Rock.Mobile.UI

Because Rock.Mobile.UI comprises such a large part of Rock.Mobile, it has been broken out into its own document.

Rock.Mobile.UI is a set of platform abstracted UI components that you can use the develop UIs for both iOS and Android and share the same code. It is regularly evolving, but contains most elements needed for a basic UI page.

Additionally, native UI can be intermixed with Rock.Mobile.UI, through a platform abstracted hierarchy method “AddAsSubview”, which will be detailed later in this document.

Table of Contents

[PlatformBaseUI 1](#_Toc418785193)

[PlatformBusyIndicator 2](#_Toc418785194)

[PlatformButton 2](#_Toc418785195)

[PlatformCardCarousel 3](#_Toc418785196)

[PlatformCircleView 3](#_Toc418785197)

[PlatformImageView 4](#_Toc418785198)

[PlatformBaseLabelUI 4](#_Toc418785199)

[PlatformLabel 5](#_Toc418785200)

[PlatformTextField 5](#_Toc418785201)

[PlatformTextView 6](#_Toc418785202)

[PlatformView 7](#_Toc418785203)

[Util 7](#_Toc418785204)

### PlatformBaseUI

Abstracted base UI class that provides an interface to platform specific UI controls. All platform UI objects derive from this.

Usage:

Namespace: Rock.Mobile.UI

Methods:

**void** Init( )

This should be called once at startup to allow the device to init anything that can be done before actually creating / rendering the UI.

(E.g. Android uses this to prepare an alpha mask necessary for certain UI effects)

**uint** BackgroundColor - Gets or sets the background color of the UI object.

**uint** BorderColor – Gets or sets the border color of the UI object. (Requires a BorderWidth of at least 1)

**float** BorderWidth – Gets or sets the border width of the UI object.

**float** Opacity – Gets or sets the opacity of the view.

**float** ZPosition – Gets or sets the Z position order of a UI object in its hierarchy.

**Note:** Android does not have a z order concept, so instead, a value of > 0 will simply bring the view to front.

**RectangleF** Bounds – Gets or sets the local-space bounds of the UI object.

**RectangeleF** Frame – Gets or sets the parent-space position and dimensions of the UI object.

**bool** Hidden – Gets or sets the Hidden flag. A value of true hides the object, a value of false unhides it.

**bool** UserInteractionEnabled – Gets or sets the user interaction flag. A value of true allows the UI object to receive input. A value of false disables input on the object.

**void** AddAsSubview( **object** masterView )

Adds the instance as a **child** of the masterView provided. (E.g. Child.AddAsSubview( parent ))

Doing it this way allows these child objects to attach themselves to **native** parent objects.

**void** RemoveAsSubview( **object** masterView )

Removes the instance as a **child** of the masterView provided. (E.g. Child.RemoveAsSubview( parent ))

Doing it this way allows these child objects to attach themselves to **native** parent objects.

### PlatformBusyIndicator

Implements a platform-native busy “spinner”.

Methods:

**PlatformBusyIndicator** Create( )

Creates and returns a new instance of a BusyIndicator.

**uint** Color – Gets or sets the color of the spinner.

### PlatformButton

Implements a platform-native UI button.

Methods:

**PlatformButton** Create( )

Creates and returns a new instance of a Button.

**OnClick** ClickEvent – Sets or gets the event to call when the button is clicked/tapped.

**void** SetFont( **string** fontName,

**float** fontSize )

Sets the Font that the button should use for its label.

**void** SizeToFit( )

Resizes the text frame to fit the current text.

**uint** TextColor – Gets or sets the color of the text.

**string** Text – Gets or sets the text for the button label.

**float** CornerRadius – Gets or sets the radius of the corners. This allows rounded corner buttons.

### PlatformCardCarousel

Implements a series of ‘nodes’ that UI views can be attached to, and allows the user to swipe between them. Similar to the home screen on Android / iOS.

Methods:

**PlatformCardCarousel** Create( **object** parentView,

**float** cardWidth,

**float** cardHeight,

**RectangleF** boundsInParent,

**float** animationDuration )

Returns a new instance of a **PlatformCardCarousel**. The values provided let the carousel know how to properly animate each “card”.

