Readme - Advanced Color Picker 1.3.0

A meticulously designed, easy-to-use color pickers asset pack made for Unity3D. We present you a professional looking, accurately calculated UI, yet highly customizable!

No matter what Unity player platform, mobile or desktop applications. This is the color picker package you need!

Index

(1) Features

- Support multiple color pickers with different settings
- Pick color from palette, easy to add/replace palette textures
- Color Dropper : pick color from screen
- Adjust the RGBA sliders to pick color
- RGBA input-fields, set color by entering the RGBA values
- Hex code input-field, convert hex code to color
- Preset Colors, auto/manual manage the preset colors. Easy, flexible, configurable layout
- Ready to use and easy to customize UI design (5 ready to use color picker templates included, and easy to create more on your own)
- Show RGB/RGBA values, show Hex color code, show the captured texture
- Save picked colors as JPG/PNG
- Highly customizable settings and concise API.

Highly cumtomizable

Enable/Disable dragging for the color picker
Enable/Disable alpha setting
Enable/Disable color dropper
Adjustable color dropper size
Flexibly change the palette texture (runtime & editing time)
Auto-fit with screen, also has extra settings for handling panel size
Scalable panel (by setting a scale factor)
Options for auto manage(save/load) preset colors to the PlayerPrefs, or by manual operations

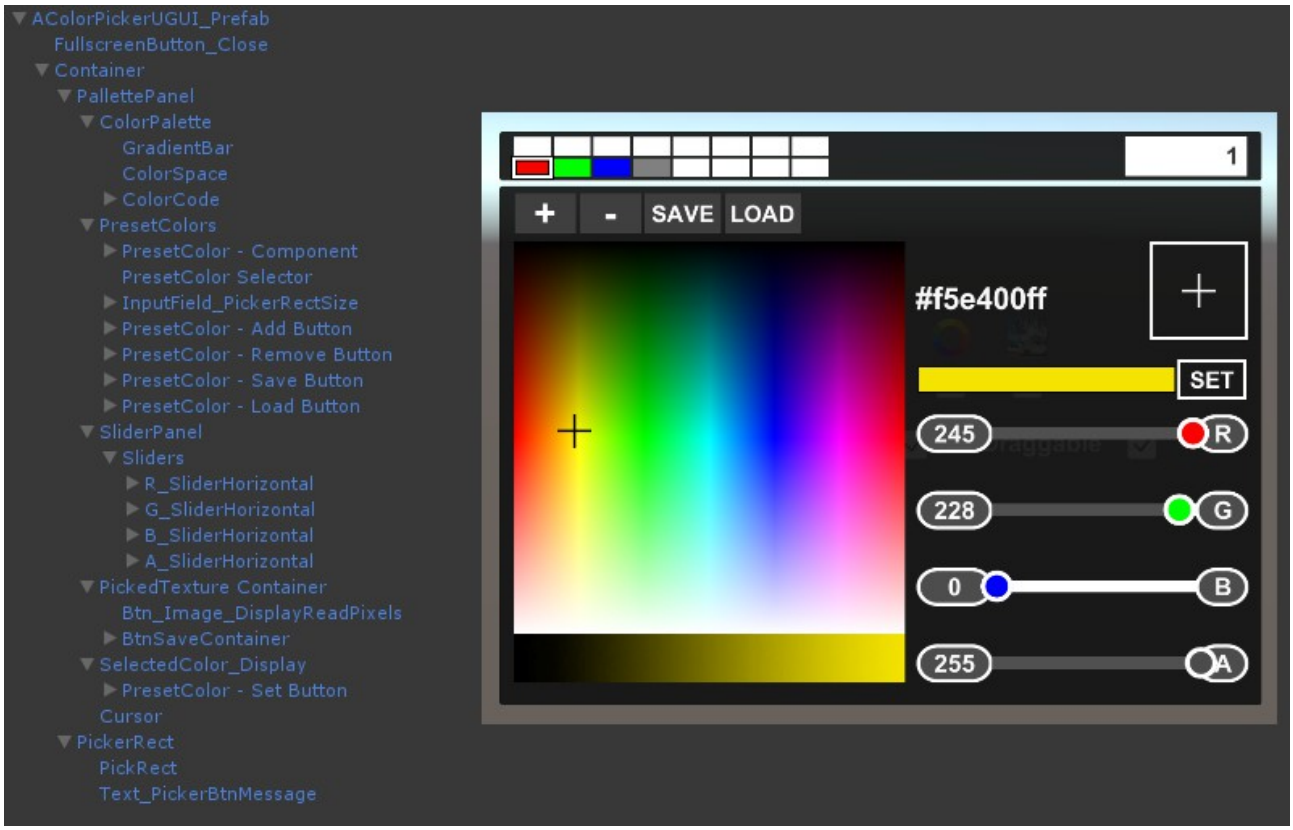
New Unity UI (compatible with other UI systems)
Tested on Unity 5, 2017 & 2018
All platform
Source code
Mobile friendly!

(2) Fast Guide

Demo scenes and carefully designed color picker prefabs are provided in the package.
Try them out and you will know how they work and know how to use the provided scripts to do more you want. The provided scripts and UI are carefully designed to make this package easy to work with, flexible to customize and highly reusable!

Just give some different settings to the provided prefabs to make some more variations. You can change their settings at runtime, or change the settings on the provided prefabs in the Unity inspector.

