DTT
APPS WEB GAMES

SINCE 2010

COLOR

Documentation| 31-05-22

Table of Contents

1. Get started quickly

Make sure you have placed the asset in your project before starting.

1. Navigate to the menu item “Tool/DTT/COLOR/Color Palette Window” at the top of your screen. This opens the window where you can view all your colors and palettes.
2. At the bottom of the window, you will see an input field where you can write your name for the palette you want to add. Once you’ve specified this, you can press the “Add” button next to it, so it’s added.
3. Now you can add colors to your palette: fill in the input field below your palette with the name of your color and the color value. This can be edited after it’s created.

You’re now ready to make use of this color in your project. You can either:

- Add the Color Matcher component to an **Image** or **Text** component in your UI, or add it to a **SpriteRenderer**. You’re then able to select your color using the menu in the inspector of the component. Now the color will automatically be applied.
- Optionally, you can press the save button in the editor window to save your colors to a script file. This will allow you to make use of your colors in your C# script files. You can do so by writing the name of your palette followed by a dot and then the name of your color. So if you have a palette named: “My Palette” and a color named: “Ocean Blue”. You can access this in code by calling: “MyPalette.OceanBlue”.

2. Introduction

DTT COLOR helps the developer by allowing them to categorize and name their colors and palettes. This enables you to consistently make use of colors in your project, and making changes to said colors will then be applied everywhere in the project.

We've designed this workflow boost to take the burden off the developers, who aren't as concerned about the specifics of the designs.

The asset also gives you the opportunity to export all your colors and palettes to a script file that can be used to reference the colors within your code and have compile-time checking.

3. Set-Up

When the asset is placed in your project, it will automatically generate a **Palette Database** asset at “Assets/DTT/COLOR/Resources”. This is the main storage location for all your palettes and colors.

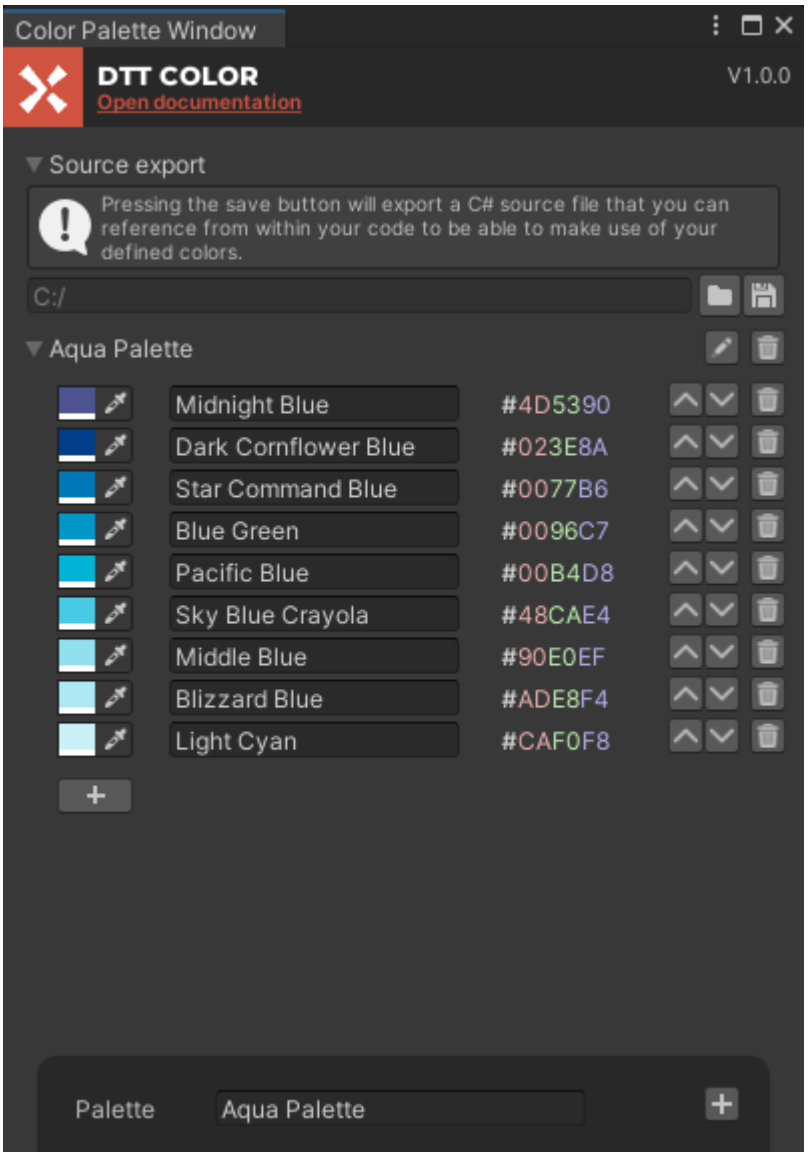
It isn’t intended to modify this asset file directly, but instead you should open its editor window. The location for this window can be found by either pressing the “Open window” button in the inspector of the Palette Database asset, or by navigating to “Tools/DTT/COLOR/Color Palette Window” menu item.

