

Simple side menu control for iOS, no code necessary! Lots of customization. Add it to your project in 5 minutes or less.

[#sidemenu](#) [#cocoapods](#) [#carthage](#) [#storyboard](#) [#menus](#) [#xcode](#) [#swift](#) [#menu](#)

410 commits

56 branches

22 releases

19 contributors

MIT

Branch: master


New pull request

Create new file

Upload files

Find file



















Clone or download



jonkykong

Update README.md

Latest commit 175042e 2 days ago

<div></div> <div>.github</div>	Create ISSUE_TEMPLATE.md	8 months ago
<div></div> <div>Example</div>	Renamed class and refactored how blurring works in UITableViewVibrant...	4 months ago
<div></div> <div>ExampleTests</div>	moving example project to main xcworkspace	a year ago
<div></div> <div>Pod/Classes</div>	Exposed defaultManager for Objective-C support.	2 months ago
<div></div> <div>Pods</div>	Additional Swift4 support updates.	4 months ago
<div></div> <div>SideMenu.xcodeproj</div>	Renamed class and refactored how blurring works in UITableViewVibrant...	4 months ago
<div></div> <div>SideMenu.xcworkspace</div>	moving example project to main xcworkspace	a year ago
<div></div> <div>SideMenu</div>	Removed redundant copies of source files causing confusion with updat...	a year ago
<div></div> <div>etc</div>	Renaming assets.	2 years ago
<div></div> <div>.gitignore</div>	Undoing .gitignore changes, removing unneeded project scheme.	a year ago
<div></div> <div>.swift-version</div>	Another attempt to get things working for Swift 3...	a year ago
<div></div> <div>.travis.yml</div>	moving example project to main xcworkspace	a year ago
<div></div> <div>CODE_OF_CONDUCT.md</div>	Create CODE_OF_CONDUCT.md	4 months ago
<div></div> <div>LICENSE</div>	Update LICENSE	a year ago
<div></div> <div>Podfile</div>	moving example project to main xcworkspace	a year ago
<div></div> <div>Podfile.lock</div>	Undoing .gitignore changes, removing unneeded project scheme.	a year ago
<div></div> <div>README.md</div>	Update README.md	2 days ago
<div></div> <div>SideMenu.podspec</div>	Updated podspec.	2 months ago

 README.md

SideMenu

pod v3.1.5 Carthage compatible license MIT platform ios  821k  71k/month  16k/week

If you like SideMenu, give it a ★ at the top right of this page.

Using SideMenu in your app? [Send](#) me a link to your app in the app store!

Hi, I'm Jon Kent and I am an iOS designer, developer, and mobile strategist. I love coffee and play the drums.

- [Hire me](#) to help you make cool stuff. *Note: If you're having a problem with SideMenu, please open an [issue](#) and do not email me.*

- Check out my [website](#) to see some of my other projects.
- Building and maintaining this **free** library takes a lot of my time and **saves you time**. Please consider paying it forward by supporting me with a small amount to my [PayPal](#). (only 2 people have donated since inception 😞 but thank you to those who have!)

- **Overview**
 - [Preview Samples](#)
- **Requirements**
- **Installation**
 - [CocoaPods](#)
 - [Carthage](#)
- **Usage**
 - [Code-less Storyboard Implementation](#)
 - [Code Implementation](#)
- **Customization**
 - [SideMenuManager](#)
 - [UISideMenuNavigationController](#)
 - [UISideMenuNavigationControllerDelegate](#)
 - [Advanced](#)
- [Known Issues](#)
- [Thank You](#)
- [License](#)

