# Color Picker (uGUI)

This package contains several color pickers with different styles. You can choose one of the ready-made prefabs or build your own color picker with a wide range of different components.

Table of Content

Find more information online:

- Manual: https://reko3d.com/colorpicker/docs/index.html
- API Documentation: https://reko3d.com/colorpicker/docs/api/ColorPicker.html
- WebGL Demo: https://reko3d.com/colorpicker/demo/

## Features

- RGB and HSV values
- Color palettes
- Alpha values
- Radial sliders
- Linear sliders
- 2D sliders
- TMPro input fields for all components
- TMPro input fields for HEX values
- Color themes for dark and light mode

## Supported environments:

- Windows, Android, iOS, WebGL, VR
- Universal Render Pipeline, Built-In Render Pipeline
- New and old input system
- Gamma and linear color space
- Screenspace and worldspace canvas
- Unity 2022 LTS and higher

# Getting Started

## Prefabs

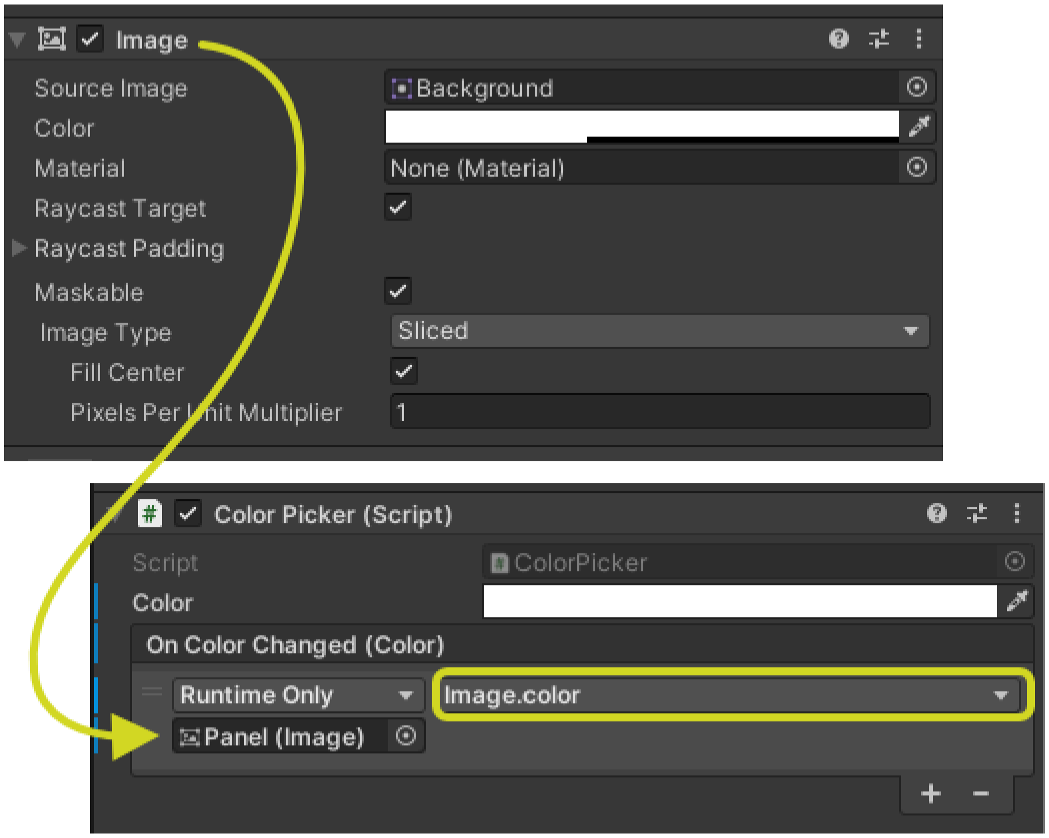
Create a canvas and add one of the color pickers from Content/Prefabs/ColorPickers

## Use selected color

The color picker has an OnColorChanged-event. You can use it in the Unity editor or by script.

For example, you can assign the selected color to a UI image panel.

- Drag the gameobject with the image component to the event handler in the inspector.
- Select Image.color as function.



Image

Source Image: Background
Color
Material: None (Material)
Raycast Target: ✓
Raycast Padding
Maskable: ✓
Image Type: Sliced
Fill Center: ✓
Pixels Per Unit Multiplier: 1

Color Picker (Script)

Script: ColorPicker
Color
On Color Changed (Color)
Runtime Only | Image.color
Panel (Image)

Alternatively, you can add a listener in a script for handling color changes.