From this window, you’re able to add new palette’s and add colors to them.

4. Editor

This section will go into the different options of the editor windows and inspectors and will explain how they work and what information they show.

Color Palette Window



Source Export

This shows the directory where the source file will be exported to when the save button is pressed. By default, is this the assets' folder of your project, but you're able to customize as you see fit.

Palettes

At the bottom of the window, you will find the section to be able to add a new palette to your project. Fill in the desired name of your palette and hit the plus button to add it.

Every palette is shown as a foldout with the name of your palette (in this case, the name is “Aqua Palette”). The palette section lists all the colors in your palette. You’re also able to delete your palette, by pressing the icon with the trash bin on it. Additionally, you can press the edit button to rename your palette. When renaming your palette, make sure to also update your references in the code since now a different name will be used, but your console will inform you with where these errors are.

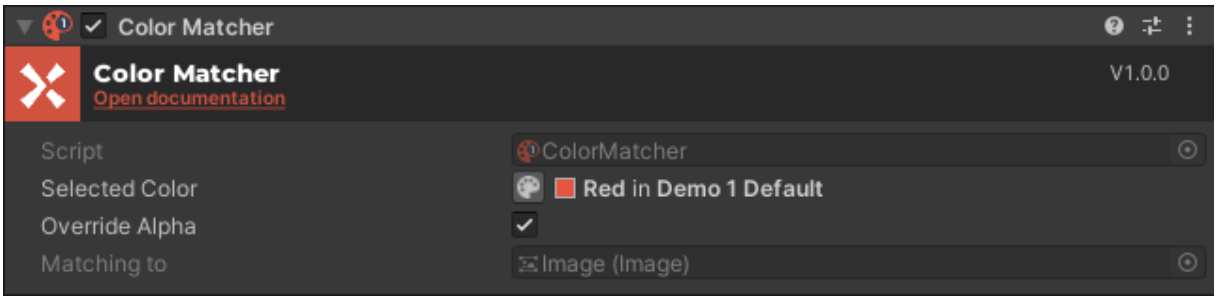
You can add a new color by filling in the fields of the “New Color” input at the bottom of the palette section. When this is ready to be added, you can hit the plus button to add it.

