
    - (string paletteName, List<PaletteColor> paletteColors)

      ```
      ColorPalette newPalette =
      ColorPaletter.AddNewCustomPalette("New Palette",
      myListOfPaletteColors);
      ```

- ColorPaletter.GetCustomPalettes()
  - **Returns** a list of your custom created color palettes

    ```
    List<ColorPalette> customPalettes =
    ColorPaletter.GetCustomPalettes()
    ```

...

If you have any further questions, please let me know at christian.oxdude@gmail.com
I hope you enjoy this package :)