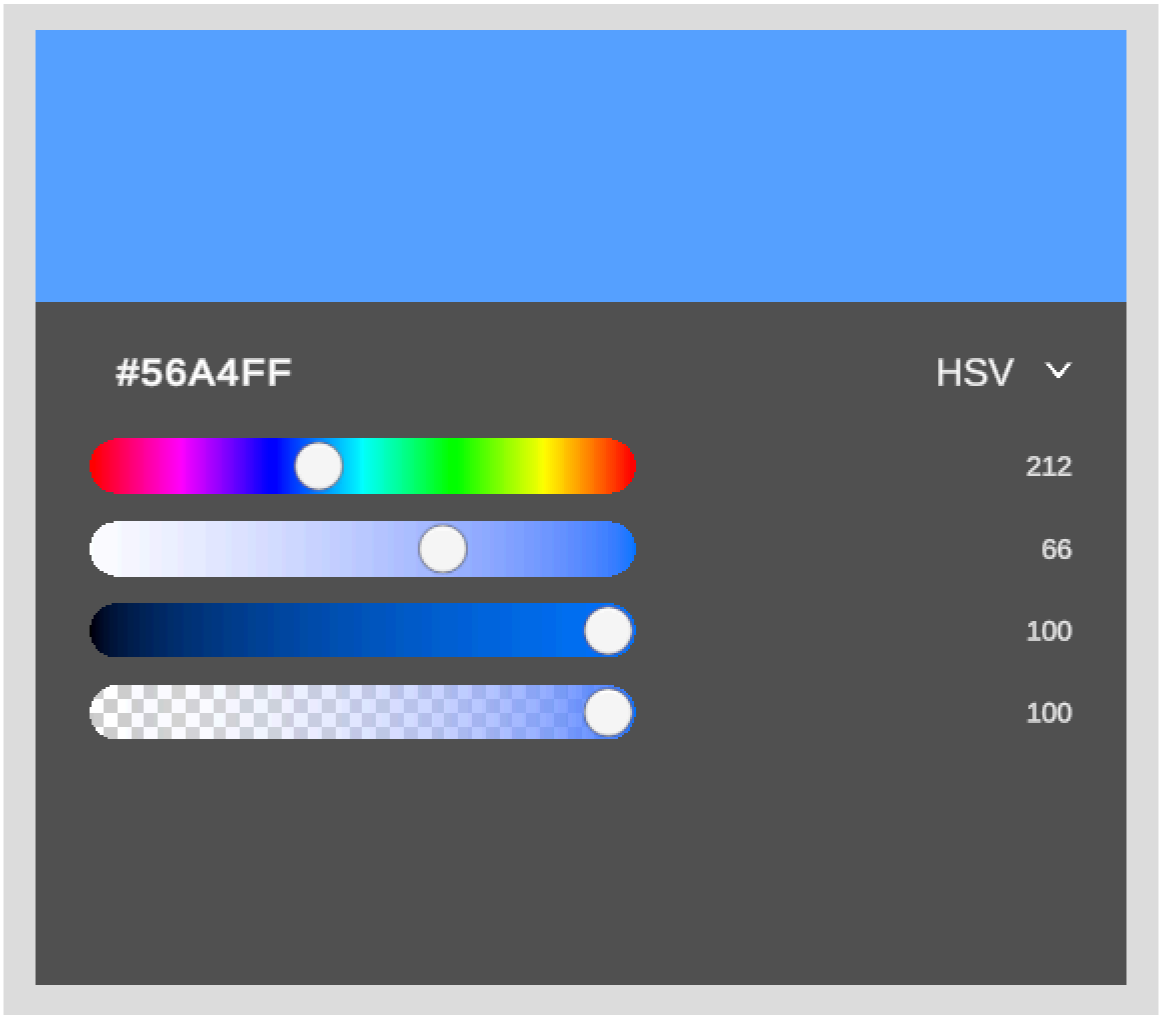
```
void Start()
{
    m_colorPicker.OnColorChanged.AddListener(OnColorChanged);
}
void OnDestroy()
{
    m_colorPicker.OnColorChanged.RemoveListener(OnColorChanged);
}
```

```
private void OnColorChanged(Color color)
{
    // Do something
}
```

