
Aiolos

Yet another iOS Floating Panel

Carthage compatible

SPM compatible

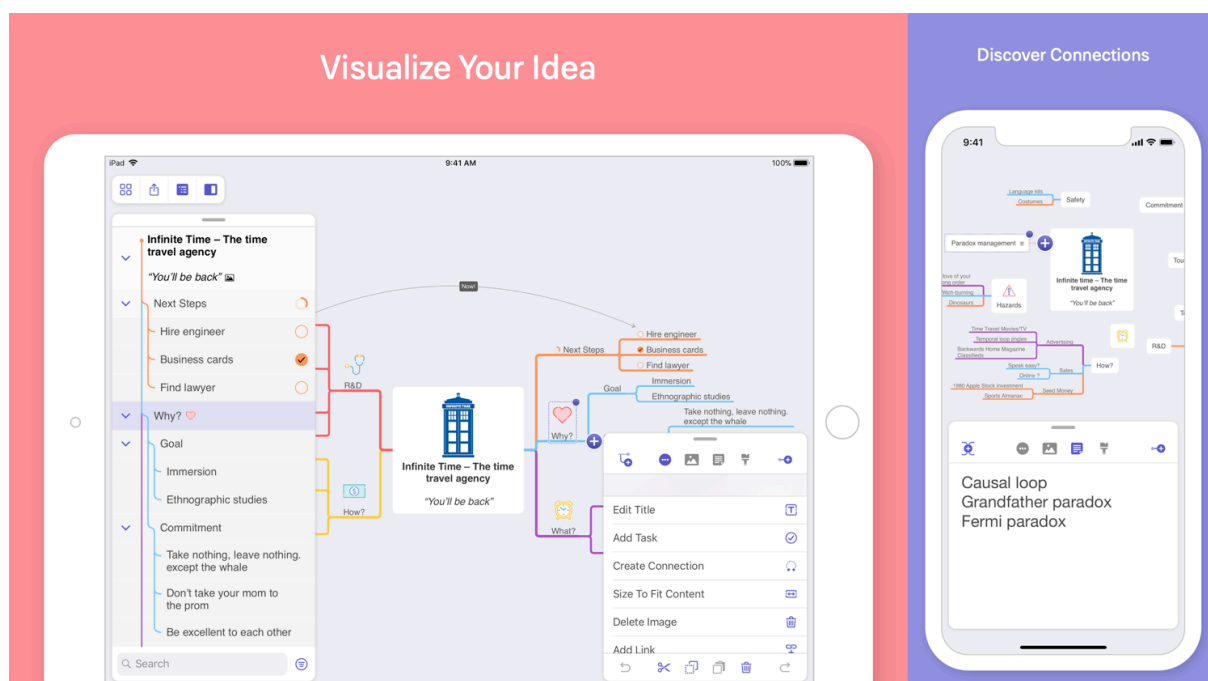
Platform iOS 11+

Language Swift 4.2

Twitter @myell0w

Aiolos, ancient greek for *quick-moving/nimble*, is a Swift UI framework inspired by the floating panel, that was introduced to Maps app in iOS 11. Give it a try in MindNode 5 for iOS (free trial available).

It is fully **gesture-driven**, takes **safe area** insets into account, has support for **right-to-left languages** baked in and automatically reacts to the on-screen **keyboard**. Compared to many other open source panel solutions, Aiolos is designed to be an always-visible child view controller, and therefore does not use the custom view controller transition API of iOS.



Integration with Carthage

Add this line to your Cartfile.

```
1 github "IdeasOnCanvas/Aiolos"
```

Integration with Swift Package Manager

Aiolos can be integrated with Swift Package Manager directly within Xcode.

Usage in Code

There's a demo app, that demonstrates how the Panel can be set up with a different configuration for iPhones and iPads.

```
1 func makePanel(with viewController: UIViewController) -> Panel {
2     // create Panel with default configuration
3     let configuration = Panel.Configuration.default
4     let panelController = Panel(configuration: configuration)
5
6     // specify, which ViewController is displayed in the panel
7     panelController.contentViewController = viewController
8
9     // setup delegates that handle size configuration and animation
    callbacks
10    panelController.sizeDelegate = self
11    panelController.animationDelegate = self
12
13    // change the configuration to fit you needs
14    panelController.configuration.position = self.panelPosition(for:
        self.traitCollection)
15    panelController.configuration.margins = self.panelMargins(for: self
        .traitCollection)
16    panelController.configuration.appearance.separatorColor = .white
17
18    // we want a different look/behaviour on iPhone compared to iPad
19    if self.traitCollection.userInterfaceIdiom == .pad {
20        panelController.configuration.appearance.maskedCorners = [.
            layerMinXMinYCorner, .layerMaxXMinYCorner, .
            layerMinXMaxYCorner, .layerMaxXMaxYCorner]
21    } else {
22        panelController.configuration.supportedModes = [.minimal, .
            compact, .expanded, .fullHeight]
23        panelController.configuration.appearance.maskedCorners = [.
            layerMinXMinYCorner, .layerMaxXMinYCorner]
24    }
25
26    return panelController
27 }
```

Configuring the size

```
1 extension ViewController: PanelSizeDelegate {
2
3     func panel(_ panel: Panel, sizeForMode mode: Panel.Configuration.
        Mode) -> CGSize {
4         let width = self.panelWidth(for: self.traitCollection, position
            : panel.configuration.position)
```

```
5     switch mode {
6     case .minimal:
7         return CGSize(width: width, height: 0.0)
8     case .compact:
9         return CGSize(width: width, height: 64.0)
10    case .expanded:
11        let height: CGFloat = self.traitCollection.
            userInterfaceIdiom == .phone ? 270.0 : 320.0
12        return CGSize(width: width, height: height)
13    case .fullHeight:
14        return CGSize(width: width, height: 0.0)
15    }
16 }
17 }
```

Reacting to Panel animations

```
1  extension ViewController: PanelAnimationDelegate {
2
3      func panel(_ panel: Panel, willTransitionTo size: CGSize) {
4          print("Panel will transition to size \(size)")
5      }
6
7      func panel(_ panel: Panel, willTransitionFrom oldMode: Panel.
            Configuration.Mode?, to newMode: Panel.Configuration.Mode, with
            coordinator: PanelTransitionCoordinator) {
8          print("Panel will transition from \(oldMode) to \(newMode)")
9          // we can animate things along the way
10         coordinator.animateAlongsideTransition({
11             print("Animating alongside of panel transition")
12         }, completion: { animationPosition in
13             print("Completed panel transition to \(newMode)")
14         })
15     }
16 }
```

Credits

Aiolos is brought to you by IdeasOnCanvas GmbH, the creator of MindNode for iOS, macOS & watchOS