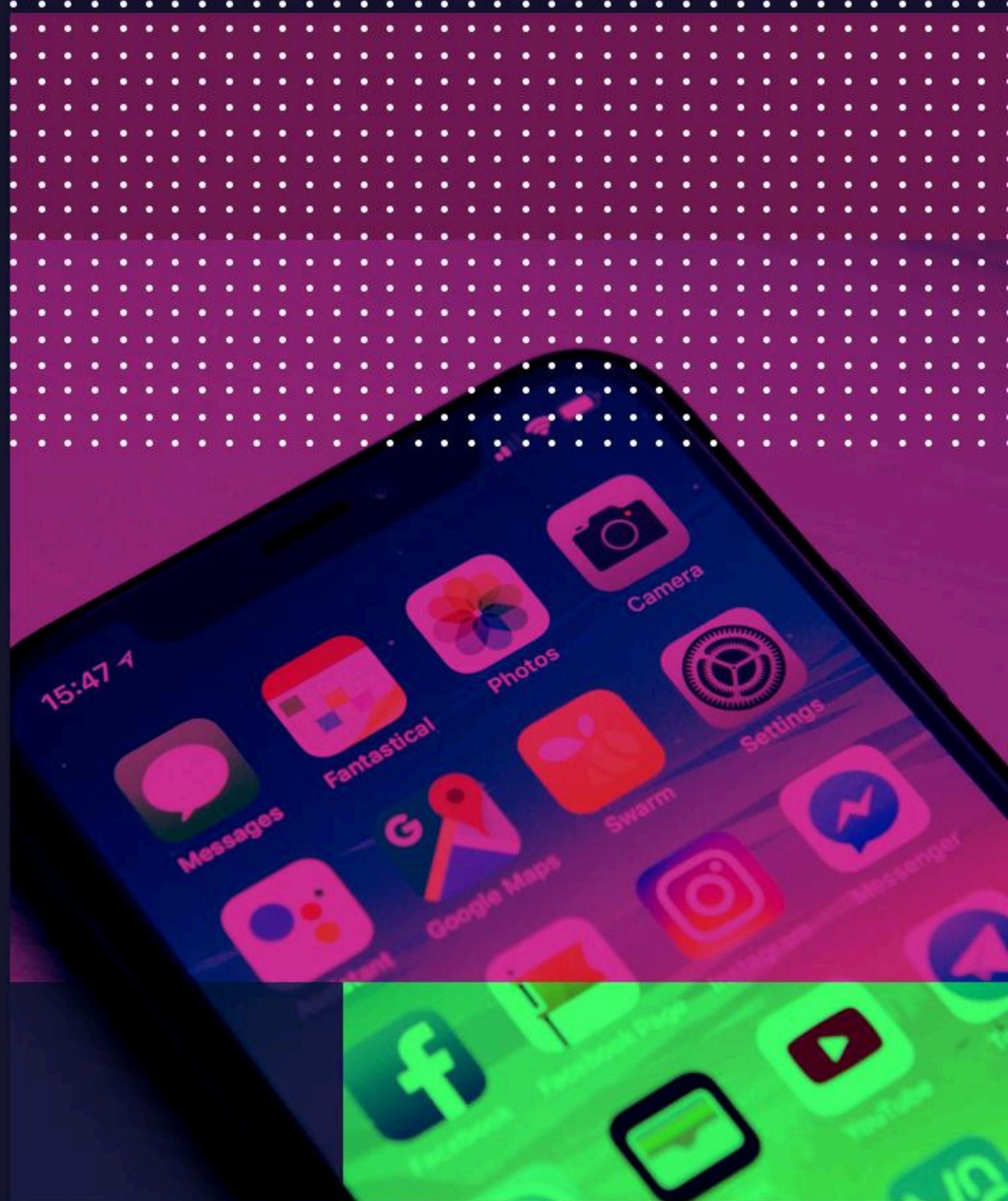


WORKSHOP

# How to build an iOS app completely from scratch



# Kristaps

- Co-founder, Swift developer at Qminder
- Speaker and mentor
- Open source and inclusivity



# Alina

- Swift developer at Mapon
- Open-minded
- Explorer



# Kārlis

- Mobile developer at Accenture
- iOS guest lecturer at Vidzeme University
- Good guy

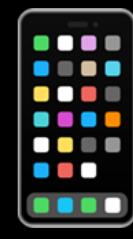


# Mr. Byte



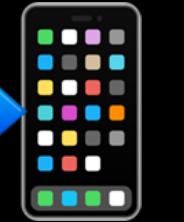
# Plan

What is iOS development?