# Predefined color pickers

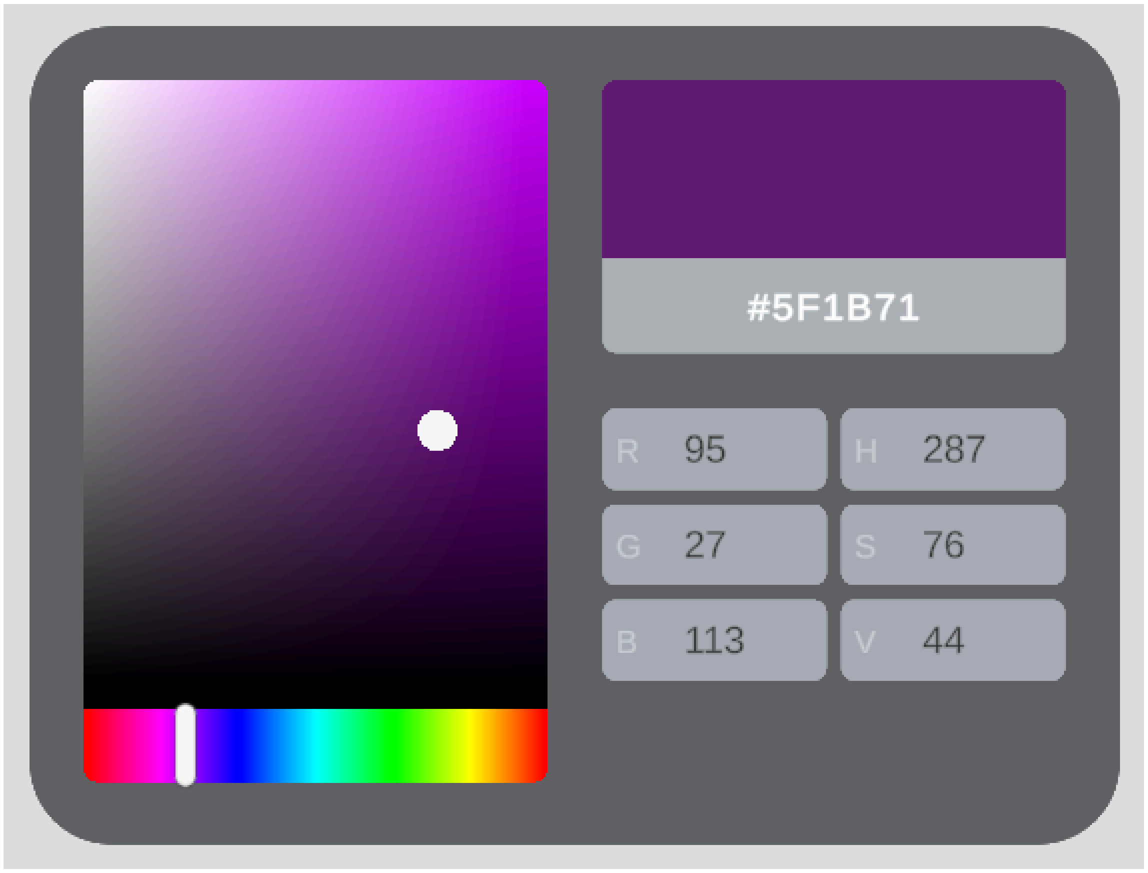
Predefined color pickers are located in Content/Prefabs/ColorPickers.

## ColorPicker #1



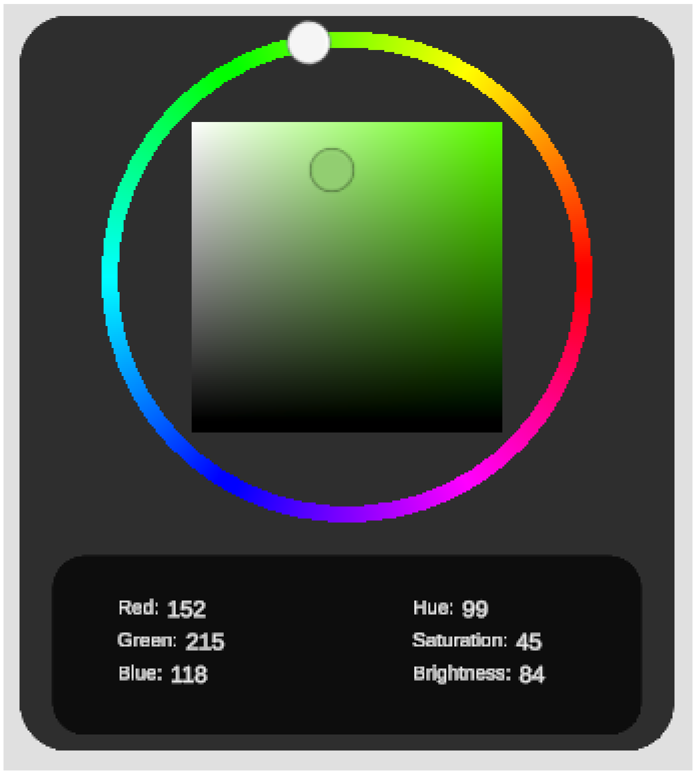
- Switch between RGB and HSV values
- Set Alpha channel
- Linear sliders
- Numerical text input for all channels
- Hex input

