



--Overview--

RTColorPicker is a run time full featured color picker for the Unity3D Game Engine. It is designed to fit nicely with new and existing game/application projects. Whether you are making a level editor or need color selection for character customization, RTColorPicker can be a good fit for your project.

There are various configuration options to alter the appearance and functionality of the RTColorPicker. Slider sections, buttons and other features can be easily turned off and on. Need just a simply Alpha slider? Simply turn off everything except for the alpha slider and you've got a scaled down alpha color picker.

The RTColorPicker does not use Unity's readpixels to determine colors using the color selection area or the sliders, it's just good old fashion math. Readpixels is only used when using the eyedropper/sample color option to generate the effect of zooming in to an area.

--Features--

- HSV, RGBA and Hex input
- Color selection box and wheel
- Eye Dropper/Sample Color tool allows for quick and easy scene color selection
- Vertical and Horizontal orientation modes

- Includes custom skin (skin can be changed)
- Old Color and New Color
- Lots of configuration options displaying just the options you need
- Colors panel which can configurable default colors and user defined custom colors
- Save and Load user defined custom colors through easy to use function callbacks
- Separate scripts for C# and JavaScript compatibility

--Demo Scene--

The Demo scenes included in the Unity Package show some example usage of the RTColorPicker. RTColorPickerScene and RTColorPickerScene_CS are the same except that RTColorPickerScene_CS uses C# scripts for both the ColorPicker and the DemoUI.

The DemoUI and DemoUI_CS files show the various ways we can use the RTColorPicker as well as example usage on how to Save/Load custom colors from an external file.

If using C# please refer to the DemoUI_CS.cs file to see how to use.

--Adding the RTColorPicker To Your Project--

After importing the RTColorPicker Unity Package...

JS:

Drag and drop the RTColorPicker_Prefab object under the RTColorPicker folder to your scene.

C#:

Drag and drop the RTColorPickerCS_Prefab object under the RTColorPicker folder to your scene.

The prefab contains a GameObject with the RTColorPicker script attached and setup. It is recommended to remove the RTColorPicker files related to the language you are not using. For example if you are only using the C# version then remove the JS related scripts under RTColorPicker/Scripts/JS. This will help avoid any naming conflicts that may arise.

Clicking the prefab in the scene you'll notice there are a lot of inspector options. We'll cover these below.

--Using the RTColorPicker--

Once the RTColorPicker Prefab has been added to your scene you will want to access it from your existing script(s). Simply add a public variable to your script for the RTColorPicker

JS:

```
var ColorPicker : RTColorPicker;
```

C#

```
public RTColorPicker ColorPicker;
```

Drag and drop the RTColorPicker from the scene on to your variable in the Unity Inspector panel. You'll now be able to access the RTColorPicker and start picking colors. Please refer to the DemoUI.js or DemoUI_CS.cs script for sample usage.

NOTE: You only need one RTColorPicker Prefab in your scene. This will allow for usage for multiple colors as seen in the DemoUI.js script. Should you need two color pickers open at the same time then you'll need to add an RTColorPicker Prefab more than once.

--RTColorPicker Public Variables--

Skin

Type: GUISkin

Default: ColorPickerSkin

This is the gui skin used by the RTColorPicker. This can be swapped out with a different gui skin.

WindowTitle

Type: String

Default: "RTColorPicker"

This is the text that appears on the top header of the color picker window.

ShowColorSelection

Type: boolean

Default: true

Toggles visibility of the ColorWheel/ColorBox area.

ShowHSVGroup

Type: boolean

Default: true

Toggles visibility of the HSV slider controls

ShowRGBGroup

Type: boolean

Default: true

Toggles visibility of the RGB slider group (does not include alpha slider)

ShowAlphaSlider

Type: boolean

Default: true

Toggles visibility of the Alpha slider

ShowHexField

Type: boolean

Default: true

Toggles visibility of the Hex Value input area

AllowWindowDrag

Type: boolean

Default: true

Enables/Disables ability to drag color picker window

AllowEyeDropper

Type: boolean

Default: true

Toggles visibility of the Eye Dropper/Sample Color button

AllowWheelBoxToggle

Type: boolean

Default: true

Toggles visibility of the Wheel/Box selection buttons which are used to toggle between wheel and box color selection modes

