

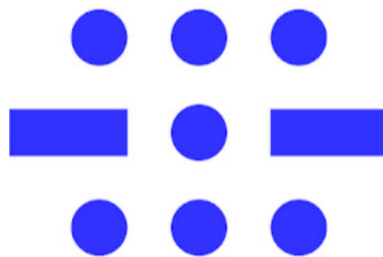
Easy UI Color Picker User Guide

Version 1.2

An asset for Unity

2019 © by
Stefan Scholl

<https://www.stefan-scholl.net/>



Stefan Scholl Software Development

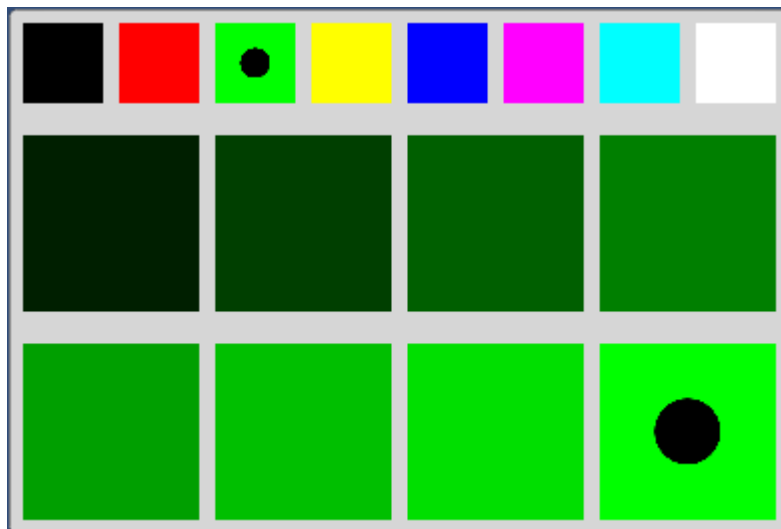
Content

Intro.....	3
Usage.....	4
Activated by script (switch on/off).....	4
Already active.....	5
Callback.....	6
API.....	7
Properties.....	7
Methods.....	7
Changelog.....	9
Version 1.2.....	9
Version 1.1.....	9

Intro









Welcome to the wonderful world of selecting one color from a small subset of colors.

Easy UI Color Picker has a very simple API. Only one or two methods out of Show, Close, and Setup are needed. Set a callback in the inspector or by code. That's it.



After selecting one of the 8 base colors (incl. black and white) you are presented by 8 shades of this color. These shades are the main colors that trigger a callback when selected.

The 8 base colors:

#000000	#FF0000	#00FF00	#FFFF00	#0000FF	#FF00FF	#00FFFF	#FFFFFF
							

Feel free to examine the three example scenes:

- 01-One-Object-One-Color
- 02-Ticker-Two-Colors
- 03-Three-Objects-Three-Colors-Always-Visible

Usage

Folder EasyUIColorPicker/Prefabs contains the prefab **EasyUIColorPicker**. Drag it into an existent Canvas in the hierarchy.

In your scripts you need to access the EasyUIColorPickerController of the prefab. It's in the namespace EasyUIColorPicker. Don't forget the

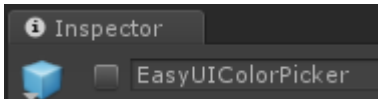
```
using EasyUIColorPicker;
```

directive.

You can use the prefab in two ways:

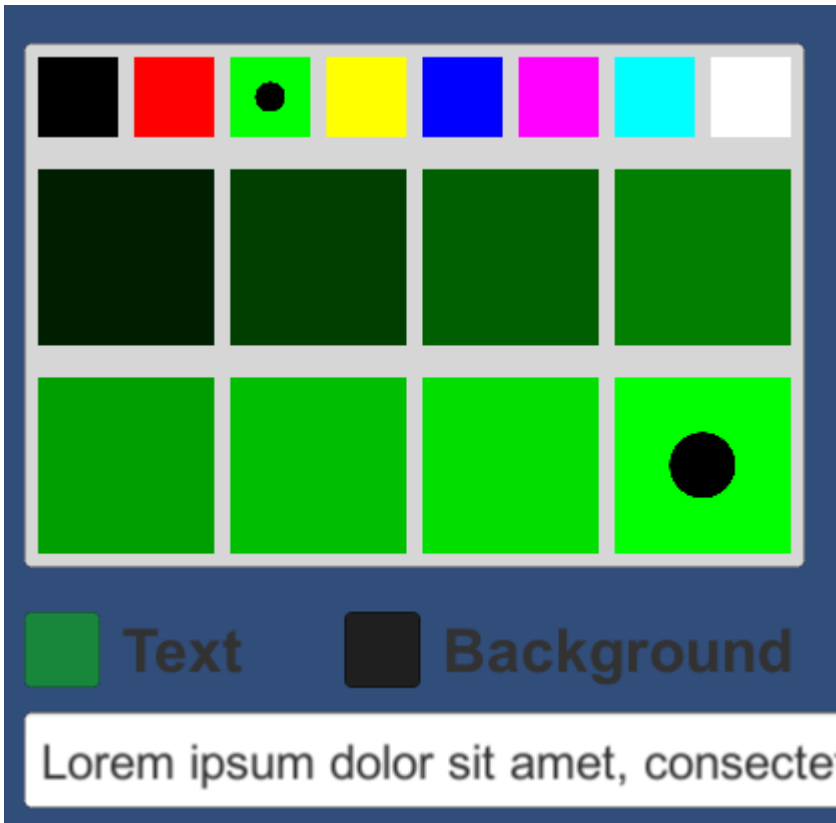
1. Activated by script (switch on/off).
2. Already active, setup by script.