## ColorPicker #2



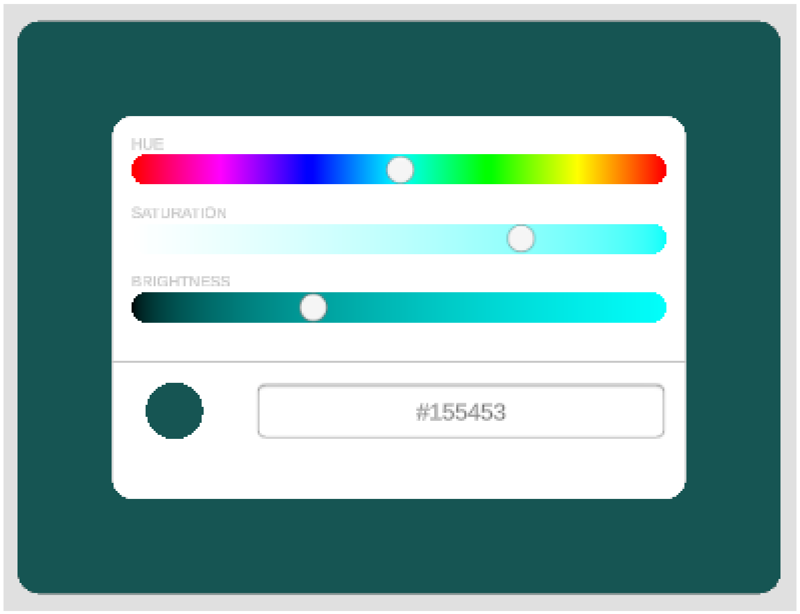
- Linear Hue Slider
- 2D Saturation/Brightness Slider
- Hex input
- Text inputs for RGB and HSV values

## ColorPicker #3



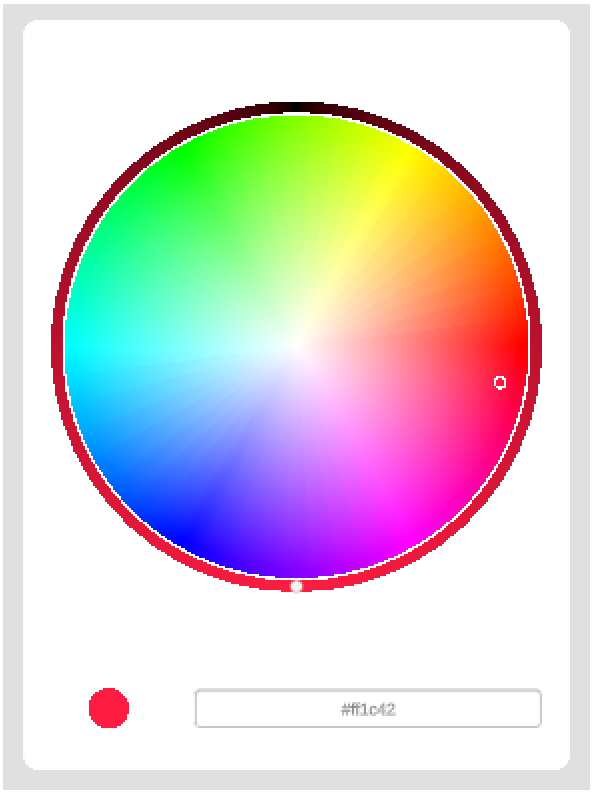
- Radial Hue Slider
- 2D Saturation/Brightness Slider
- Text inputs for RGB and HSV values