Lunch 

Build an app →



January 9, 2007



**2.2 billion**

**2.2 million**

**120\$ billion**



Mac



iPhone



Apple Watch



iPad



Apple TV

# Swift

June 2, 2014 / December 3, 2015

Developed by Apple

General purpose language

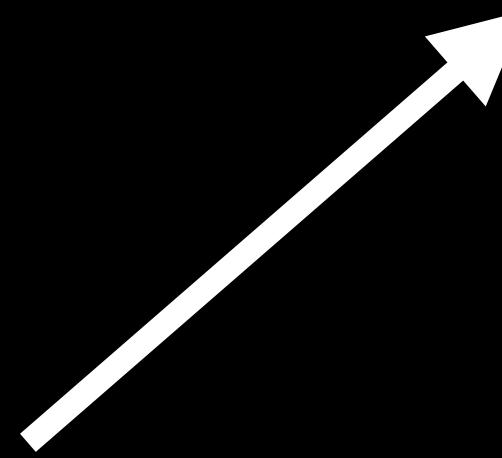


*“A programming language is a vocabulary  
and set of grammatical rules for instructing a  
computer or computing device to perform  
specific tasks.”*

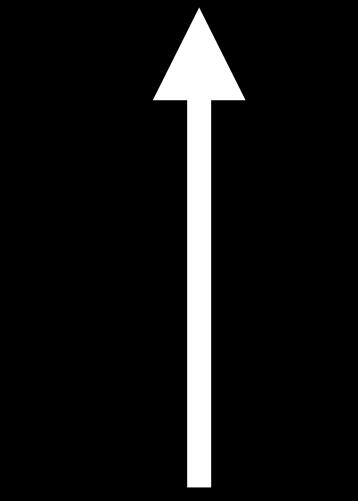
# *Programming ABC*

# Variables

```
var x: Int = 1
```



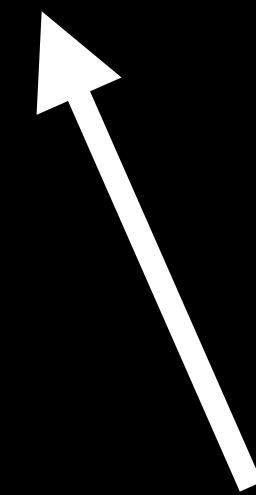
**keyword**



**name**



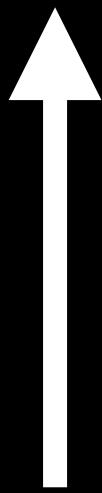
**data type**



**value**

# Constants

```
let pi: Double = 3.14
```



**keyword**

# Different variables

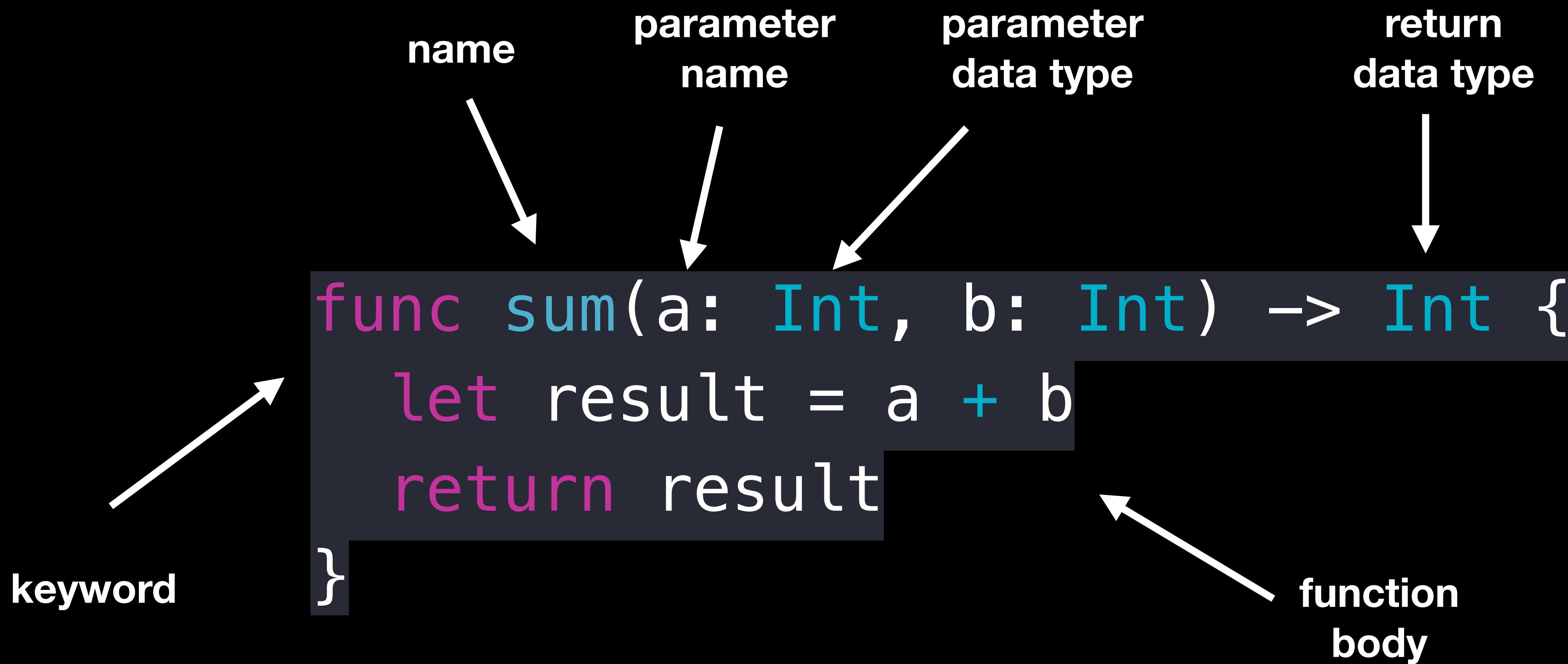
```
var number: Int = 13
```

```
var decimal: Float = 2.76
```

```
var name: String = "RigaTechGirls 🧑💻"
```

```
var isItSpringNow: Bool = false
```