Overview

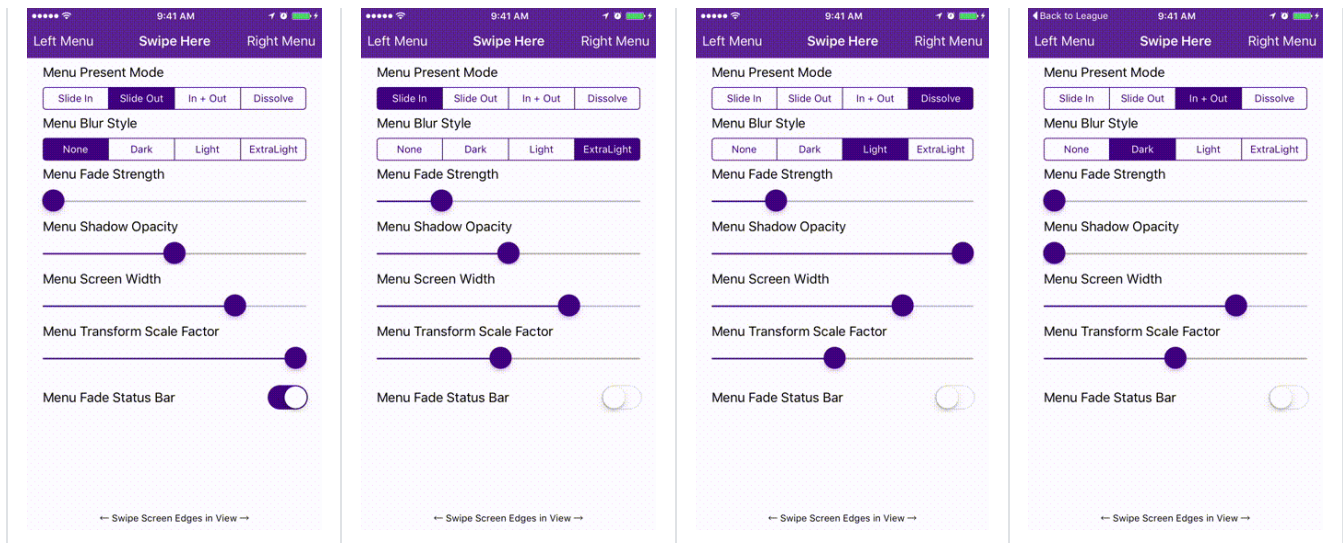
SideMenu is a simple and versatile side menu control written in Swift.

- **It can be implemented in storyboard without a single line of [code](#).**
- Four standard animation styles to choose from (there's even a parallax effect if you want to get weird).
- Highly customizable without needing to write tons of custom code.
- Supports continuous swiping between side menus on both sides in a single gesture.
- Global menu configuration. Set-up once and be done for all screens.
- Menus can be presented and dismissed the same as any other view controller since this control uses [custom transitions](#).
- Animations use your view controllers, not snapshots.
- Properly handles screen rotation and in-call status bar height changes.

Check out the example project to see it in action!

Preview Samples

Slide Out	Slide In	Dissolve	Slide In + Out



Requirements

- Xcode 9.
- iOS 8 or higher.

Installation

CocoaPods

[CocoaPods](#) is a dependency manager for Cocoa projects. You can install it with the following command:

```
$ gem install cocoapods
```

To integrate SideMenu into your Xcode project using CocoaPods, specify it in your Podfile :

```
source 'https://github.com/CocoaPods/Specs.git'
platform :ios, '8.0'
use_frameworks!

pod 'SideMenu'

# For Swift 3 (no longer maintained), use:
# pod 'SideMenu', '~> 2.3.4'
```

Then, run the following command:

```
$ pod install
```

Carthage

[Carthage](#) is a decentralized dependency manager that builds your dependencies and provides you with binary frameworks.

You can install Carthage with [Homebrew](#) using the following command:

```
$ brew update
$ brew install carthage
```

