Pretty Console 1.0.2

Pretty Console is a tool that allows you to create colorful and more distinguishable debug logs in the Unity console. Not only does it make it more presentable for screenshots - but easily readable and more pleasing to the eye. The asset also automatically displays the name of the class or method at the start of the log which helps with debugging. If you would like to suggest features please e-mail me at: troy@helloworldstudios.co.uk

1. Console Methods

To begin, Implement the *PrettyConsole* namespace at the top of your Script.

```
using PrettyConsole;
```

1a. Log

To create a simply log use the following method.

```
Console.Log("This is a normal log.");
```

You can add a simple parameter in the form of either a Color, *FontStyle* or *PrefixStyle*.
Note: The values of *PrefixType* are either *ClassName*, *MethodName* or *NoPrefix*.

```
Console.Log("This is a color log.", Color.cyan);
Console.Log("This is a font style log.", FontStyle.Italic);
Console.Log("And this is a custom prefix type log.", PrefixType.MethodName);
```

You can pass three parameters too, if you want a value to remain as default pass null.

```
Console.Log("Bold magenta log", Color.magenta, FontStyle.Bold, PrefixType.NoPrefix);
```

Logging warnings and errors works the same too.

```
Console.LogWarning("Warning log.", PrefixType.NoPrefix);
Console.LogError("Error log.", FontStyle.BoldAndItalic);
```

1b. String

If you click on the Log in the Unity Console it will open up the Pretty Console script. If you want it to open up your script (where the log is called) you can use *Console.String* instead. This works just like the Log methods and includes *Console.StringWarning* and *Console.StringError*. Unlike *Log* it returns a string variable that then you can log yourself.

```
Debug.Log(Console.String("Normal string log."));
Debug.LogWarning(Console.StringWarning("Warning string log."));
```

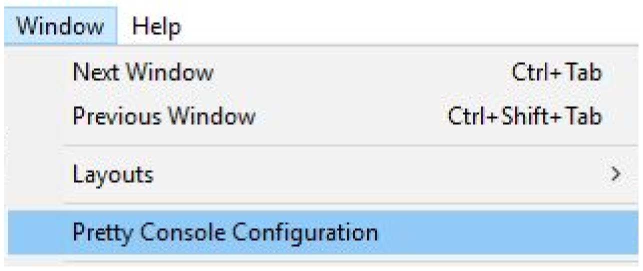
1c. Quotes

Surrounding a section of the string in the defined quote character (apostrophed by default) will change the color of that section in the console window.

```
Console.Log("Normal color 'quote color' normal color.");
```

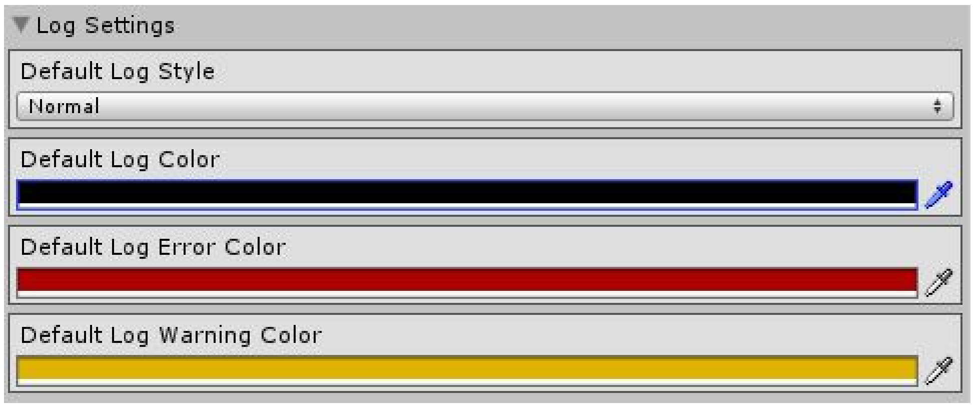
2. Configuration Window

The Pretty Console Configuration Window can opened by going to *Window > Pretty Console Configuration*.



2a. Log Settings

The log settings allow you to customize the default color and font style of logs. This means if no color is given as a parameter when a Console method is called, the log will be the colors below. The same principle goes for the the default font style.