## ColorPicker #4



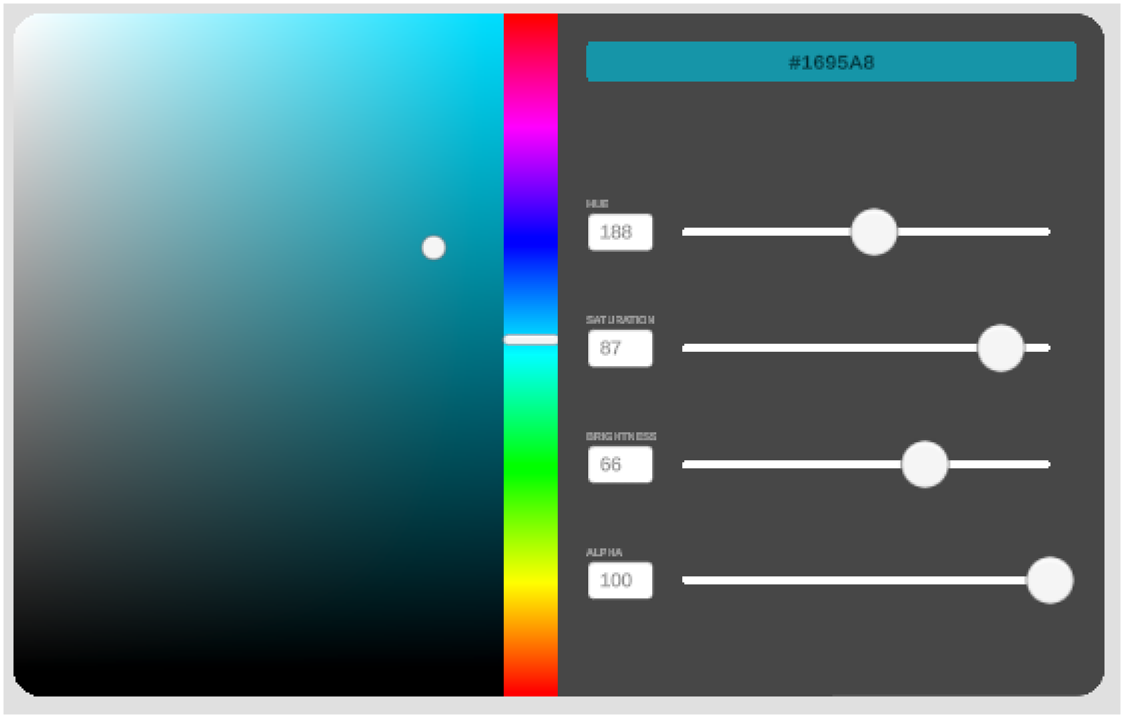
- Sliders for Hue, Saturation, Brightness
- HEX input
- Colored background

## ColorPicker #5



- Radial Hue/Saturation Slider
- Radial Brightness Slider
- Hex input

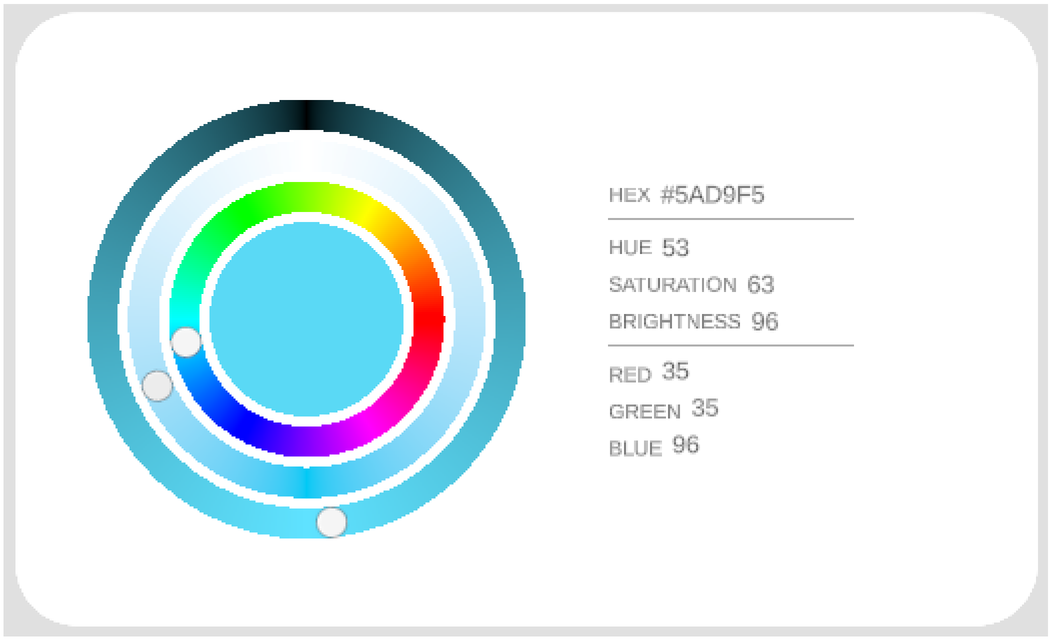
## ColorPicker #6



- 2D Saturation/Brightness Slider
- Vertical Hue Slider

- Hex input
- HSV and Alpha Values (Sliders & Text)