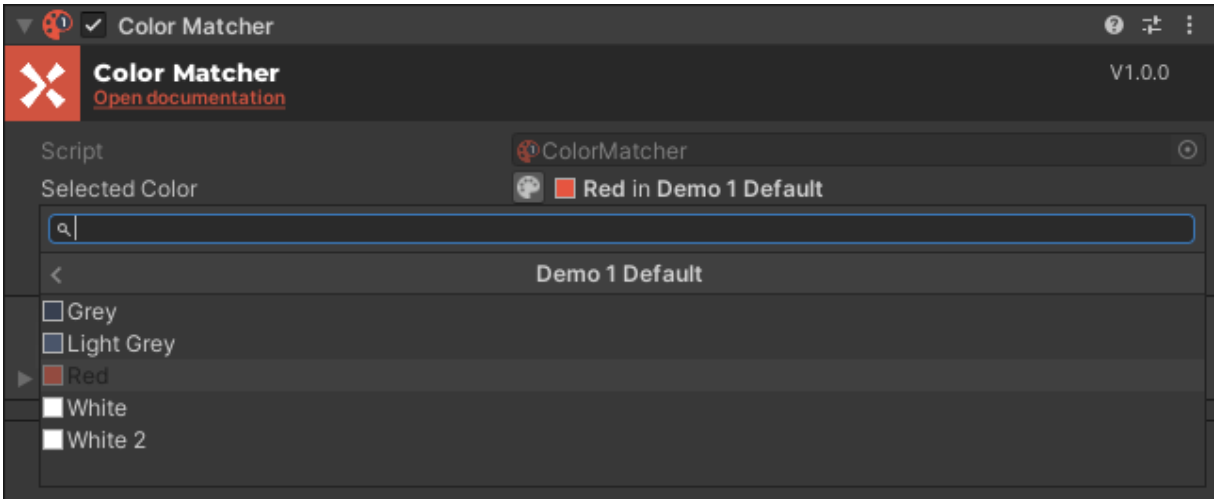
Color Field

Every color has a few options to customize as you see fit:

- The actual color value, this is what is used when referencing this color.
- The name of the color, this is used to quickly identify and separate the color, but also for naming the variables when this is exported to a C# source file.
- The hex value is used to show you what the hex value of your color is and quickly identify the different RGB channels using the colors.
- The delete button is there for when you want to remove a color that won’t need any more.

Color Matcher



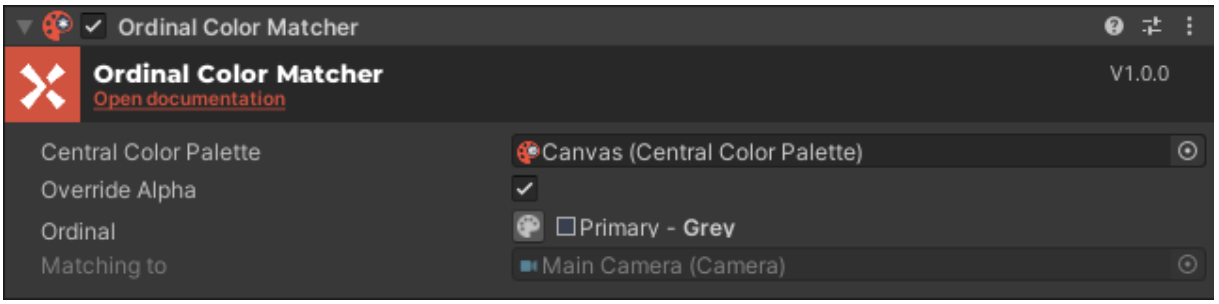


The Color Matcher component allows you to apply the colors from your palettes to UI elements and **Sprite Renderers**. This makes it easier to update all your styling consistently.

Selected Color

This field displays the currently selected color and gives you the option to select a new one. By pressing the button with palette icon, a small window will appear that enables you to choose one of your created palettes. Once your desired palette is selected, you're able to choose the color you want to apply.

Ordinal Color Matcher



The Ordinal Color Matcher allows you to automatically match a color, but do this based on a centralized palette and the ordinality (primary/secondary/tertiary/etc.) This allows you to easily switch between palettes and all your UI elements will follow accordingly.

Central Color Palette

This is the component on which you define the central palette you want to use. The Ordinal Color Matcher listens to this component for what palette it should use.

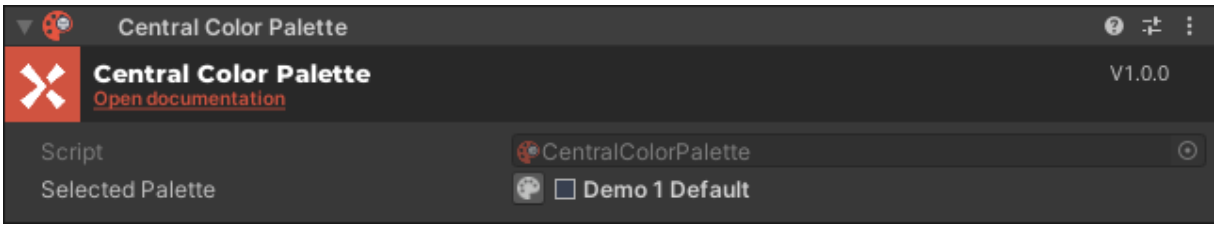
Override Alpha

Disable this toggle to keep your alpha value on the component you're matching a color to.