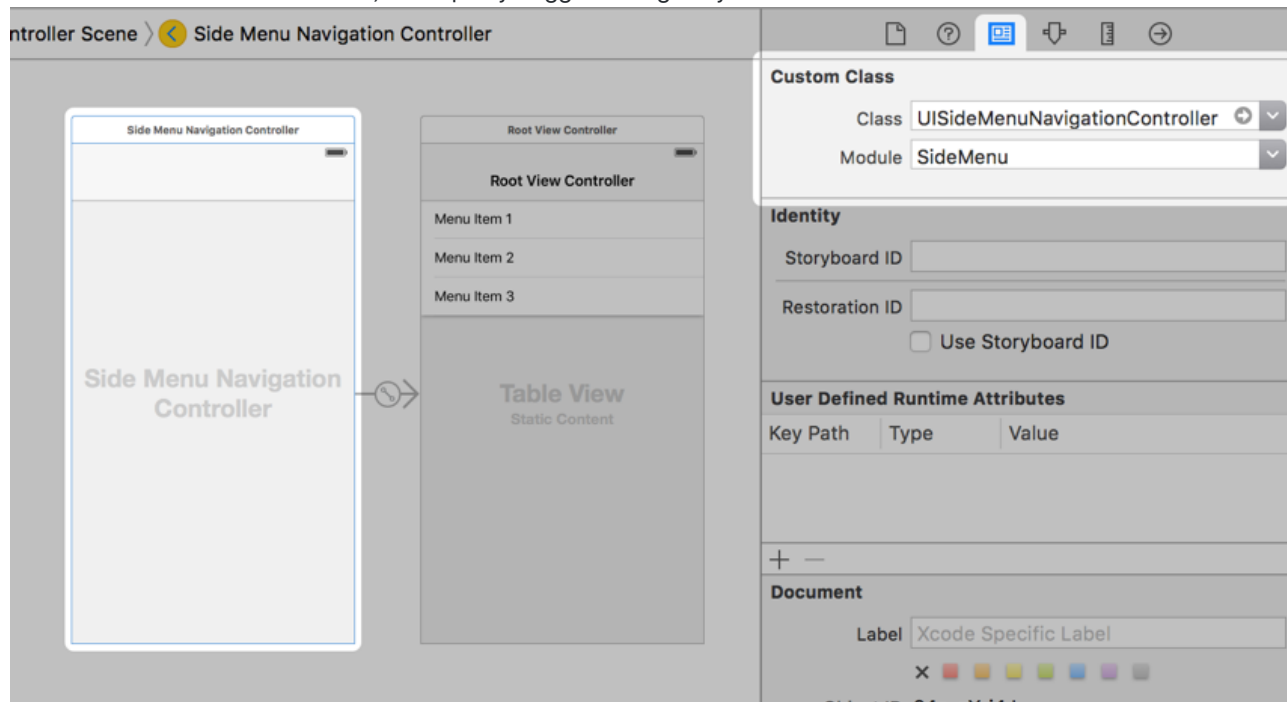
To integrate SideMenu into your Xcode project using Carthage, specify it in your `Cartfile` :

```
github "jonkykong/SideMenu" "master"
```