Log Settings

Default Log Style: Normal

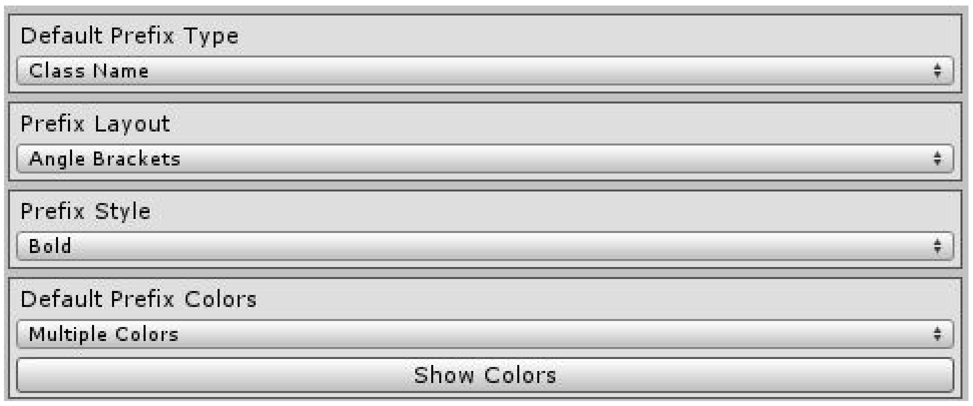
Default Log Color

Default Log Error Color

Default Log Warning Color

2b. Prefix Settings

The Prefix settings allow you to customize the prefixes and default prefix settings. The *Default Prefix Type* will be change what the prefix says if the log hasn't been called with a *PrefixType* parameter (Either method or class name). The *Prefix Layout* and *Prefix Style* will apply to all log prefixes. *Default Prefix Colors* allows you to choose one or many colors that will be assigned to prefixes on runtime (if they haven't been assigned a color). Press the *Show Colors* button to bring up an editable list of colors if *Default Prefix Colors* is set to *Multiple Colors*.



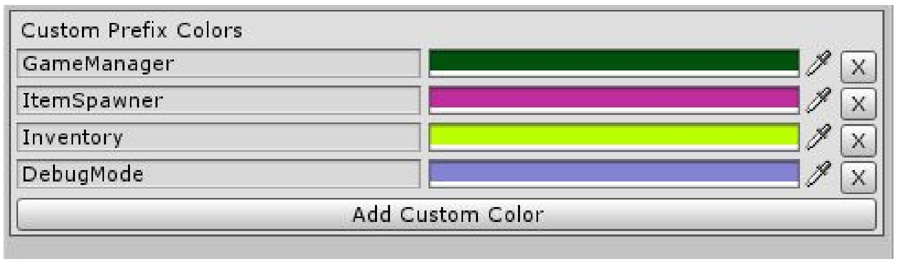
Default Prefix Type: Class Name

Prefix Layout: Angle Brackets

Prefix Style: Bold

Default Prefix Colors: Multiple Colors

Custom Prefix Colors allow you to assign a color of your choice to a prefix title (either a method or class name depending on the *MethodType*). For example, looking at the picture below we know that any log from the Game Manager class will be in dark green while logs from the DebugMode class will be in blue.

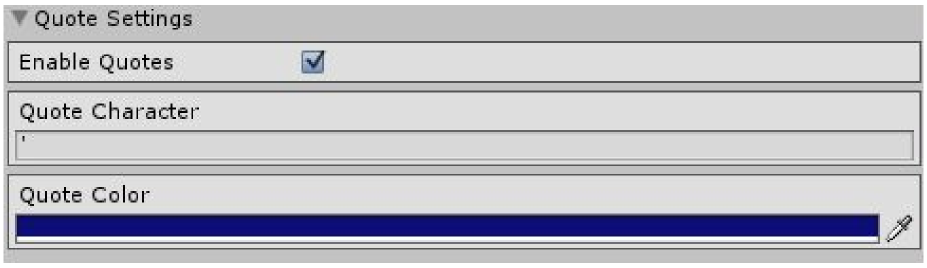


Custom Prefix Colors

GameManager	🟩	X
ItemSpawner	🟪	X
Inventory	🟨	X
DebugMode	🟦	X

2c. Quote Settings

The Quote Settings allow you to customize the character that starts and ends quotes (apostrophe by default) and also change the color of the text inside the quote. You can also disable quotes if you don't want the color to change.



Quote Settings

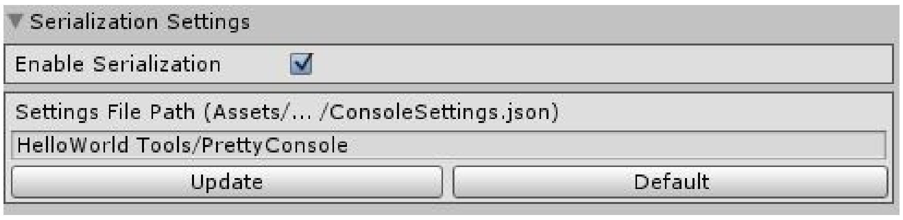
Enable Quotes ✓

Quote Character
'

Quote Color 🟦

2d. Serialization Settings

Serialization Settings lets you customize the location of the JSON file that contains the user's custom settings. You can change the location of where the file is saved to fit your project's file structure. You can also disable serialization but this is not recommended as all the settings will remain as default until you re-enable it. If you have Serialization disabled you can open the *PrettyConsole.cs* file and find the *PrettyConsoleSettings* class and edit the values there. Remember to press the *Update* button before navigating away from this tab and press *Default* for the path to return back to the one shown below.



Serialization Settings

Enable Serialization ✓

Settings File Path (Assets/... /ConsoleSettings.json)
HelloWorld Tools/PrettyConsole

3. Example Script

Inside the *Pretty Console* folder there is an Example Script, add the script to an active Game Object then play the scene for an example of some of the logs that you can create.