## ColorPicker #7

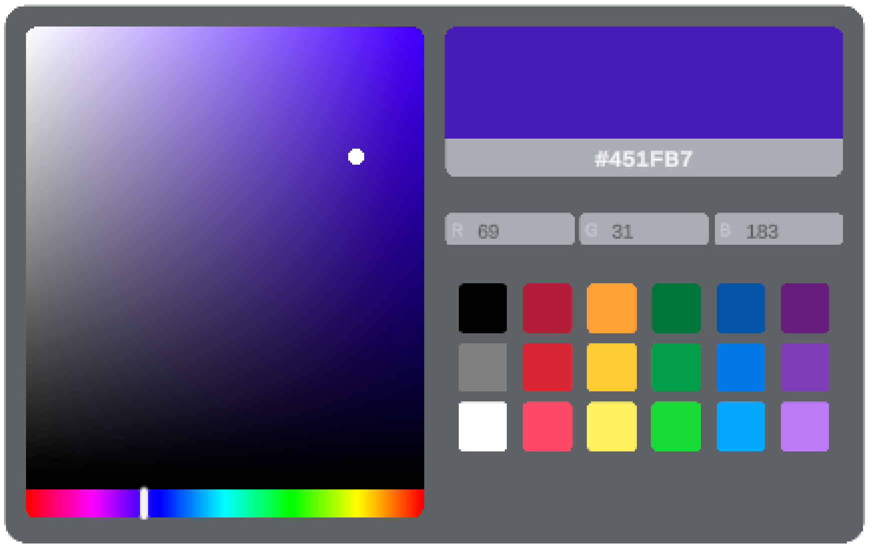

HEX #5AD9F5

HUE 53
SATURATION 63
BRIGHTNESS 96

RED 35
GREEN 35
BLUE 96

- Radial sliders for Hue, Saturation, Brightness
- Text inputs for Hex, HSV and RGB

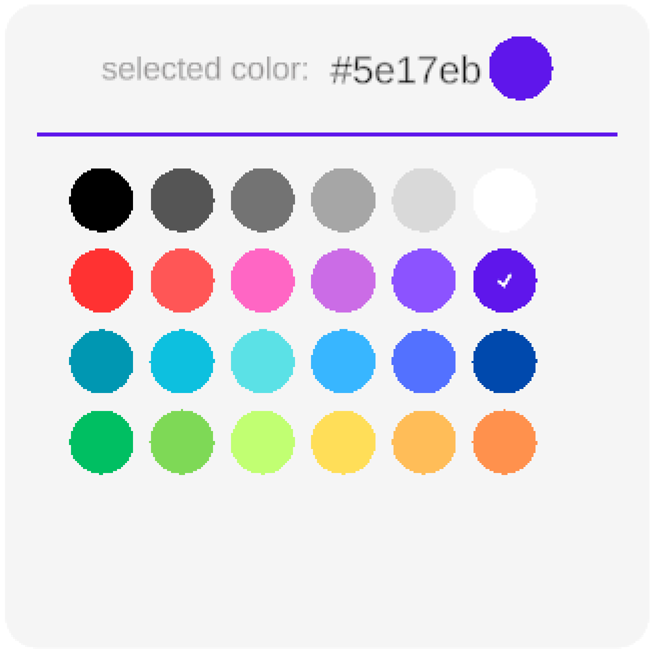
## ColorPicker #8


#451FB7

R 69 G 31 B 183

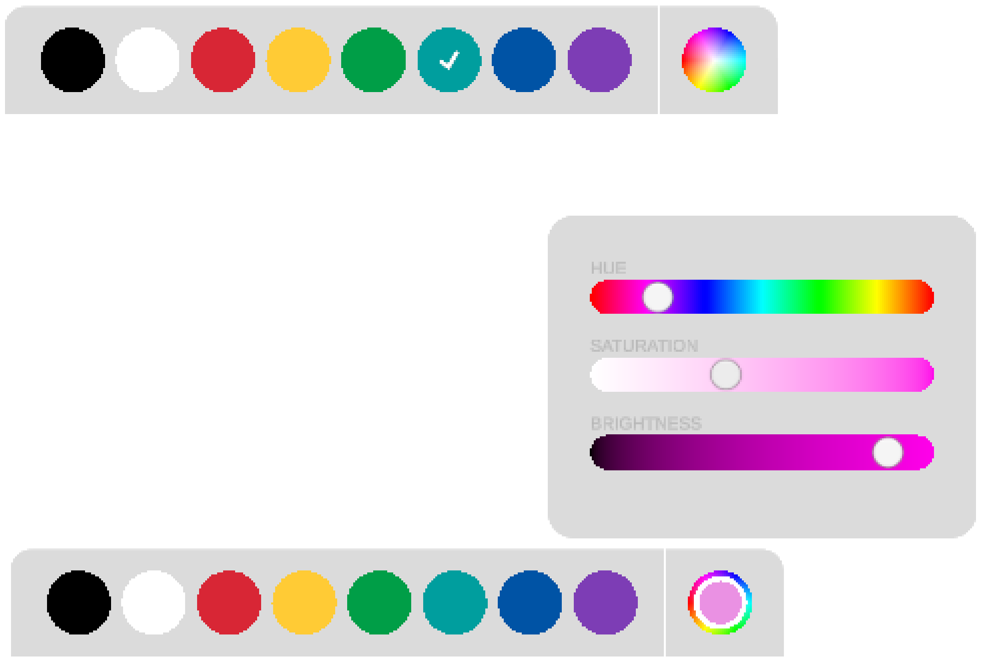
- 2D Saturation/Brightness Slider
- Vertical Hue Slider
- Hex input
- RGB inut
- Color palette