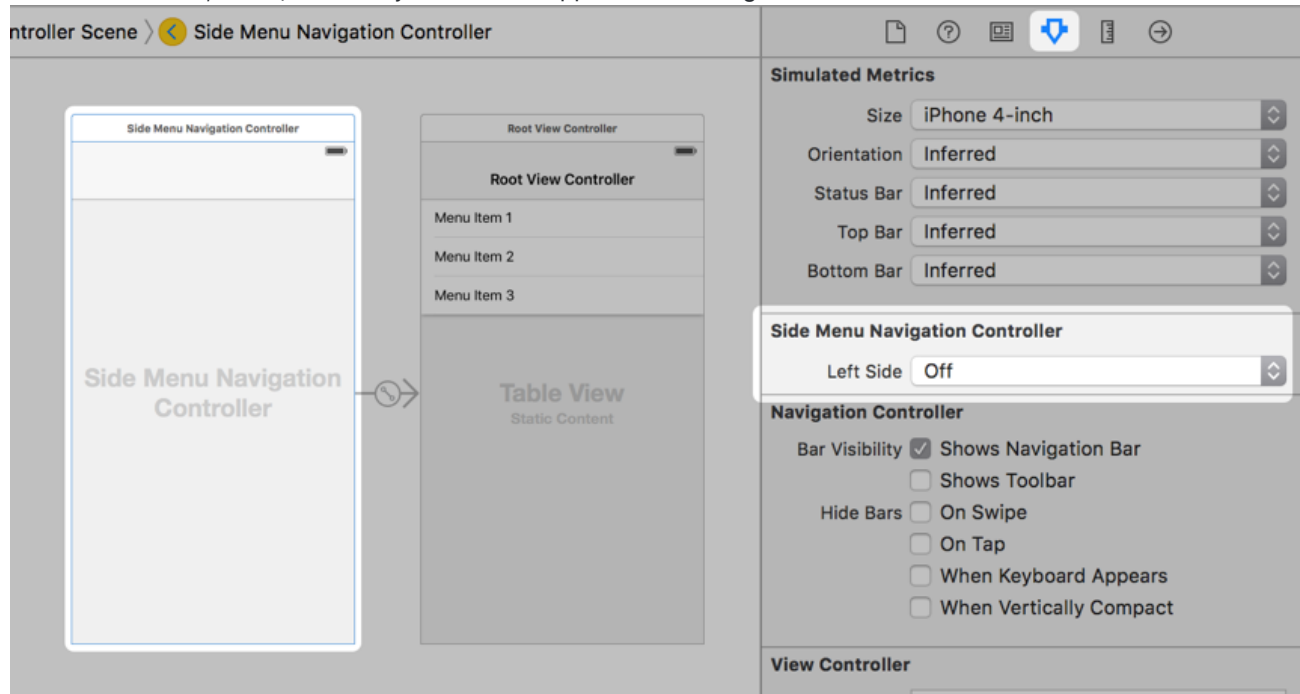
Usage

Code-less Storyboard Implementation

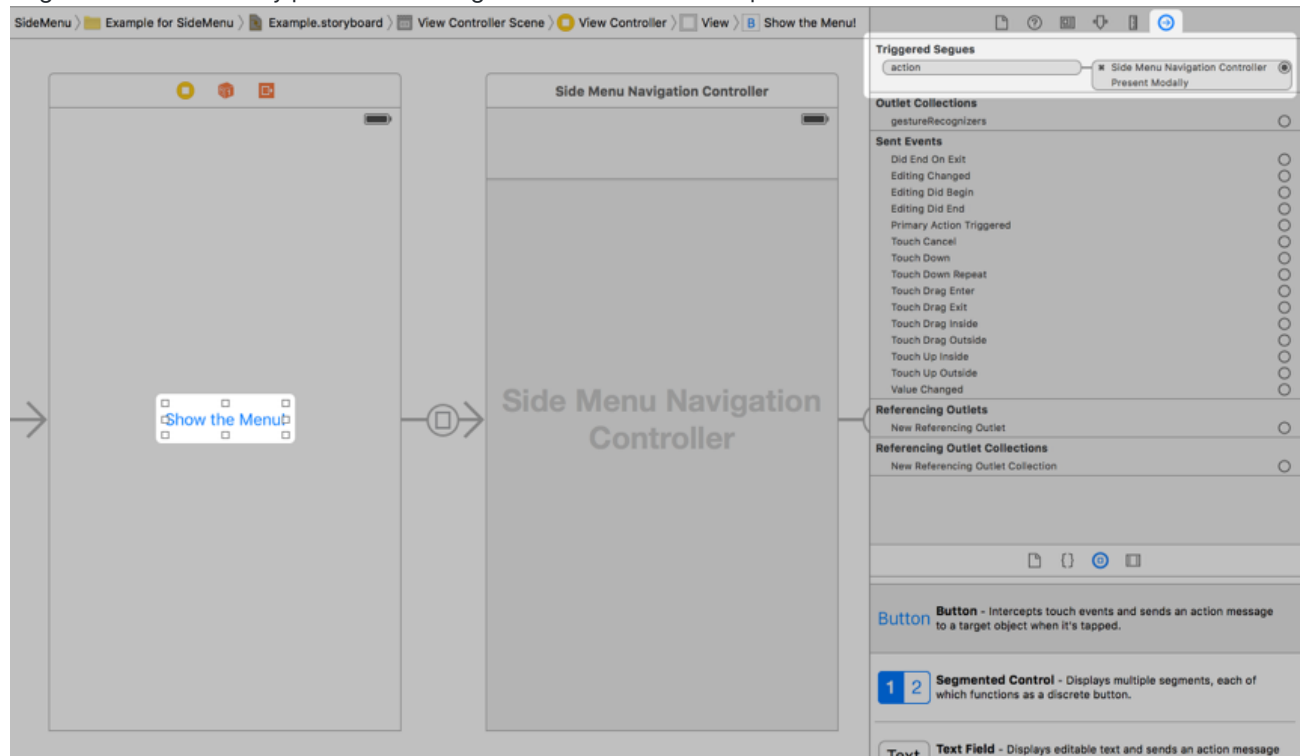
1. Create a Navigation Controller for a side menu. Set the `Custom Class` of the Navigation Controller to be `UISideMenuNavigationController` in the **Identity Inspector**. Set the `Module` to `SideMenu` (ignore this step if you've manually added SideMenu to your project). Create a Root View Controller for the Navigation Controller (shown as a `UITableViewController` below). Set up any Triggered Segues you want in that view controller.



