

# OnGUI ColorPicker readme2

- [Usages](#)
  - [Simple Usages](#)
  - [Using custom styles.](#)
  - [Get/Set value](#)
- [Demos](#)
- [Contacts](#)

## Usages

### Simple Usages

```
public class TmpForScreenshot : MonoBehaviour {
    ColorPicker colorPicker=new ColorPicker();

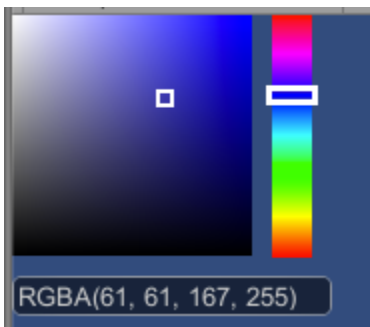
    Rect paletteRect = new Rect(0 , 0 , 120, 120);
    Rect sliderRect  = new Rect(130, 0 , 20 , 120);
    Rect textRect    = new Rect(0 , 130, 160, 20 );

    Color32 colorValue;

    void Start () {
        colorPicker.initialize(paletteRect,sliderRect); // initialize new ColorPicker with
        palette and slider position
    }

    void OnGUI () {
        colorValue = colorPicker.OnGUI(); //render colorPicker and take active color value
        GUI.TextField(textRect, colorValue.ToString()); //just show value here
    }
}
```

You will see on the screen this objects



### Using custom styles.

ColorPicker support custom styles of pointer, you can use your own textures. for that.

you can find example in **ModalColorPickerCustomStyles** class.

On initialization you need to provide 2 more option (both of them are optional, if you miss anyone it will take default textures)

```
colorPicker.initialize(palleteRect, sliderRect, activeColorPointerTexture,  
sliderThumbStyle);
```

**activeColorPointerTexture** - is a texture, will be good if it will have odd number of pixels, like 15x15 or 27x27, so center pixel will exactly point to selected color.

**sliderThumbStyle** - is a style, in style you need to fill following properties

- normal.background.texture (your slider texture, and again. will be good if it will have height wit odd number of pixels)
- border (simply it's a stripes on the edge of your texture which not be stretched on resize of your texture)
- overflow - if you want your thumb slider bigger than hueRect, this value can be positive, otherwise negative
- alignment = TextAnchor.MiddleCenter;
- fixedHeight = height of your thumb, good if it will have the same value as your texture.height

## Get/Set value

to get selected color, you can use following methods

```
HSVColor hsv= colorPicker.getHSV();  
Color32 rgb = colorPicker.getRGB();
```

to set color to colorPicker you can use any of this method (set color is a alias of setRGBColor)

```
colorPicker.setColor(Color32 color);  
colorPicker.setRGBColor(Color32 color);()
```

## Demos

In the package you will find demo scene (Assets/ColorPicker/Demo)

With following scripts in (Assets/ColorPicker/Demo/Scripts) folder

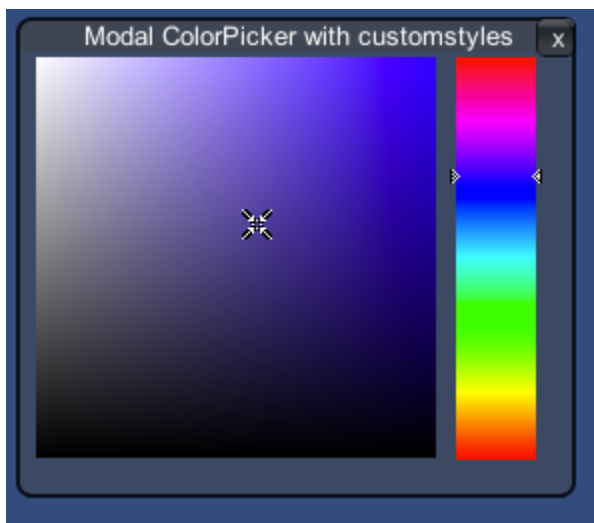
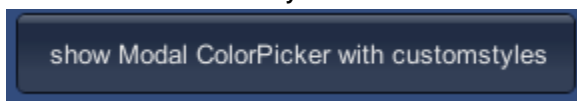
- **ScaleSetup.cs** - this script show settings to which used in [GUI.matrix](#) method in another scripts



- **FloatingColorPicker.cs** - this window contains color picker, RGB, HSV values, with options to change selected colors using sliders



- **GetValueFromColorPicker.cs** - scripts described above in this document and not used on the scene
- **ModalColorPickerCustomStyles.cs** - Modal window with colorpicker with custom styles , which by default hided and showed as button



- **SimpleColorPicker.cs** - simple script which just render colorpicker

## Contacts

if you will have any questions, please [use this thread](#) on the unity3d forum

or can contact me by email: [nicloay@gmail.com](mailto:nicloay@gmail.com)