## ColorPicker #9



- Color palette

## ColorPicker #10



- Color palette
- Popup with HSV input sliders

# Create custom color pickers

## Architecture

The top game object has to contain a ColorPicker component.

The children contain components for input or output of color values. E.g. a radial slider for setting the hue value.

An input component usually consists of a ISingleInput and a ColorPickerBinding. The ISingleInput, e.g. a radial slider, defines how the input is entered by the user. The ColorBinding defines which color value is modified, e.g. hue.

A color picker game object usually has a structure like this:

```
- Color Picker
    - Hue Component
      Slider (ISingleInput<float>)
      ColorBinding
    - Saturation Component
      Slider (ISingleInput<float>)
      ColorBinding
    - Value Component
      Slider (ISingleInput<float>)
      ColorBinding
    - Color Output
```

## Input components

Input components implement the interface ISingleInput. Multidimensional components such as the 2D slider use an ISingleInput component for each of their components.

The input components do not depend on a ColorPicker-object and work on their own. They just define the input and output of a single value.

Currently, the following input components are available in the package:

- Linear slider (float)
- Radial slider (float)
- 2D slider (two floats)
- Text input (float)
- Hex input (string)
- Color palette (Color)

## Color bindings

The component ColorPickerBinding is necessary for connecting an input component with a color picker.

The binding only works if

- a ColorPicker component is placed on a parent object of the binding and
- a ISingleInput component is placed on the same game object.

The bindings work in both directions: as input to the color picker and as output if the color has changed.

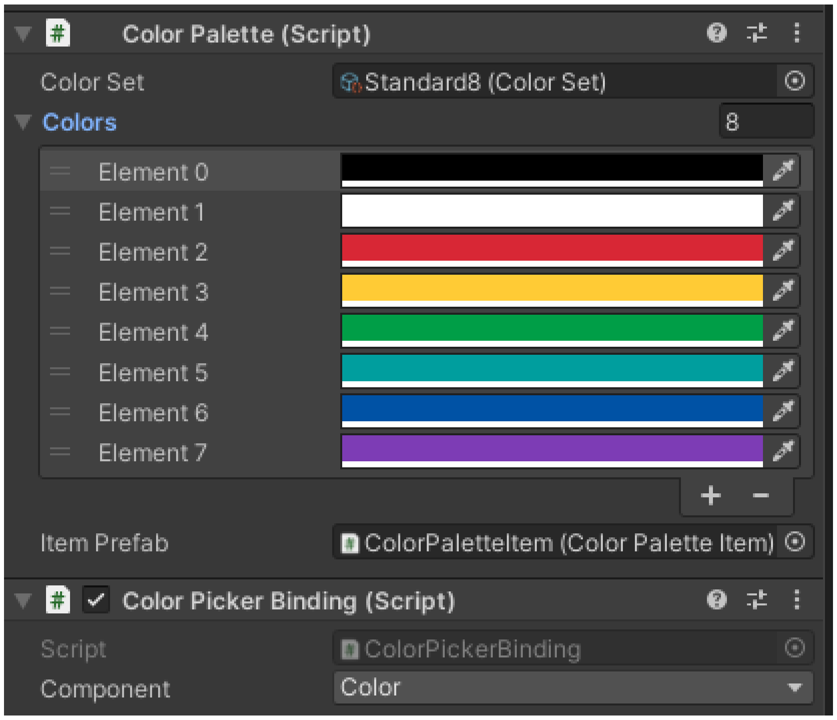
It's possible to use multiple components for the same color component. For example, a slider and a text input for the Hue component.

## Color output

The ColorOutput component needs to be a child of a ColorPicker. It applies the selected color to a UI Graphic in the same game object.

It's possible to apply a conversion to the color. You can set one or more HSV-values to their maximum. E.g. a color should be fully saturated.