# Functions



# Functions

```
func sayHello(name: String) {  
    let helloText = "Hello \(name)"  
    print(helloText)  
}
```

name → parameter name → parameter data type

keyword → function body

# Conditionals

```
if day == "Monday" {  
    // work hard  
}  
else if day == "Friday" {  
    // play hard  
}  
else {  
    // rest  
}
```

condition

keyword

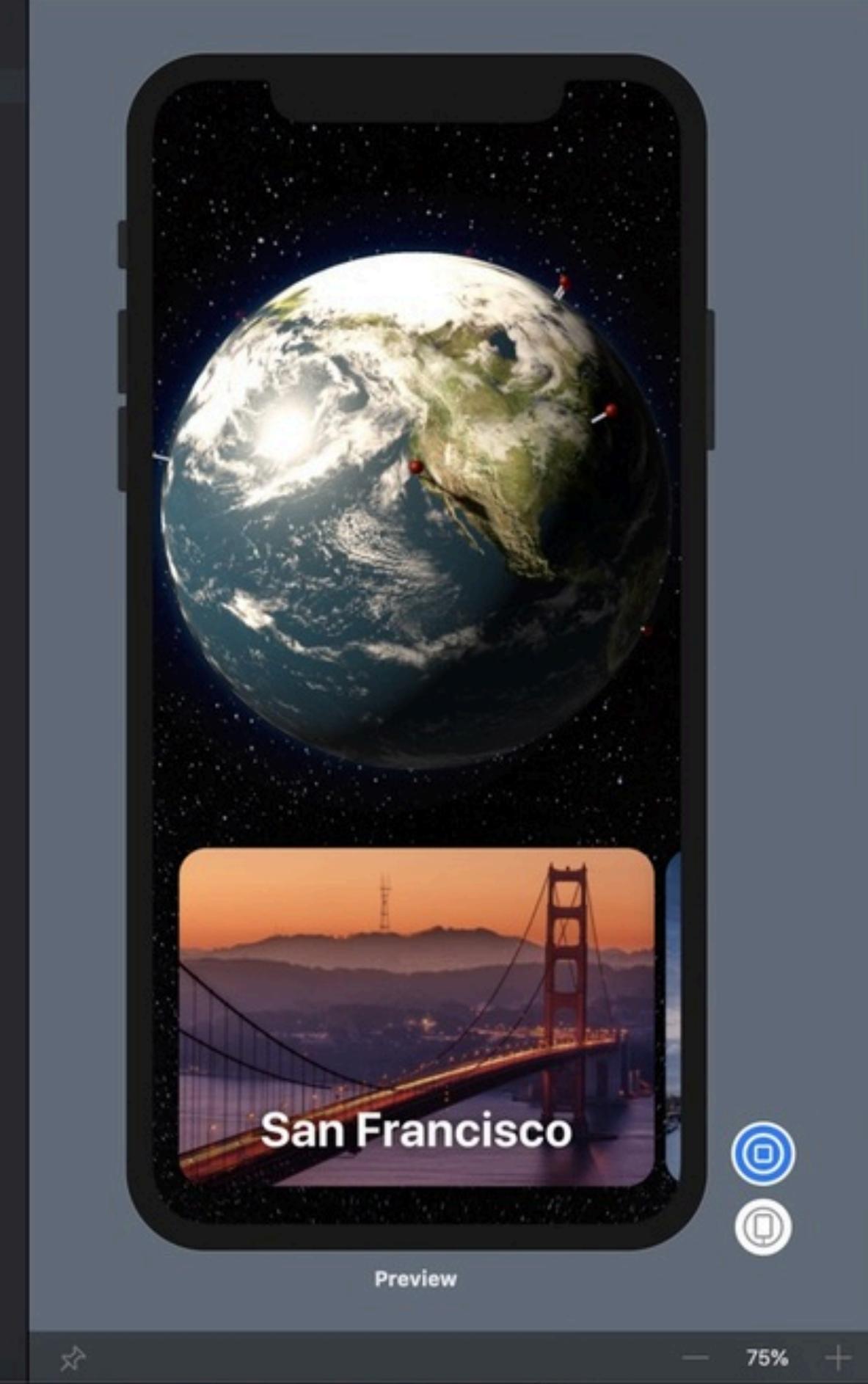
# *Tools*

# Xcode

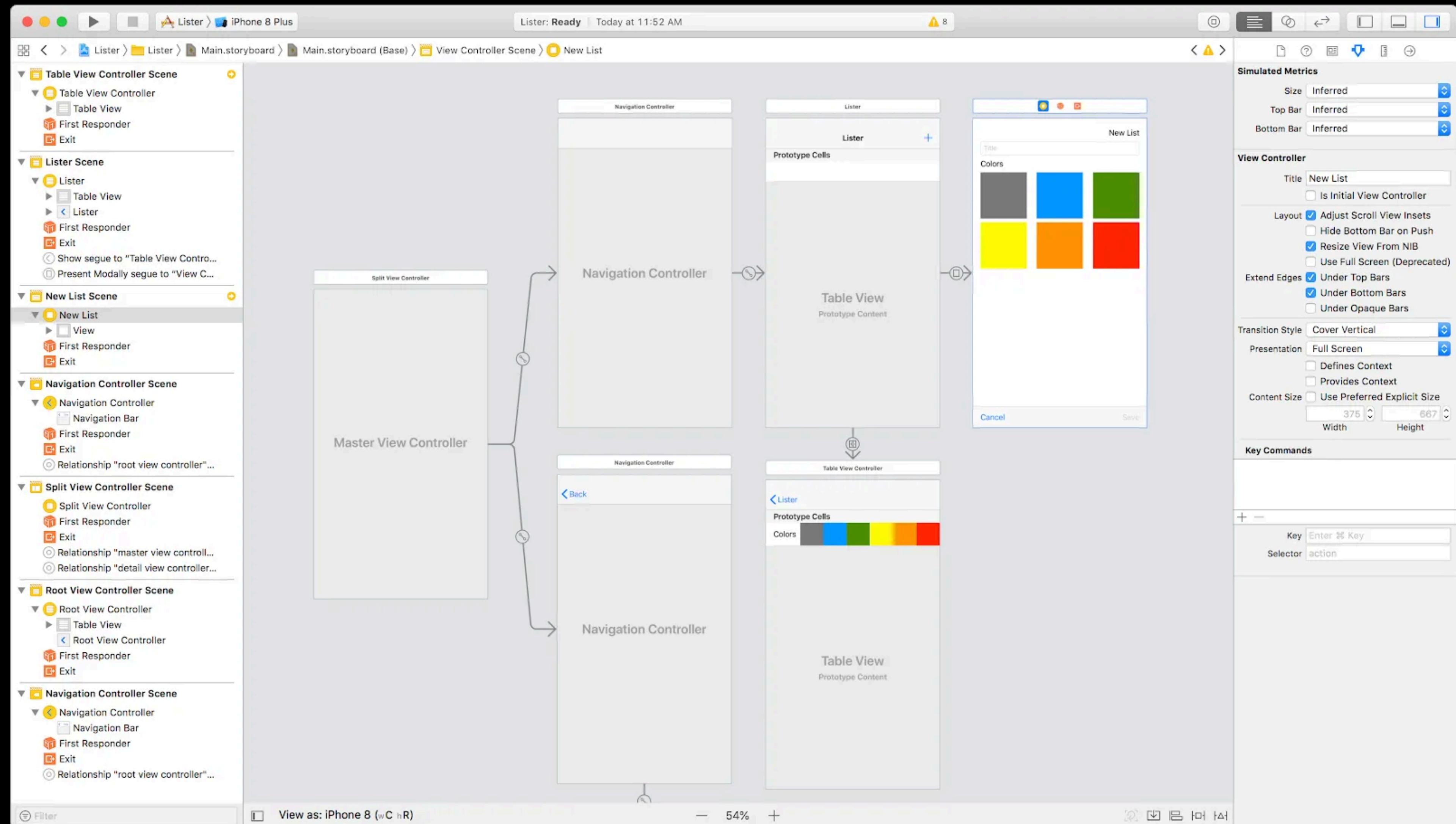
The screenshot shows the Xcode interface with a project named "Travel". The left sidebar displays the file structure:

