MaterialUI Documentation

Contents

[Miscellaneous 2](#_Toc405855221)

[Colors 2](#_Toc405855222)

[Image Assets 2](#_Toc405855223)

[Core Components 2](#_Toc405855224)

[ButtonConfig 2](#_Toc405855225)

[CheckboxConfig 2](#_Toc405855226)

[RadioGroupConfig 2](#_Toc405855227)

[SwitchConfig 3](#_Toc405855228)

[InputFieldConfig 3](#_Toc405855229)

[SliderConfig 3](#_Toc405855230)

[SelectionBoxConfig 4](#_Toc405855231)

[Tools and Extra Classes 5](#_Toc405855232)

[Anims 5](#_Toc405855233)

[ToastControl 5](#_Toc405855234)

[CustomInkBlotCreator 6](#_Toc405855235)

[ShadowSnap 6](#_Toc405855236)

[ShadowGen 7](#_Toc405855237)

[Toaster 7](#_Toc405855238)

# Miscellaneous

## Colors

MaterialUI comes with the whole Material Design color palette, in an easy-to-use Unity library format. To access these colors, simply click on the dropdown in the bottom-right of the color picker, and pick your desired library.

## Image Assets

MaterialUI comes with all the assets needed for the core components, in a variety of sizes to suit most needs. As more stuff is added, so will more components. At some point, all the icons from Google’s set will be added as well. Images for components are located in “Assets/MaterialUI/Images/Components” and Icons in “Assets/MaterialUI/Images/Icons”

# Core Components

## ButtonConfig

**inkBlotEnabled** (bool) – Is the Ink Blot enabled?

**autoInkBlotSize** (bool) – Is the size of the Ink Blot automatically calculated? If true, then the size is the length of the longest side of the parent RectTransform / 1.5

**inkBlotSize** (int) – The radius, in pixels, of the ink blot when fully expanded

**inkBlotSpeed** (float) – The speed of the Ink Blot animation

**inkBlotColor** (Color) – The color of the Ink Blot, does not affect the alpha

**inkBlotStartAlpha** (float) – The starting alpha of the Ink Blot

**inkBlotEndAlpha** (float) – The ending alpha of the Ink Blot

**highlightOnClick** (bool) – Does the button highlight when clicked? The highlight color is dictated by inkBlotColor

**highlightOnHover** (bool) – Does the button highlight when hovered over?

**shadows** (array) – The shadows that this script controls

**shadowNormalSize** (int, range(0,3) ) – The size of the shadows when the button is not being interacted with

**shadowHoverSize** (int, range (0,3) ) – The size of the shadows when the button is hovered over / clicked on

## CheckboxConfig

**frameImage** (Image) – The image of the ‘frame’ part

**boxImage** (Image) – The image of the ‘box’ part

**checkImage** (Image) - The image of the ‘check’ part

**animationDuration** (float) – The duration of the toggle animation

## RadioGroupConfig

**radioOnColor** (Color) – The color of the currently selected dot and ring

**animationDuration** (float) – The duration of the toggle animation

## SwitchConfig

**switchOnColor** (Color) – The color of the switch when active

**animationDuration** (float) – The duration of the toggle animation

**switchImage** (Image) – the image of the round switch

**switchImage** (Image) – the image of the long switch background

## InputFieldConfig

**activeColor** (Color) – The color of the bottom line and placeholder text when active

**animationDuration** (float) – The duration of the select/deselect animation

**placeholderText** (Text) – The placeholder text

**activeLine** (Image) – the image of the active line (hidden when not active)

## SliderConfig

**textColor** (Color) – The color of the popup text

**textHasDecimal** (bool) – Does the popup text have a decimal?

**hasPopup** (bool) – Does the slider have a popup with text?

**animationDuration** (float) – The duration of the select/deselect animation

**handle** (RectTransform) – The RecsTransform of the handle

**popup** (RectTransform) – The RectTransform of the popup

**popupText** (Text) – The popup text

## SelectionBoxConfig

**listItems** (array) – The Items that will appear in the list when expanded (modify this to change the items in the list)

**expandDirection** (Popup, Center, PopDown) – Which direction will the list expand?

**autoMaxItemHeight** (bool) – Is the max height of the list limited by the screen height (If true, will enable list scrolling if the list is too long for the screen)

**maxItemHeight** (int) – The max height (number of items) of the list (If the actual list length is higher, list scrolling will be enabled)