## Color palettes


Color Palette (Script)

| | |
|---|---|
| Color Set | Standard8 (Color Set) |
| Colors | 8 |
| Element 0 | ⬛ |
| Element 1 | ⬜ |
| Element 2 | 🟥 |
| Element 3 | 🟨 |
| Element 4 | 🟩 |
| Element 5 | 🟦 |
| Element 6 | 🟦 |
| Element 7 | 🟪 |
| Item Prefab | ColorPaletteItem (Color Palette Item) |

Color Picker Binding (Script)

| | |
|---|---|
| Script | ColorPickerBinding |
| Component | Color |

Color palettes use a predefined set of colors. You can define the colors in the editor. Either assign a ColorSet asset from Content/ColorSets or change the colors directly in the inspector.

You can create your own Color Set in the Project View with Create/Color Picker/Color Set.

The color palette is connected to a color picker like other components. Add a binding to the game object and select Color as component.

Each color is visualized with a ColorPaletteItem. It contains a field for the color and for being selected. Do not change the color palette items inside the color palette. These are automatically recreated in edit mode and at runtime.

## Predefined components

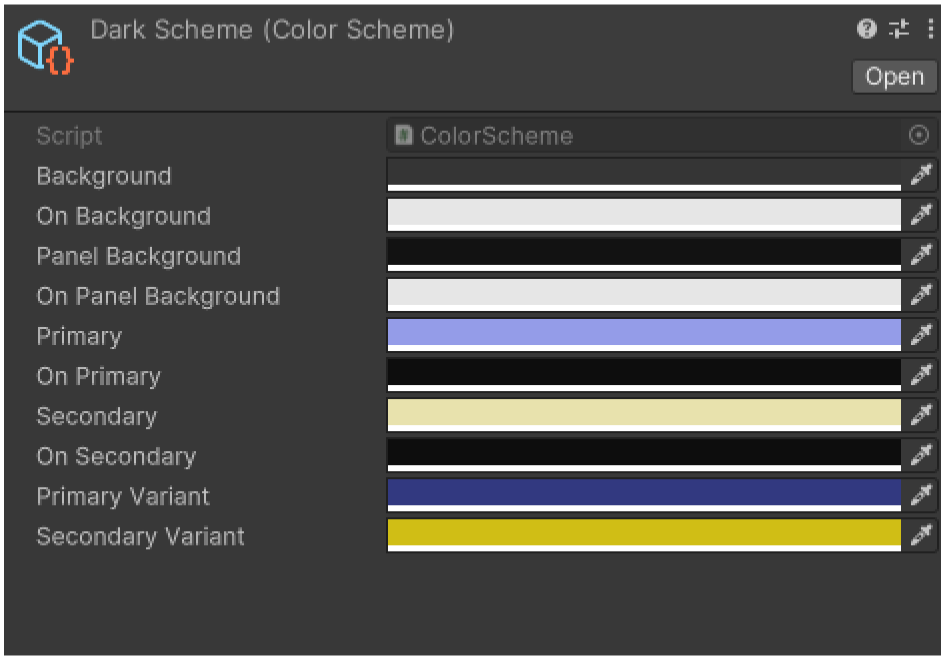
There are multiple ready-made components in Content/Prefabs/Components.

Some of them already contain a binding to the color picker. They contain a color component in their name, e.g. hue or saturation. Just drop them inside a color picker and they will automatically work.

Some components, e.g. the LinearSlider-prefab, are generic and do not contain a color binding yet. You have to add the color binding manually.

# Color themes

A ColorScheme defines a set of colors. They refer to typical UI elements such as Background, Panel, Primary, etc. There are already some color schemes in Reko/Shared/Content/ColorSchemes. You can create your own with "Create>Reko>Color Scheme".


Dark Scheme (Color Scheme)

| | |
|---|---|
| Script | ColorScheme |
| Background | ⬛ |
| On Background | ⬜ |
| Panel Background | ⬛ |
| On Panel Background | ⬜ |
| Primary | 🟪 |
| On Primary | ⬛ |
| Secondary | 🟨 |
| On Secondary | ⬛ |
| Primary Variant | 🟦 |
| Secondary Variant | 🟨 |

A color scheme is applied to a scene with a ColorTheme object. A color theme contains two color schemes: a dark color scheme and a light color scheme. It's possible to switch between them during runtime.

A ThemeColor can be assigned to any visual UI element. It defines which color property (Background, Panel, etc.) should be applied at runtime. If a ColorTheme is present in the scene, the color of the visual item changes acoordingly at runtime.

Color themes are independent from the color picker. They are a generic solution to switch UI elements between dark and light colors at runtime.

# Changelog

## Version 1.2.0 (December 2024)

- Color themes
- Shared code in separate folder

## Version 1.1.0 (July 2024)

- Color palettes
- New color picker prefabs #8, #9, #10

## Version 1.0.0 (October 2023)

- First release
- Color picker prefabs #1 - #7