AllowOrientationChange

Type: boolean

Default: true

Toggles visibility of the Orientation button which is used to toggle between vertical and horizontal display modes

AllowColorPanel

Type: boolean

Default: true

Toggles visibility of the Color Panel button which is used to toggle visibility of the Default/Custom Colors Panel

AllowCustomColors

Type: boolean

Default: true

If false custom colors will not be displayed or usable in the Colors Panel area

DefaultColorsFile

Type: TextAsset

Default: DefaultColors (located under Demo Assets)

This is a text asset object that defines what colors should appear for the Default Colors in the Colors Panel. Colors in the text asset should be in hex colors and should be one color per line. Open the DefaultColors file located under the Demo Assets folder for example usage.

MaxCustomColors

Type: int

Default 24

The maximum allowable custom colors that can be added in the Colors Panel. Does not effect Default Colors.

ColorWheelBtnStyle

Type: GUIStyle

Default: N/A

The button style for the Color Picker Wheel toggle button

ColorBoxBtnStyle

Type: GUIStyle

Default: N/A

The button style for the Color Picker Box toggle button

EyeDropperBtnStyle

Type: GUIStyle

Default: N/A

The button style for the Eyedropper/Color Sample button

OrientationBtnStyle

Type: GUIStyle

Default: N/A

The button style for the Orientation Toggle button

ColorPanelBtnStyle

Type: GUIStyle

Default: N/A

The button style for toggling the Color Panel open/closed

GroupToggleBtnStyle

Type: GUIStyle

Default: N/A

The button style for the HSV/RGB/Alpha slider groups header

ColorBtnStyle

Type: GUIStyle

Default: N/A

The button style for the colors displayed in the Colors Panel
for default/custom colors

--RTColorPicker Functions--

If a function is specific to a certain language it is denoted before the function definition below. If there is no language specified it is the same for either JS or C#.

Show(c : RGBColor)

Make the colorpicker visible and defines the passed in RGBColor as the target for color selection.

JS: Show(c : Color, tagStr : String, changeCallback : Function)

Function shows the color picker and sets unity color as active target.

The tagStr is returned in the changeCallback function so that multiple colors can be handled from one RTColorPicker. The changeCallback function is called everytime the color changes in the color picker. See DemoUI.js for example usage.

C#: Show(c : Color, tagStr : String)

Function shows the color picker and sets unity color as active target. The tagStr is returned in the changeCallback function so that multiple colors can be handled from one RTColorPicker. The changeCallback function is called everytime the color changes in the color picker. See DemoUI.js for example usage.

Show(m : Material)

Shows the color picker and set's the passed in material as the target for color selection. The materials main color will automatically be used by the color picker.

Show(l : Light)

Shows the color picker and set's the passed in Light's color value as the target for color selection. The Alpha value of the color will automatically be used to adjust the lights intensity.

Show(c : Camera)

Shows the color pcker and set's the passed in Camera's background color value as the target for color selection

Hide()

Hides the color picker

IsOpen()

Returns true if the color picker window is currently visible.

GetCurrentRGBColor()

Returns the currently selected color in the color picker as an RGBColor object.

GetCurrentUnityColor()

Returns the currently selected color in the color picker as a standard Unity color

SetPos(x : int, y : int)

Set's the Color Picker window position in screen coordinates

LoadCustomColors(dataStr : String)

Loads custom colors from dataStr. dataStr should contain colors in hex format separated by line breaks.

SetMode(mode : CPMODE)

Set's the current color selection mode: CPMODE.COLORWHEEL

or CPMODE.COLORBOX

JS: SetCallbackFunc(funcName : String, func : Function)

Set's a callback function for the ColorPicker. Passed in function can be null to clear the callback. This function is only applicable if you are using the Javascript version of RTColorPicker. See DemoUI.js for example usage.

FuncName Definitions:

“OnOK”: Function is called when the user presses the Ok button

“OnCancel”: Function is called when the user presses Escape, X or Cancel

“OnCustomColorSave”: Function is called when the custom colors in the Colors Panel has changes and need saving

“OnCustomColorLoaded”: Function is called when custom colors have been loaded successfully