From UI Design to UIKit

How do I build UI?

No Interface Builder

No Interface Builder

Helpers for AutoLayout

Helpers for AutoLayout

Debugging using Reveal.app

Debugging using Reveal.app

Using UILayoutGuides

Bundle Identifier: me.vdka.FromUIDesignToUIKit

Version: 1.0

Build: 1

Signing

Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team: Ethan Jackwitz (Personal Team - ethanjackwit...

Provisioning Profile: Xcode Managed Profile

Signing Certificate: iPhone Developer: ethanjackwitz@gmail.com (MN...

Deployment Info

Deployment Target: 11.3

Devices: Universal

Main Interface: Main

Device Orientation:
- [x] Portrait
- [ ] Upside Down
- [x] Landscape Left
- [x] Landscape Right

Status Bar Style: Default

- [ ] Hide status bar
- [ ] Requires full screen

App Icons and Launch Images

App Icons Source: AppIcon

Launch Images Source: Use Asset Catalog...

Launch Screen File: LaunchScreen

Embedded Binaries

Bundle Identifier: me.vdka.FromUIDesignToUIKit

Version: 1.0

Build: 1

Signing

Automatically manage signing

Xcode will create and update profiles, app IDs, and certificates.

Team: Ethan Jackwitz (Personal Team - ethanjackwit...

Provisioning Profile: Xcode Managed Profile

Signing Certificate: iPhone Developer: ethanjackwitz@gmail.com (MN...

Deployment Info

Deployment Target: 11.3

Devices: Universal

Main Interface:

Device Orientation:

- [x] Portrait
- [ ] Upside Down
- [x] Landscape Left
- [x] Landscape Right

Status Bar Style: Default

- [ ] Hide status bar
- [ ] Requires full screen

App Icons and Launch Images

App Icons Source: AppIcon

Launch Images Source: Use Asset Catalog...

Launch Screen File: LaunchScreen

Embedded Binaries

Helpers

```
extension UIView {

    func addSubviewEnablingAutoLayout(_ subview: UIView) {
        self.addSubview(subview)
        subview.translatesAutoresizingMaskIntoConstraints = false
    }
}
```

Helpers

```
extension NSLayoutConstraint {

    func withPriority(_ priority: UILayoutPriority) -> NSLayoutConstraint {
        self.priority = priority
        return self
    }
}

extension UILayoutPriority: ExpressibleByIntegerLiteral {

    public init(integerLiteral value: Int) {
        self.init(rawValue: Float(value))
    }
}
```

Helpers

```
protocol ReusableView {
    static var reuseIdentifier: String { get }
}

extension ReusableView where Self: UIView {
    static var reuseIdentifier: String {
        return String(describing: self)
    }
}
```

Helpers

```
extension UITableView {

    // register for the Class-based cell
    func register<T: UITableViewCell>(_: T.Type) {
        register(T.self, forCellReuseIdentifier: T.reuseIdentifier)
    }

    func dequeueReusableCell<T: UITableViewCell>(forIndexPath indexPath: IndexPath) -> T {
        guard let cell = dequeueReusableCell(withIdentifier: T.reuseIdentifier, for: indexPath) as? T else {
            fatalError(" ... ")
        }
        return cell
    }

    // register for the Class-based header/footer view
    func register<T: UITableViewHeaderFooterView>(_: T.Type) {
        register(T.self, forHeaderFooterViewReuseIdentifier: T.reuseIdentifier)
    }

    func dequeueReusableView<T: UITableViewHeaderFooterView>() -> T? {
        let v = dequeueReusableHeaderFooterView(withIdentifier: T.reuseIdentifier) as? T
        return v
    }
}
```

Helpers

```
extension UITableViewCell: ReusableView {}
extension UITableViewHeaderFooterView: ReusableView {}
```

Reveal

revealapp.com

Peek

github.com/shaps80/Peek