- Travel (root folder)
  - ReadMe.md
  - Travel (folder)
    - Travel.entitlements
    - AppDelegate.swift
    - Model (folder)
    - Discover (folder)
      - DiscoverView.swift
      - DiscoverTileView.swift
      - GlobeView.swift
      - DestinationsListView.swift
      - PagingScrollView.swift
      - CardsController.swift
    - Globe Scene (folder)
    - Plan (folder)
    - Journal (folder)
      - Journal.storyboard
      - StoryboardHostView.swift
      - JournalView.swift
      - JournalViewController.swift
      - JournalViewCell.swift
      - JournalTableController.swift
      - JournalDetailTableViewCell.swift
      - JournalPreviewViewCell.swift
      - JournalHeaderView.swift
      - JournalAddController.swift
      - JournalImagePickerController.swift
      - RoundedCornerButton.swift
  - Weather (folder)
  - Login Screen (folder)
    - Login.storyboard
    - LoginViewController.swift
    - ForgotPasswordController.swift
    - ForgotPasswordController.xib
    - ForgotPasswordStatusView.swift
  - Main Screen (folder)

```
8 import SwiftUI
9
10 struct DiscoverView : View {
11     let sceneController: GlobeSceneController
12     @State private var selection: Region? = nil
13
14     var body: some View {
15         let pagingScrollViewController =
16             sceneController.pagingScrollViewController
17         pagingScrollViewController.didChangeToPageHandler = { page in
18             self.selection = DataSource.shared.regions[page]
19         }
20
21         return GeometryReader { container in
22             return ZStack(alignment: .bottom) {
23                 GlobeView(
24                     selection: self.$selection,
25                     sceneController: self.sceneController
26                 )
27
28                 PagingTilesView(
29                     containerSize: container.size,
30                     pagingScrollViewController: pagingScrollViewController
31                 ) { region in
32                     self.selection = region
33                 }
34             }
35             .background(Color.black)
36         }
37     }
38
39 struct PagingTilesView<T> : View where T : PagingScrollViewController {
40     let containerSize: CGSize
41     let pagingScrollViewController: T
42     var selectedTileAction: (Region) -> ()
43
44     var body: some View {
45         let tileSize = containerSize.width * 0.9
46         let tileHeight = CGFloat(240.0)
47         let verticalTileSpacing = CGFloat(8.0)
```



# User Interface



# iOS Simulator

The screenshot shows the Xcode interface with the project "Solar System iOS" open. The main editor displays the `MainViewController.swift` file, which contains Swift code for managing a planet details view controller. The code includes imports for `UIKit`, `SceneHUDDelegate`, and `PlanetDetailsVCDelegate`. It defines a `MainViewController` class that implements these protocols. The `viewDidLoad` method initializes a `SceneViewController` and adds its view to a container view. The `SceneViewController` has constraints relative to the container view's anchors. The `self.solarSystemVC = solarSystemVC` line at the bottom is commented out.

