

Color Picker Quick start guide.



Really quick start.

1. Create new scene.
2. Add new Canvas.
3. Attach ColorButton prefab to your Canvas (located in Assets\GravityBox\ColorPicker folder).
4. Configure Color, Alpha and HDR options.
5. Launch a Game and click that button!

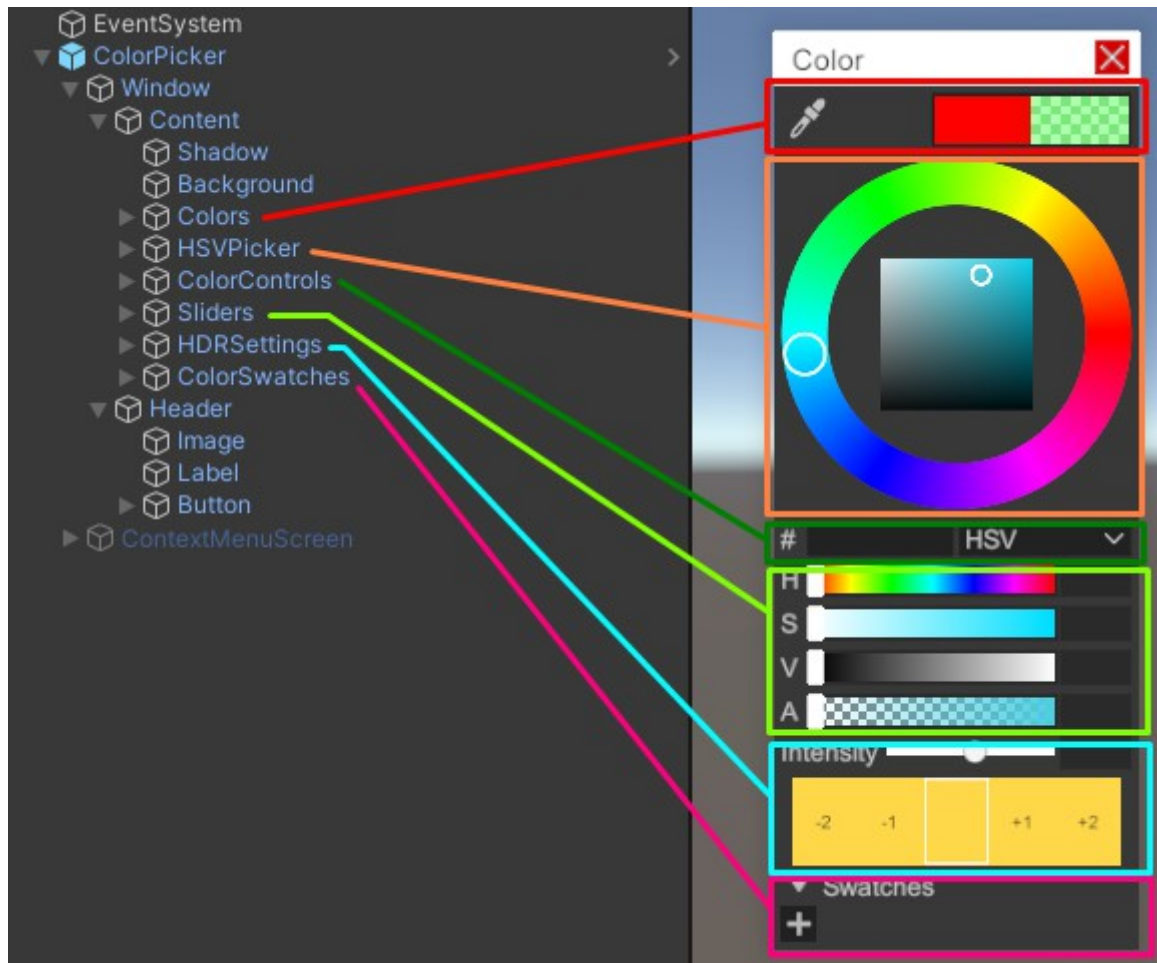
Step by step and in details.

First thing to notice is that using **ColorButton** in your project is completely optional and you can use whatever approach that suits your project. In the end its just a matter of Instantiating (or activating) color picker in your scene. So in order for it to work you just need a single **ColorPicker** prefab located in the same location as that color button.



GRAVITY
BOX
STUDIO

Anatomy of Color Picker



Color Picker look and works pretty similar to built in Unity Editor's own picker. It works out of the box and with idea that you don't need to setup anything for it to work. But it's better to have an idea of inner architecture and how it's interconnected. Picker consists from few key control parts you need to know about.

1. **ColorPicker.** Holds **ColorPickerWindow** – main window management script and **CloseWindowOnClick** script which does that only one thing – closes window when clicked outside of picker window.
2. **Colors.** No special scripts used except a *Button* that starts a screen pixel picking process and a *Button* to reset colour back to old version.
3. **HSV Picker.** Consists of three elements: Hue ring gradient and Saturation/Value box gradient both of which make use of **ColorPickerHSV** script controlling logic behind selecting those colors. Third control is a hidden *RawImage* displaying zoomed in part of the screen when picking pixel from screen.
4. **Color Controls.** Hexadecimal *InputField* and colour mode *Dropdown* switches sliders below between HSV, RGB 0-255, RGB 0-1 modes.
5. **Sliders.** Make use of two scripts **ColorPickerSlider** and **SliderValue**. First one used for managing slider looks, logic and values, second – compound script for UI slider and Input field to play together nicely.
6. **HDR Settings.** Uses **ColorPickerHDRSettings** script used to manage HDR controls. Also uses **SliderValue** script for slider.
7. **Color Swatches.** Loads/Saves and manages color presets (or swatches) by **ColorPickerSwatches** script. Contains Toggle control for folding out a list of items and an item template with **ColorPickerSwatchesItem** script used to hold color settings info.

Special Mention goes to **ColorPickerCurrentColor** ScriptableObject which simply put is a Color conversion/EventDispatcher hybrid object which receives color value from current input (Slider/InputField/Preset or even screen) converts it and dispatches events to other controls to keep all of them in sync. Also holds a latest color from active picker in various formats including its HDR values. **ColorPickerManager** ScriptableObject is completely optional to use but holds a working example of code used to show ColorPicker on the screen. Used by Color Button.

Anatomy of Color Button

Though completely optional to use ColorButton may be a good quick starting example completely replicating similar controls from Unity Editor itself.



Uses two small scripts:

- **ColorButton**. Management of Button looks and logic. Like calling Picker window or starting quick screen color capture process.
- **ColorImage**. Graphic component used to generate mesh for **color/alpha/hdr** indicator part of a control.

Thats about it considering Color Picker quick start. In case of questions, info or ideas consider contacting me via old trusty

email: defaxer@mail.com

twitter(X): [@defaxer](https://twitter.com/defaxer)

or even [Facebook](https://www.facebook.com/defaxer)