Activated by script (switch on/off)



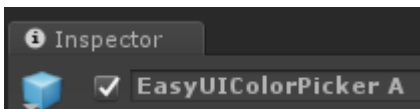
The two example scenes 01-One-Object-One-Color and 02-Ticker-Two-Colors call EasyUIColorPickerController's method Show to activate the game object and preselect a color.

```
public void TextColorButtonClicked()  
{  
    TextColorButton.interactable = false;  
    TextColorPicker.Show(_textColorImage.color);  
}
```



The color picker then is visible and the supplied color gets preselected. If the supplied color isn't in the subset used by the color picker, the nearest color is selected.

Already active



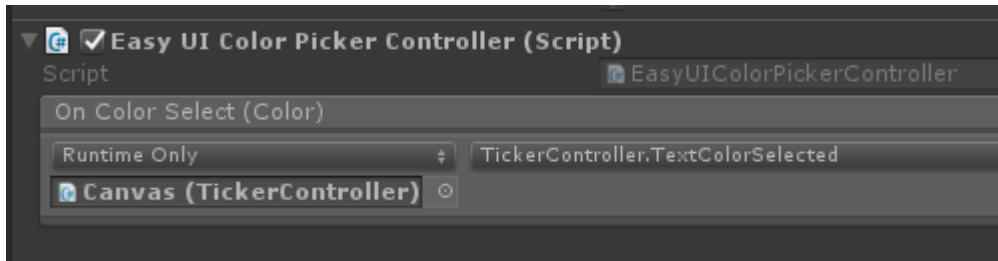
The scene 03-Three-Objects-Three-Colors-Always-Visible uses the method `Setup` to prepare the already visible color picker.

```
private void Start()
{
    _mat = GetComponent<Renderer>().material;
    [...]
    EasyUIColorPicker.Setup(_mat.color);
}
```

Callback

You could check if the color property of the `EasyUIColorPickerController` has changed but using a callback is more sensible.

You can set the `OnColorSelect` callback directly in the inspector:



Or via code (see example code `BallColor.cs` from scene `03-Three-Objects-Three-Colors-Always-Visible`):

```
private void Start()
{
    [...]
    EasyUIColorPicker.OnColorSelect.AddListener(ColorSelected);
    [...]
}

public void ColorSelected(Color color)
{
    _mat.color = color;
}
```

If you need a feedback for the selected base color you can use the `OnBaseColorSelect` callback the same way.

API

Properties

`public bool interactable`

Enable or disable the color selector. Defaults to true, which means the color selector is enabled. Disabled it appears grayed out.

`public Color color`

The last selected color. Default (before Show or Setup) is `Color.black`.

`public Color baseColor`

The last selected base color. Default (before Show or Setup) is `Color.black`.

Is available even before the main color is selected.

`public EasyUIColorPickerEvent OnColorSelect`

The callback. Provide a method with a signature like

```
void Callback(Color color)
```

It gets called when the main color is selected.

`public EasyUIColorPickerEvent OnBaseColorSelect`

The callback. Provide a method with a signature like

```
void Callback(Color color)
```

It gets called when the base color is selected.

Methods

`public void Show(Color initColor)`

Displays the color selector with a preselected color `initColor`. The nearest (RGB) color is used as Easy UI Color Picker doesn't provide all available colors.

`public void Close()`

Closes the color selector.

`public void Setup(Color initColor)`

Selects the color `initColor` and initializes the Easy UI Color Picker if needed. The nearest (RGB) color is used as Easy UI Color Picker doesn't provide all available colors.

Changelog

Version 1.2

- Documentation comments for API code.
- Boolean `interactable` to disable the color selector.
- Example 03-Three-Objects-Three-Colors-Always-Visible now disables all other color selectors when you select a base color in one of them. Enables them again when you select the color.

Version 1.1

- Namespace for examples changed from `EasyUIColorPicker` to `Examples_EasyUIColorPicker`. The examples now have “using `EasyUIColorPicker` ;” and are easier to understand for new users to C#.
- Property `baseColor` to access the base color as soon as it is selected.
- Callback `OnBaseColorSelect` gets called when a base color gets selected.