**currentSelection** (int) – The ID of the currently selected item (default is -1)

**animationDuration** (float) – The duration of the expand/contract animation

**listLayer** (GameObject) – The layer of the list

**selectedText** (Text) – The text that appears when the list is not expanded

**cancelLayer** (Image) – The invisible layer that contracts the list when clicked

**scrollbar** (Image) – The image of the scrollbar, only visible if list scrolls

**icon** (Image) – The icon next to the selectedText

# Tools and Extra Classes

## Anims

Anims is a static class designed to help with the eased animation of objects. There are 3 methods to choose from:

**EaseInQuint** (startValue, endValue, time, duration) (float)

**EaseInOutQuint** (startValue, endValue, time, duration) (float)

**EaseOutQuint** (startValue, endValue, time, duration) (float)

Which each have 4 parameters:

**startValue** (float) – The value at the start of the animation - this value usually doesn't change between frames

**endValue** (float) – The value at the end of the animation - this value usually doesn't change between frames

**time** (float) – How much time (in seconds, as float) since the start of the animation - raising this value progresses the animation

**duration** (float) - How long the animation is

Example:

myFloat = Anims.EaseOutQuint (0f, 1f, Time.realTimeSinceStartup, 10f);

This would progress myFloat from 0 to 1 over 10 seconds, getting slower as it nears 1.

## ToastControl

Toast control is a class designed to help with displaying toast messages.

**InitToastSystem**() (void) – If using the ToastControl class, InitToastSystem must be called before any toasts are created.

**MakeToast**(content, duration, panelColor, textColor, fontSize) (void) – This creates a toast message

**content** (string) – The string of the message

**duration** (float) – The time that the toast is displayed

**panelColor** (Color) – The color of the toast panel

**textColor** (Color) – The color of the toast text

**fontSize** (int) –The size of the toast message

## CustomInkBlotCreator

This class is designed to help with creating ink blots over images.

**inkBlotEnabled** (bool) – Is the Ink Blot enabled?

**autoInkBlotSize** (bool) – Is the size of the Ink Blot automatically calculated? If true, then the size is the length of the longest side of the parent RectTransform / 1.5

**inkBlotSize** (int) – The radius, in pixels, of the ink blot when fully expanded

**inkBlotSpeed** (float) – The speed of the Ink Blot animation

**inkBlotColor** (Color) – The color of the Ink Blot, does not affect the alpha

**inkBlotStartAlpha** (float) – The starting alpha of the Ink Blot

**inkBlotEndAlpha** (float) – The ending alpha of the Ink Blot

**toggleMask** (bool) – Is the mask of the image toggled on and off when no ink blots are visible to improve performance? Disable if you need the mask for other things.

**dragCheck** (bool) – Is the checking of the click and pointer position delayed so as to not create an ink blot if the user if swiping (ie. Scrolling through a list)?

**dragLimit** (bool) – How much is the drag check delayed?

## ShadowSnap

This class is designed to help with keeping shadows snapped to their target image. Keep in mind that this script will run in editor mode.

**targetRect** (RectTransform) – The RectTransform of the target object/image to snap to.

**xPadding** (float) – The amount of horizontal padding between the edge of the target image and shadow image

**yPadding** (float) – The amount of vertical padding between the edge of the target image and shadow image

**snapEveryFrame** (bool) – Does the shadow snap every frame? If not required, disable for better performance

## ShadowGen

This class is designed to help with generating soft drop shadows for any image. To use, attach this to an object with an Image component, set sourceImage as the image you want shadowed, configure the settings to your liking, and hit ‘Generate Shadow’. Please note that the sprite of the source image **must be set as read-write enabled**! Generated shadow sprites are put in “Assets/MaterialUI/GeneratedShadows” and assigned a random name.

**sourceImage** (Image) – The image that you want shadowed

**blurRange** (int, range(0,5)) – How much the edges of the shadow should be blurred in each iteration.

**blurIterations** (int) – The number of blur iterations that will be done on the shadow

**shadowRelativePosition** (Vector3) – The difference in position from the source image to the shadow

**shadowRelativeSize** (Vector2) – The difference in size from the source image to the shadow

**shadowAlpha** (float) – The alpha level of the shadow

## Toaster

This script serves mainly as an easy, non-code way to instantiate toast messages. To use, simply attach to an object and call the PopupToast() method.

**content** (string) – The string of the message

**duration** (float) – The time that the toast is displayed

**panelColor** (Color) – The color of the toast panel

**textColor** (Color) – The color of the toast text

**fontSize** (int) –The size of the toast message