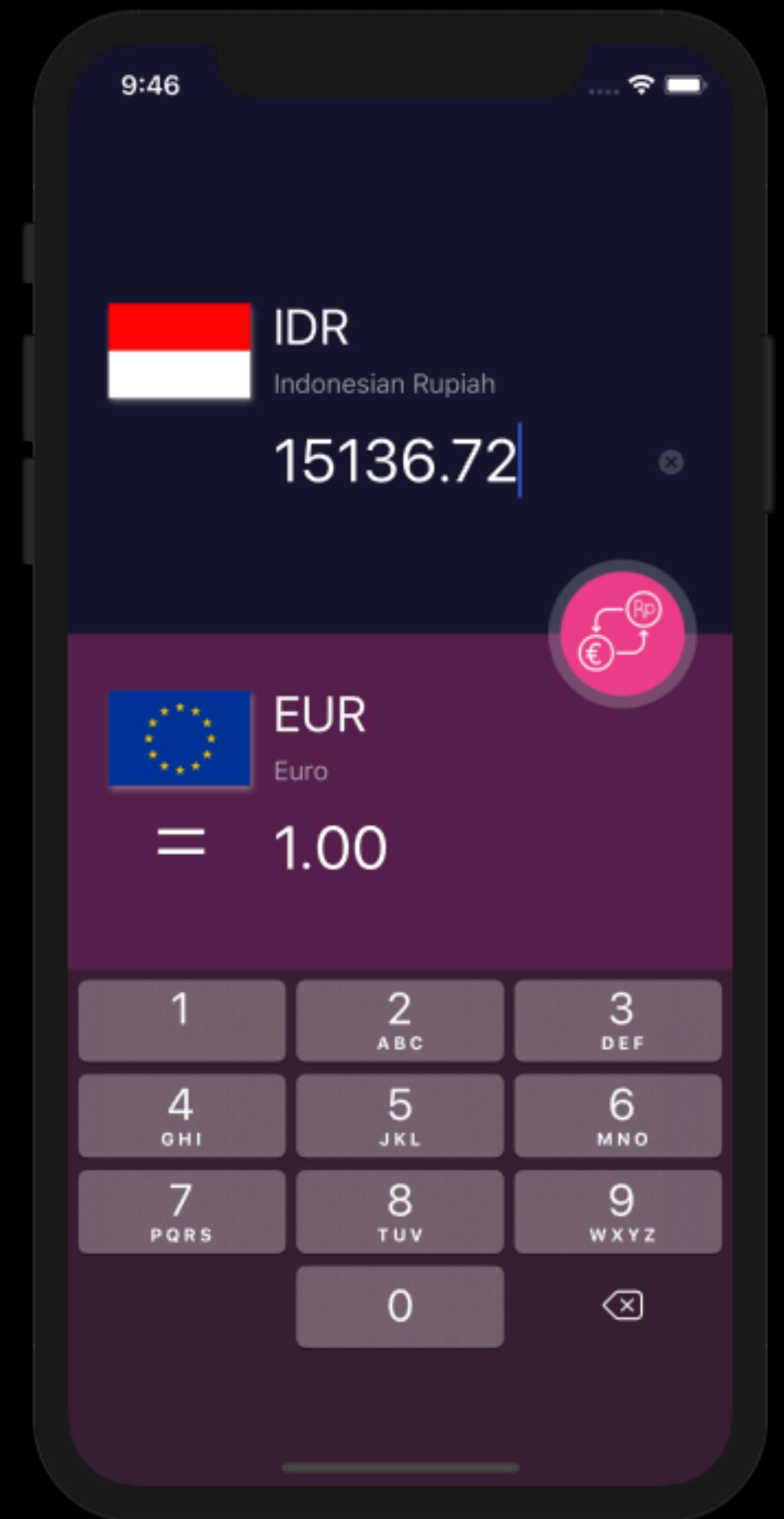
```
7
8 import UIKit
9
10 class MainViewController: UIViewController, SceneHUDDelegate,
11     PlanetDetailsVCDelegate {
12
13     @IBOutlet weak var contentContainerView: UIView!
14     weak var sceneHUDController: SceneHUDViewController?
15
16     weak var solarSystemVC: SceneViewController?
17     weak var planetDetailsVC: SceneDetailsController?
18
19     override func viewDidLoad() {
20         super.viewDidLoad()
21
22         title = "The Solar System"
23
24         // Initiate solar system controller
25         let solarSystemVC =
26             storyboard!.instantiateViewController(withIdentifier:
27             "solarSystemVC") as! SceneViewController
28
29         solarSystemVC.view.translatesAutoresizingMaskIntoConstraints =
30             false
31         addChild(solarSystemVC)
32         contentContainerView.addSubview(solarSystemVC.view)
33
34         // Constraints
35         solarSystemVC.view.leadingAnchor.constraint(equalTo:
36             contentContainerView.leadingAnchor).isActive = true
37         solarSystemVC.view.trailingAnchor.constraint(equalTo:
38             contentContainerView.trailingAnchor).isActive = true
39         solarSystemVC.view.topAnchor.constraint(equalTo:
40             contentContainerView.topAnchor).isActive = true
41         solarSystemVC.view.bottomAnchor.constraint(equalTo:
42             contentContainerView.bottomAnchor).isActive = true
43
44         self.solarSystemVC = solarSystemVC
45 }
```

The right side of the screenshot shows the iOS Simulator running the "Planets" application. The app displays a list of planets with their names and brief descriptions. The visible items are:

- Mercury**: Mercury is the smallest and innermost planet in the Solar System. Its orbital period around the Sun is approximately 88 Earth days.
- Venus**: Venus is the second planet from the Sun, orbiting it every 224.7 Earth days. It rotates in the opposite direction to most other planets.
- Earth**: Earth, otherwise known as the World or the Globe, is the third planet from the Sun and the only one known to harbor life.
- Mars**: Mars is the fourth planet from the Sun and the second-smallest planet in the Solar System.
- Jupiter**: Jupiter is the fifth planet from the Sun and the largest in the Solar System. It is a giant planet with many moons.