2. Set the `Left Side` property of the `UISideMenuNavigationController` to `On` if you want it to appear from the left side of the screen, or `Off/Default` if you want it to appear from the right side.



3. Add a `UIButton` or `UIBarButtonItem` to a view controller that you want to display the menu from. Set that button's `Triggered Segues` action to modally present the Navigation Controller from step 1.



That's it. *Note: you can only enable gestures in code.*

Code Implementation

First:

```
import SideMenu
```

In your view controller's `viewDidLoad` event, do something like this (**IMPORTANT: If you're seeing a black menu when you use gestures, read this section carefully!**):

```
// Define the menus
let menuLeftNavigationController = UISideMenuNavigationController(rootViewController: YourViewController)
// UISideMenuNavigationController is a subclass of UINavigationController, so do any additional configuration
// of it here like setting its viewControllers. If you're using storyboards, you'll want to do something
// let menuLeftNavigationController = storyboard!.instantiateViewController(withIdentifier: "LeftMenuNavigationController")
SideMenuManager.default.menuLeftNavigationController = menuLeftNavigationController

let menuRightNavigationController = UISideMenuNavigationController(rootViewController: YourViewController)
// UISideMenuNavigationController is a subclass of UINavigationController, so do any additional configuration
// of it here like setting its viewControllers. If you're using storyboards, you'll want to do something
// let menuRightNavigationController = storyboard!.instantiateViewController(withIdentifier: "RightMenuNavigationController")
SideMenuManager.default.menuRightNavigationController = menuRightNavigationController

// Enable gestures. The left and/or right menus must be set up above for these to work.
// Note that these continue to work on the Navigation Controller independent of the view controller it contains.
SideMenuManager.default.menuAddPanGestureToPresent(toView: self.navigationController!.navigationBar)
SideMenuManager.default.menuAddScreenEdgePanGesturesToPresent(toView: self.navigationController!.view)
```

Then from a button, do something like this:

```
present(SideMenuManager.default.menuLeftNavigationController!, animated: true, completion: nil)

// Similarly, to dismiss a menu programmatically, you would do this:
dismiss(animated: true, completion: nil)
```

That's it.