Arguments:

**object** parentView – The parent UI view that the carousel should attach its nodes to.

**float** cardWidth – The width of each “card” that will be attached.

**float** cardHeight – The height of each “card” that will be attached.

**RectangleF** boundsInParent – Defines the region that the cards should exist in.

**float** animationDuration – When animating the card movement, defines the duration in seconds.

**int** CenterCardIndex – Gets the index of the card currently centered on screen. This is the index that should be used to know where in the backing list the carousel is.

**bool** Hidden – Hides or unhides the carousel and all child cards.

**void** AddCard( **PlatformView** view )

Adds a new “card” to the carousel. Other than its width and height needing to match what was provided to the constructor, this view can be anything.

**void** LayoutChanged( **float** cardWidth,

**float** cardHeight,

**RectangleF** boundsInParent )

Should be called when the layout of the device has changed.

Arguments:

**float** cardWidth – The new width each card should be.

**float** cardHeight – The new height each card should be.

**RectangleF** boundsInParent – The new bounds that the carousel should confine itself to.

**void** Clear( )

Removes all cards and completely resets the carousel.

**void** TouchesBegan( )

Should be called when the parent receives TouchesBegan( ) or equivalent.

**void** TouchesEnded( )

Should be called when the parent receives TouchesEnded( ) or equivalent.

### PlatformCircleView

Implements a UI view that renders as a circle.

Methods:

**PlatformCircleView** Create( )

Returns a new instance of a CircleView.

**object** PlatformNativeObject( )

Returns the native platform object. This allows native code to access features of the **CircleView** that aren’t implemented in platform code.

(e.g. iOS would return a UIView and Android would return a CircleView)

### PlatformImageView

Implements a UI object that can display an image.

Methods:

**PlatformImageView** Create( )

Creates a new instance of an ImageView.

**object** PlatformNativeObject( )

Returns the native platform object. This allows native code to access features of the **ImageView** that aren’t implemented in platform code.

(e.g. iOS would return a UIImageView and Android would return a BorderedRectImageView)

**float** CornerRadius – Gets or sets the corner radius. Allows rounded corners.

**void** SetImage( **MemoryStream** image ) – Sets the image to be displayed.

**bool** ScaleForDPI – Sets whether to downsample the image while loading it to match the device’s DPI scale.

**ScaleType** ImageScaleType – Sets the type of scaling that should be used. Available options are:

Center – Centers the image with no scaling.

ScaleAspectFill – Scales the image to fill the whole viewing area, but

maintains aspect ratio, which means the image might be cropped around edges.

ScaleAspectFit – Scales the image to fully fit within the bounds, maintaining aspect ratio with no cropping.

**void** SizeToFit( ) – Updates the bounds to fit around the image.

**void** Destroy( ) – Frees resources used in storing the image in memory.

**Note: You must call this on Android or memory leaks will occur.**

### PlatformBaseLabelUI

An abstract class that provides common functionality for UI Labels.

Methods:

**void** SetFont( **string** fontName, **float** fontSize )

Sets the font to use for the label.

**void** TextColor – Gets or sets the text color.

**string** Text – Gets or sets the text for the lable.

**TextAlignment** TextAlignment – Gets or sets the alignment value for text.

Options are:

Left, Center, Right, Justified, Natural

**void** SizeToFit( )

Sets the boundaries of the label to fit the text.

**float** CornerRadius – Gets or sets the corner radius for the label. Allows rounded corners. BorderWidth must be at least 1.

### PlatformLabel

Implements a standard UI label.

Methods:

**PlatformLabel** CreateLabel( )

Creates and returns a new instance of a standard label.

**PlatformLabel** CreateRevealLabel( )

Creates and returns a new instance of a **Reveal** Label.

A reveal label supports a ‘reveal’ animation when tapped.

**float** GetFade( )

Returns the 0 – 1 reveal amount of the label **if a reveal label**.