The simulator interface includes a "Done" button, a back arrow, and a search icon. The status bar at the bottom indicates "iPhone X - 12.0".

*Let's build an app*



# Step 1 - Create project

New Xcode project

Select iOS Simulator

Run empty app

# Step 2 - Assets

<http://bit.ly/rtg-ios>

# Step 2' - Assets

Download assets

Add to Asset Catalog

Dark mode

# Step 2' - Colors

dark-purple

#FFFFFF

#14132F

light-purple

#932574 opacity 50%

#ED408A opacity 30%

# Step 3 - User Interface

Text labels

Inputs

Flags

Buttons

# Step 4 - Layout elements

Put elements on the screen

Constraints

Screen resolutions

# Step 5 - Programming

Convert function from IDR to EUR

Format number

# Step 6 - Connect UI

IBOutlets

IBActions

Connect interface to code

# Step 7 - Advanced I

Convert automatically

# Step 8 - Advanced II

Convert vice versa

# Step 9 - Extra

Build currency selection

VDN has more crazy rate

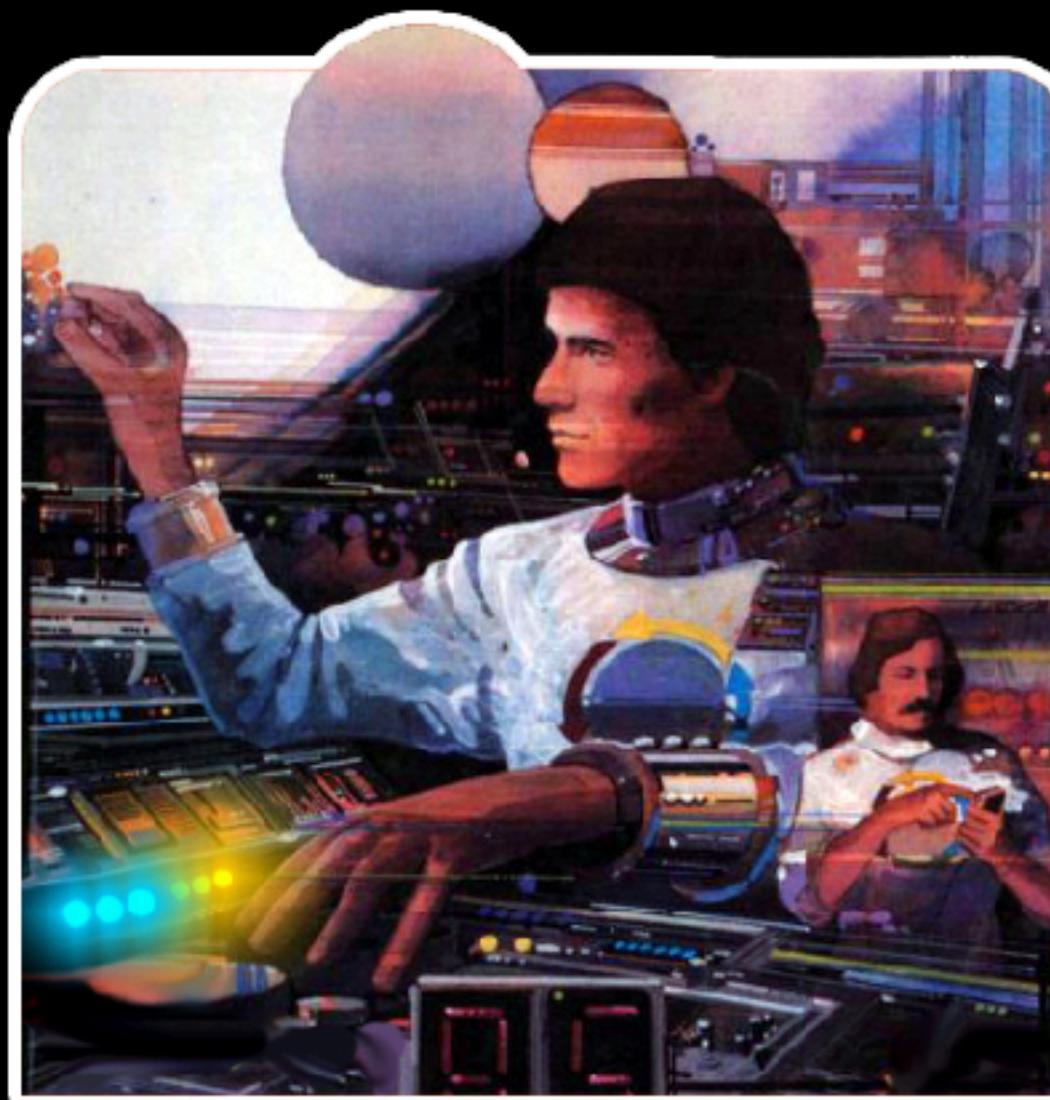
# What's next?