Customization

SideMenuManager

Just type `SideMenuManager.default.menu...` and code completion will show you everything you can customize (for Objective-C, use `SideMenuManager.defaultManager.menu...`). Defaults values are shown below for reference:

```
/**
The push style of the menu.

There are six modes in MenuPushStyle:
- defaultBehavior: The view controller is pushed onto the stack.
- popWhenPossible: If a view controller already in the stack is of the same class as the pushed view controller, it will be popped.
- preserve: If a view controller already in the stack is of the same class as the pushed view controller, it will be preserved.
- preserveAndHideBackButton: Same as .preserve and back buttons are automatically hidden.
- replace: Any existing view controllers are released from the stack and replaced with the pushed view controller.
- subMenu: Unlike all other behaviors that push using the menu's presentingViewController, this behavior pushes the menu as a subview of the presentingViewController.
*/
open var menuPushStyle: MenuPushStyle = .defaultBehavior

/**
The presentation mode of the menu.

There are four modes in MenuPresentMode:
- menuSlideIn: Menu slides in over of the existing view.
- viewSlideOut: The existing view slides out to reveal the menu.
- viewSlideInOut: The existing view slides out while the menu slides in.
- menuDissolveIn: The menu dissolves in over the existing view controller.
*/
open var menuPresentMode: MenuPresentMode = .viewSlideOut
```

```

/// Prevents the same view controller (or a view controller of the same class) from being pushed more th
open var menuAllowPushOfSameClassTwice = true

/**
Width of the menu when presented on screen, showing the existing view controller in the remaining space.

Note that each menu's width can be overridden using the `menuWidth` property on any `UISideMenuNavigatio
*/
open var menuWidth: CGFloat = max(round(min((appScreenRect.width), (appScreenRect.height)) * 0.75), 240)

/// Duration of the animation when the menu is presented without gestures. Default is 0.35 seconds.
open var menuAnimationPresentDuration: Double = 0.35

/// Duration of the animation when the menu is dismissed without gestures. Default is 0.35 seconds.
open var menuAnimationDismissDuration: Double = 0.35

/// Duration of the remaining animation when the menu is partially dismissed with gestures. Default is 0
open var menuAnimationCompleteGestureDuration: Double = 0.35

/// Amount to fade the existing view controller when the menu is presented. Default is 0 for no fade. Se
open var menuAnimationFadeStrength: CGFloat = 0

/// The amount to scale the existing view controller or the menu view controller depending on the `menuF
open var menuAnimationTransformScaleFactor: CGFloat = 1

/// The background color behind menu animations. Depending on the animation settings this may not be vis
open var menuAnimationBackgroundColor: UIColor?

/// The shadow opacity around the menu view controller or existing view controller depending on the `men
open var menuShadowOpacity: Float = 0.5

/// The shadow color around the menu view controller or existing view controller depending on the `menuF
open var menuShadowColor = UIColor.black

/// The radius of the shadow around the menu view controller or existing view controller depending on th
open var menuShadowRadius: CGFloat = 5

/// The left menu swipe to dismiss gesture.
open weak var menuLeftSwipeToDismissGesture: UIPanGestureRecognizer?

/// The right menu swipe to dismiss gesture.
open weak var menuRightSwipeToDismissGesture: UIPanGestureRecognizer?

/// Enable or disable gestures that would swipe to dismiss the menu. Default is true.
open var menuEnableSwipeGestures: Bool = true

/// Enable or disable interaction with the presenting view controller while the menu is displayed. Enabl
open var menuPresentingViewControllerUserInteractionEnabled: Bool = false

/// The strength of the parallax effect on the existing view controller. Does not apply to `menuPresent
open var menuParallaxStrength: Int = 0

/// Draws the `menuAnimationBackgroundColor` behind the status bar. Default is true.
open var menuFadeStatusBar = true

/// The animation options when a menu is displayed. Ignored when displayed with a gesture.
open var menuAnimationOptions: UIViewAnimationOptions = .curveEaseInOut

/// The animation spring damping when a menu is displayed. Ignored when displayed with a gesture.
open var menuAnimationUsingSpringWithDamping: CGFloat = 1

/// The animation initial spring velocity when a menu is displayed. Ignored when displayed with a gestur
open var menuAnimationInitialSpringVelocity: CGFloat = 1

```

```

/**
Automatically dismisses the menu when another view is pushed from it.

Note: to prevent the menu from dismissing when presenting, set modalPresentationStyle = .overFullScreen
of the view controller being presented in storyboard or during its initialization.
*/
open var menuDismissOnPush = true

/// Forces menus to always animate when appearing or disappearing, regardless of a pushed view controller
open var menuAlwaysAnimate = false

/**
The blur effect style of the menu if the menu's root view controller is a UITableViewController or UINavigationController.

- Note: If you want cells in a UITableViewController menu to show vibrancy, make them a subclass of UITableViewCells.
*/
open var menuBlurEffectStyle: UIBlurEffectStyle?

/// The left menu.
open var menuLeftNavigationController: UISideMenuNavigationController?

/// The right menu.
open var menuRightNavigationController: UISideMenuNavigationController?

/**
Adds screen edge gestures to a view to present a menu.

- Parameter toView: The view to add gestures to.
- Parameter forMenu: The menu (left or right) you want to add a gesture for. If unspecified, gestures will be added to both.
- Returns: The array of screen edge gestures added to `toView`.
*/
@discardableResult open func menuAddScreenEdgePanGesturesToPresent(toView: UIView, forMenu: UIRectEdge? = nil) -> [UIScreenEdgeGestureRecognizer]

/**
Adds a pan edge gesture to a view to present menus.

- Parameter toView: The view to add a pan gesture to.
- Returns: The pan gesture added to `toView`.
*/
@discardableResult open func menuAddPanGestureToPresent(toView: UIView) -> UIPanGestureRecognizer