**void** SetFade( **float** fadeAmount )

Sets the 0 – 1 reveal amount of the label **if a reveal label.**

**void** AnimateToFade( **float** fadeAmount )

Animates the reveal amount of the label **if a reveal label.**

**void** AddUnderline( )

Adds an underline to a label. (Works for both Labels and RevealLabels)

### PlatformTextField

Implements a text field. Text fields are designed to be only single line.

For multiline, use a **PlatformTextView**.

Methods:

**PlatformTextField** Create( )

Creates a new instance of a TextField.

**object** PlatformNativeObject – Returns the native platform object. For iOS, that is a UITextField. For Android, it is an BorderedRectEditText.

**string** Placeholder – Gets or sets the text to show when there is no text in the field.

**string** PlaceholderTextColor **–** Gets or sets the color of placeholder text.

**KeyboardAppearanceStyle** KeyboardAppearance – Gets or sets the appearance of the software keyboard used for editing this text field. **This only applies to iOS.**

Options are: Light, Dark

**AutoCorrectionType** AutoCorrectionType – Gets or sets the type of auto correction for the text field.

Options are: Default, No, Yes

**AutoCapitilizationType** AutioCapitalizationType **–** Gets or sets the auto capitalization type for the text field.

Options are: None, Words, Sentences, All

**void** ResignFirstResponder( )

Forces the software keyboard to close.

### PlatformTextView

Implements a text field designed for multiline text editing. For singleline text editing, see **PlatformTextField**

Methods:

**PlatformTextView** Create( )

Creates and returns a new instance of a **TextView**.

**object** PlatformNativeObject – Returns the native platform object. For iOS, that is a UITextView. For Android, it is an BorderedRectEditText.

**string** Placeholder – Gets or sets the text to show when there is no text in the field.

**string** PlaceholderTextColor **–** Gets or sets the color of placeholder text.

**bool** ScaleHeightForText – Lets the text field “wrap” the text, and grow as more text is added. If false, the text will be a fixed size.

**KeyboardAppearanceStyle** KeyboardAppearance – Gets or sets the appearance of the software keyboard used for editing this text field. **This only applies to iOS.**

Options are: Light, Dark

**AutoCorrectionType** AutoCorrectionType – Gets or sets the type of auto correction for the text field.

Options are: Default, No, Yes

**AutoCapitilizationType** AutioCapitalizationType **–** Gets or sets the auto capitalization type for the text field.

Options are: None, Words, Sentences, All

**void** ResignFirstResponder( )

Forces the software keyboard to close.

**void** AnimateOpen( )

Animates the text view open, rather than in a single frame.

**void** AnimateClosed( )

Animates the text view closed, rather than in a single frame.

### PlatformView

Implements a standard UI view.

Methods:

**PlatformView** Create( )

Returns a newly created View.

**PlatformNativeObject –** Gets the native platform object. On iOS, this is a UIView. On Android, it is a BorderedRectView.

**float** CornerRadius – Gets or sets the corner radius for the view. Allows rounded corners. BorderWidth must be at least 1.

### Util

Common functionality needed by Rock.Mobile.UI.

Methods:

**void** AnimateBackgroundColor( **PlatformBaseUI** view,

**uint** targetColor,

**SimpleAnimator.AnimationComplete**

onCompletion = null)

Animates the background color of a view, and optionally calls the provided delegate on completion.

**void** AnimateBackgroundOpacity( **PlatformBaseUI** view**,**

**float** targetOpacity,

**SimpleAnimator.AnimationComplete**

onCompletion = null)

Animates the opacity of the provided view, and optionally calls the provided delegate on completion.

**Android.Graphics.Color** GetUIColor( **uint** color )

*or*

**UIKit.UIColor** GetUIColor( **uint** color )

Returns a newly created platform color object from a **uint** color in the format 0xRRGGBBAA

**uint** UIColorToInt( **Android.Graphics.Color** color )

*or*

**uint** UIColorToInt( **CoreGraphics.CGColor** color )

**uint** UIColorToInt( **UIKit.UIColor** color )

Returns a **uint** color in the format 0xRRGGBBAA from the provided **color** object.