Treehouse

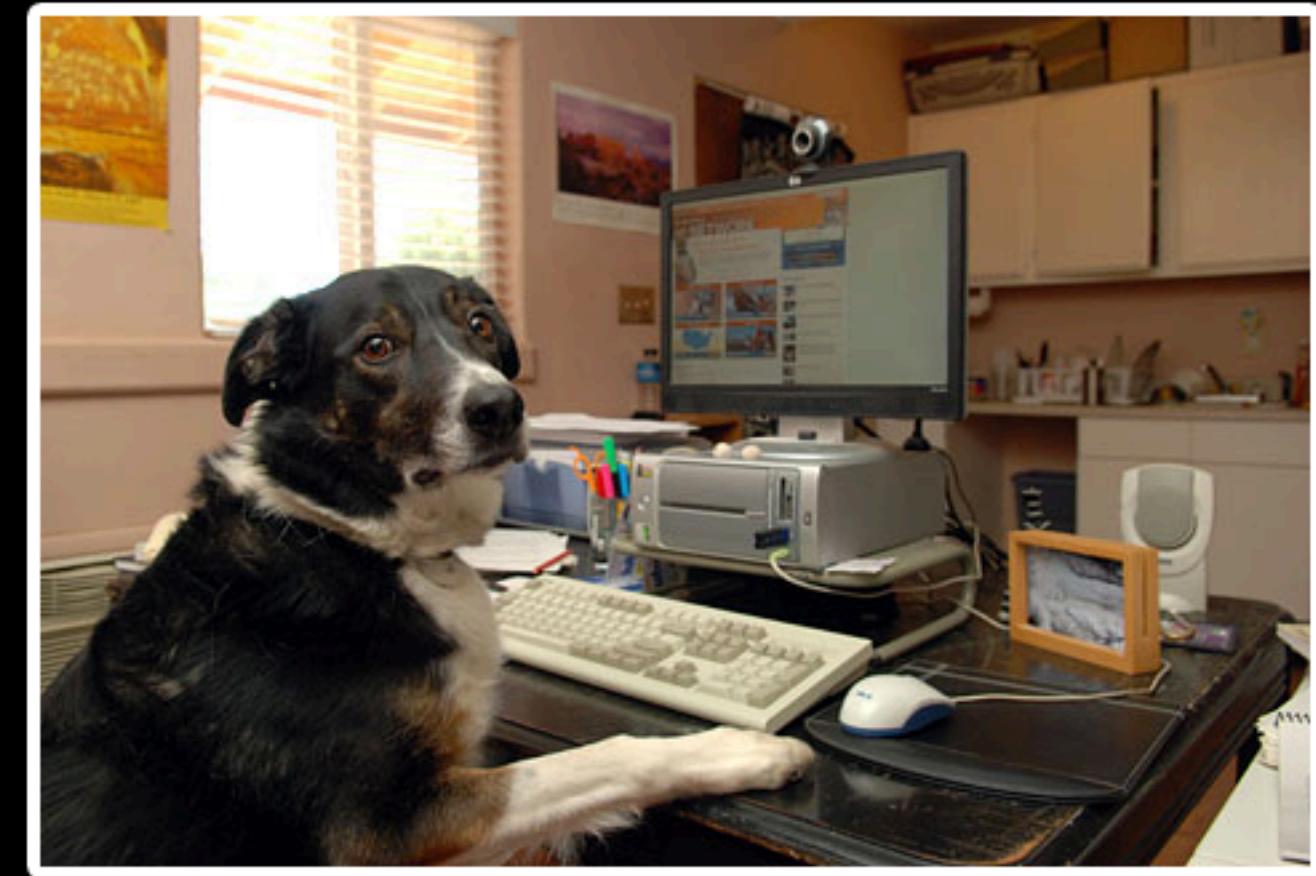
Hacking with Swift

Just start building

# THE TWO STATES OF EVERY PROGRAMMER

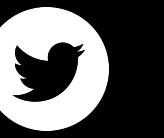


I AM A GOD.



I HAVE NO IDEA  
WHAT I'M DOING.

@georgeb3dr



fassko

[www.kristaps.me](http://www.kristaps.me)