Ordinal

Here you can select the ordinal that should be used. It will display the current value of the ordinal so you can easily select what you need.

Central Color Matcher



Selected Palette

Here you can select the palette that should be used for all the **Ordinal Color Matcher**.

5. API

PaletteDatabase

This is the main storage point for all palettes and colors. This class inherits from ScribtableObject and as such is serializable in your project. It is expected that one instance always exists in "Assets/DTT/COLOR/Resources", and this should be the only instance that should be used.

The class itself allows for read-only operations on the palettes it composes.

Property Name	Type	Description
Instance	PaletteDatabase	This is a static getter to get the main instance of the ScriptableObject.
Palettes	ReadOnlyCollection<Palette>	All the palettes in the database. These are only read-only, so it can't be removed or added to. These operations should be done from the editor window.

Method name	Parameters	Description
GetColor	Guid paletteGuid Guid paletteColorGuid	Allows you to retrieve a color based on the unique identifiers of the palette and palette color.
GetPalette	Guid paletteGuid	Allows you to retrieve a color based on the unique identifier of the palette.

ColorMatcher

This workflow component allows for saving a reference to a palette color and applying that to the graphical component that is on the same GameObject.

Property Name	Type	Description
Selection	PaletteColorSelection	The currently selected palette color that this matcher uses.

PaletteColorSelection

This class serializes a reference to a palette and a color from the PaletteDatabase. This can be used to easily select colors in the inspector.

Property Name	Type	Description
PaletteColor	PaletteColor	The selected color. If it doesn't exist, it will return null. Except for Guid.Empty, which will return a default blank color.
Palette	Palette	The selected palette. If it doesn't exist, it will return null. Except for Guid.Empty, which will return a default palette.

Palette

Contains the data required for existing as a palette, this means a name and a collection of Palette Colors.

Property Name	Type	Description
GUID	Guid	The unique identifier of this palette.
Name	string	The name of this palette.

Count	int	The amount of colors that are in this palette.
Indexer [int index]	PaletteColor	The indexer used to get PaletteColors by using an index.

Method name	Parameters	Description
Contains	PaletteColor item	Checks if the item passed is contained in the current collection. Returns true if it is, returns false if it isn't.

PaletteColor

This is the actual color reference that is saved in the project. Since the normal UnityEngine.Color is not a reference type this can't actually be saved. That's why we wrap it in a class called PaletteColor, this is something we can save. And in this class we can add additional data, like a name and GUID for being able to identify.

Property Name	Type	Description
GUID	Guid	The unique identifier of this color.
Color	Color	The color used in this PaletteColor.
Name	string	The name of this color.

6. Known Limitations

- None.

7. Support and feedback

If you have any questions regarding the use of this asset, we are happy to help you out. Always feel free to contact us at:

unity-support@d-tt.nl

(We typically respond within 1-2 business days)

We are actively developing this asset, with many future updates and extensions already planned. We are eager to include feedback from our users in future updates, be they 'quality of life' improvements, new features, bug fixes or anything else that can help you improve your experience with this asset. You can reach us at the email above.

Reviews and ratings are very much appreciated as they help us raise awareness and to improve our assets.

DTT stands for Doing Things Together

DTT is an app, web and game development agency based in the centre of Amsterdam. Established in 2010, DTT has over a decade of experience in mobile, game, and web based technology.

Our game department primarily works in Unity where we put significant emphasis on the development of internal packages, allowing us to efficiently reuse code between projects. To support the Unity community, we are publishing a selection of our internal packages on the Asset Store, including this one.

More information about DTT (including our clients, projects and vacancies) can be found here:

https://www.d-tt.nl/en/