(3) Prefab Structure



* More color picker prefab templates are provided in the asset, the structure looks similar in the Hierarchy window, but the UI looks so different(please refer to the screenshots on our website or AssetStore page)

(4) Coding Examples

The **AcolorPicker.cs** script holds most of the color picker functions, variables and callbacks. Below are some useful code snippets for easy integrating the color picker in your app. You can also find these examples in the demoscene scripts.

To instantiate a color picker:
AcolorPicker picker = AcolorPicker.Create(string: prefabName);
or
AcolorPicker picker = AcolorPicker.Create(Transform: parentTransform, string: prefabName);

Call the Setup method for some basic setup:
picker.Setup(GradientBarUsage: gradientBarUsage, Action: onCloseAction);
or
picker.Setup(GradientBarUsage: gradientBarUsage, bool: hasPickerRect,
Action: onCloseAction);
or
picker.Setup(GradientBarUsage: gradientBarUsage, int: pickerRectSize,
Action: onCloseAction);
or
picker.Setup(Action: onCloseAction);

Set the callback to receive picked color:
picker.SetOnColorChangeCallback(MethodName_Or_Action);

Set the callback to receive picked texture:
picker.SetOnPickedSampleTextureCallback(MethodName_Or_Action);

Toggle draggable feature:
picker.ToggleDraggable(bool: isDraggable);

Toggle picker rect(dropper) feature:
picker.TogglePickerRect(bool: enableDropper);

Set the panel position:
picker.SetPanelPosition(float: X, float: Y);

Set the picker rect(dropper) position:
picker.SetPickerRectPosition(float: X, float: Y);

Set the color picker scale:
picker.SetContainerScale(float: scaleFactor);

Set the color picker scale by canvas size:
picker.SetScaleByCanvasSize(int: canvasSize);

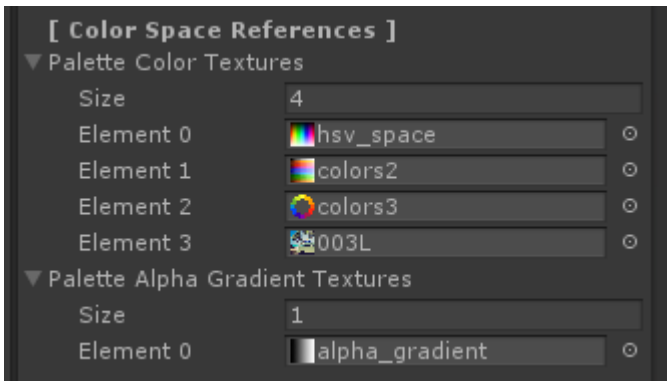
Set the picker rect(dropper) size:
picker.SetPickerRectSize(int: size);

Set the panel width with a constant width value:
Picker.SetPanelWidth(int: constantWidth);

Set the panel width with a fill-rate with respect to the screen width:
Picker.SetPanelWidth(float: fillRate);

Enable or Disable the Gradient bar and Alpha, set to use the gradient bar as Alpha Gradient or Color Gradient:
picker.SetGradientBar(GradientBarUsage: gradientBarUsage);
GradientBarUsage enum options:
Disable: Hide gradient bar, hide alpha settings
AlphaGradient: Show gradient bar, use it as alpha gradient, show alpha settings
ColorGradient: Show gradient bar, use it as color gradient, show alpha settings
ColorGradient_NoAlpha: Show gradient bar, use it as color gradient, hide alpha settings
Disable_HasAlpha: Hide gradient bar, show alpha settings

You can also change the palette textures at runtime.
Below shows the code and prepared texture references in the prefab.



Change the palette color texture (by index):
picker.ChangePaletteColorTexture(int: index);

Change the palette color texture with a new texture:
picker.ChangePaletteColorTexture(Texture2D: newTexture);

Change the palette alpha gradient texture (by index):
picker.ChangePaletteColorTexture(Texture2D: newTexture);

Change the palette alpha gradient texture with a new texture:
picker.ChangePaletteAlphaGradientTexture(Texture2D: newTexture);

< Extra >

Make an UI object(e.g. Image, RawImage, Text) as draggable:
DDraggableUI.SetDraggable(GameObject: targetUI, bool: isEnabled);

Set canvases to be managed by the DcanvasLayerHandler, for automatically handle the canvas layers when the user drag/tap on the UI:
DCanvasLayerHandler.iAddCanvas(Canvas: targetCanvas);

(5) Make Your Own Color Picker Prefab

How if you still need some custom designed color picker for your app?

We suggest to clone from the provided prefabs, give the new prefab(s) an unique name.

Put the new prefab(s) in the ‘Resources’ folder.

Modify the new prefab as you want.

Instantiate the new color picker in your script:
AColorPicker picker = AcolorPicker.Create(string: YourColorPickerPrefabName);
or
AColorPicker picker = AcolorPicker.Create(Transform: parentTransform,
string: YourColorPickerPrefabName);

THANK YOU

Thank you for using this package!
For any question, suggestion and bug report please contact us at swan.ob2@gmail.com.
Remember to rate this asset on the Asset Store. Your review is always appreciated, and very important to the development of this asset!

Review And Rating

Visit our asset page to find out more!
https://www.swanob2.com/assets

SWAN DEV