```

UISideMenuNavigationController

UISideMenuNavigationController supports the following customizations and properties:

```

/// SideMenuManager instance associated with this menu. Default is `SideMenuManager.default`. This property is weak.
open weak var sideMenuManager: SideMenuManager! = SideMenuManager.default

/// Width of the menu when presented on screen, showing the existing view controller in the remaining space.
@IBInspectable open var menuWidth: CGFloat = 0

/// Whether the menu appears on the right or left side of the screen. Right is the default. This property is animated.
@IBInspectable open var leftSide: Bool = false

/// Indicates if the menu is anywhere in the view hierarchy, even if covered by another view controller.
open var isHidden: Bool

```

UISideMenuNavigationControllerDelegate

To receive notifications when a menu is displayed from a view controller, have it adhere to the `UISideMenuNavigationControllerDelegate` protocol:

```
extension MyViewController: UISideMenuNavigationControllerDelegate {

    func sideMenuWillAppear(menu: UISideMenuNavigationController, animated: Bool) {
        print("SideMenu Appearing! (animated: \(animated))")
    }

    func sideMenuDidAppear(menu: UISideMenuNavigationController, animated: Bool) {
        print("SideMenu Appeared! (animated: \(animated))")
    }

    func sideMenuWillDisappear(menu: UISideMenuNavigationController, animated: Bool) {
        print("SideMenu Disappearing! (animated: \(animated))")
    }

    func sideMenuDidDisappear(menu: UISideMenuNavigationController, animated: Bool) {
        print("SideMenu Disappeared! (animated: \(animated))")
    }

}
```

Note: setting the `sideMenuDelegate` property on `UISideMenuNavigationController` is optional. If your view controller adheres to the protocol then the methods will be called automatically.

Advanced

For simplicity, `SideMenuManager.default` serves as the primary instance as most projects will only need one menu across all screens. If you need to show a different `SideMenu`, such as from a modal view controller presented from a previous `SideMenu`, do the following:

1. Declare a variable containing your custom `SideMenuManager` instance. You may want it to define it globally and configure it in your app delegate if menus will be used on multiple screens.

```
let customSideMenuManager = SideMenuManager()
```

2. Setup and display menus with your custom instance the same as you would with the `SideMenuManager.default` instance.
3. If using Storyboards, subclass your instance of `UISideMenuNavigationController` and set its `sideMenuManager` property to your custom instance. This must be done before `viewDidLoad` is called:

```
class MySideMenuNavigationController: UISideMenuNavigationController {

    let customSideMenuManager = SideMenuManager()

    override func awakeFromNib() {
        super.awakeFromNib()

        sideMenuManager = customSideMenuManager
    }

}
```

Alternatively, you can set `sideMenuManager` from the view controller that segues to your `UISideMenuNavigationController`:

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
```

```
if let sideMenuNavigationController = segue.destination as? UISideMenuNavigationController {  
    sideMenuNavigationController.sideMenuManager = customSideMenuManager  
}  
}
```

Important: displaying SideMenu instances directly over each other is not supported. Use `menuPushStyle = .subMenu` instead.

Known Issues

- Issue [#258](#).
- Don't try to change the status bar appearance when presenting a menu. When used with quick gestures/animations, it causes the presentation animation to not complete properly and locks the UI. This was fixed in iOS 9.3. See [radar 21961293](#) for more information.

Thank You

A special thank you to everyone that has [contributed](#) to this library to make it better. Your support is appreciated!

License

SideMenu is available under the MIT license. See the LICENSE file for more info.