



Social Media Apps Programming

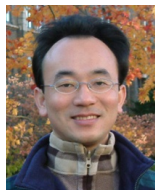
Developing iPhone / iPad Native Apps with Swift (XCode)

1061SMAP04

TLMXM1A (8648) (M2143) (Fall 2017)

(MIS MBA) (2 Credits, Elective) [Full English Course]

Fri 8,9 (15:10-17:00) B206



Min-Yuh Day, Ph.D.

Assistant Professor

Department of Information Management

Tamkang University

<http://mail.tku.edu.tw/myday>



Course Schedule (1/2)



Week	Date	Subject/Topics
1	2017/09/22	Course Orientation and Introduction to Social Media and Mobile Apps Programming
2	2017/09/29	Introduction to Android / iOS Apps Programming
3	2017/10/06	Developing Android Native Apps with Java (Android Studio)
4	2017/10/13	Developing iPhone / iPad Native Apps with Swift (XCode)
5	2017/10/20	Mobile Apps using HTML5/CSS3/JavaScript
6	2017/10/27	jQuery Mobile
7	2017/11/03	Create Hybrid Apps with Phonegap
8	2017/11/10	jQuery Mobile/Phonegap
9	2017/11/17	jQuery Mobile/Phonegap

Course Schedule (2/2)



Week	Date	Subject/Topics
10	2017/11/24	Midterm Project Report
11	2017/12/01	Case Study on Social Media Apps Programming and Marketing in Google Play and App Store
12	2017/12/08	Google Cloud Platform
13	2017/12/15	Google App Engine
14	2017/12/22	Google Map API
15	2017/12/29	Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)
16	2018/01/05	Twitter API
17	2018/01/12	Final Project Presentation
18	2018/01/19	Final Exam Week (Final Project Presentation)

Android /iOS Apps Programming

Native Apps

Hybrid Apps

Mobile Web Apps



App Development Comparison



Native Apps	Full	Very Fast	Expensive	Available	Mandatory
--------------------	------	-----------	-----------	-----------	-----------

Hybrid Apps	Full	Native Speed as Necessary	Reasonable	Available	Low Overhead
--------------------	------	---------------------------	------------	-----------	--------------

Web Apps	Partial	Fast	Reasonable	Not Available	None
-----------------	---------	------	------------	---------------	------

Outline

- **Developing iPhone / iPad Native Apps with Swift 4 (Xcode 9)**
 - Mac OS X 10.8, 10.9, 10.10, 10.11, 10.12, 10.13
 - Xcode 6, Xcode 7, Xcode 8, Xcode 9
 - iOS 8, iOS 9, iOS 10, iOS 11
- **Building Your First iOS App with Xcode 9**



Xcode 8



Swift 3



Xcode 9



Swift 4

Building Your First iOS App with Xcode 9

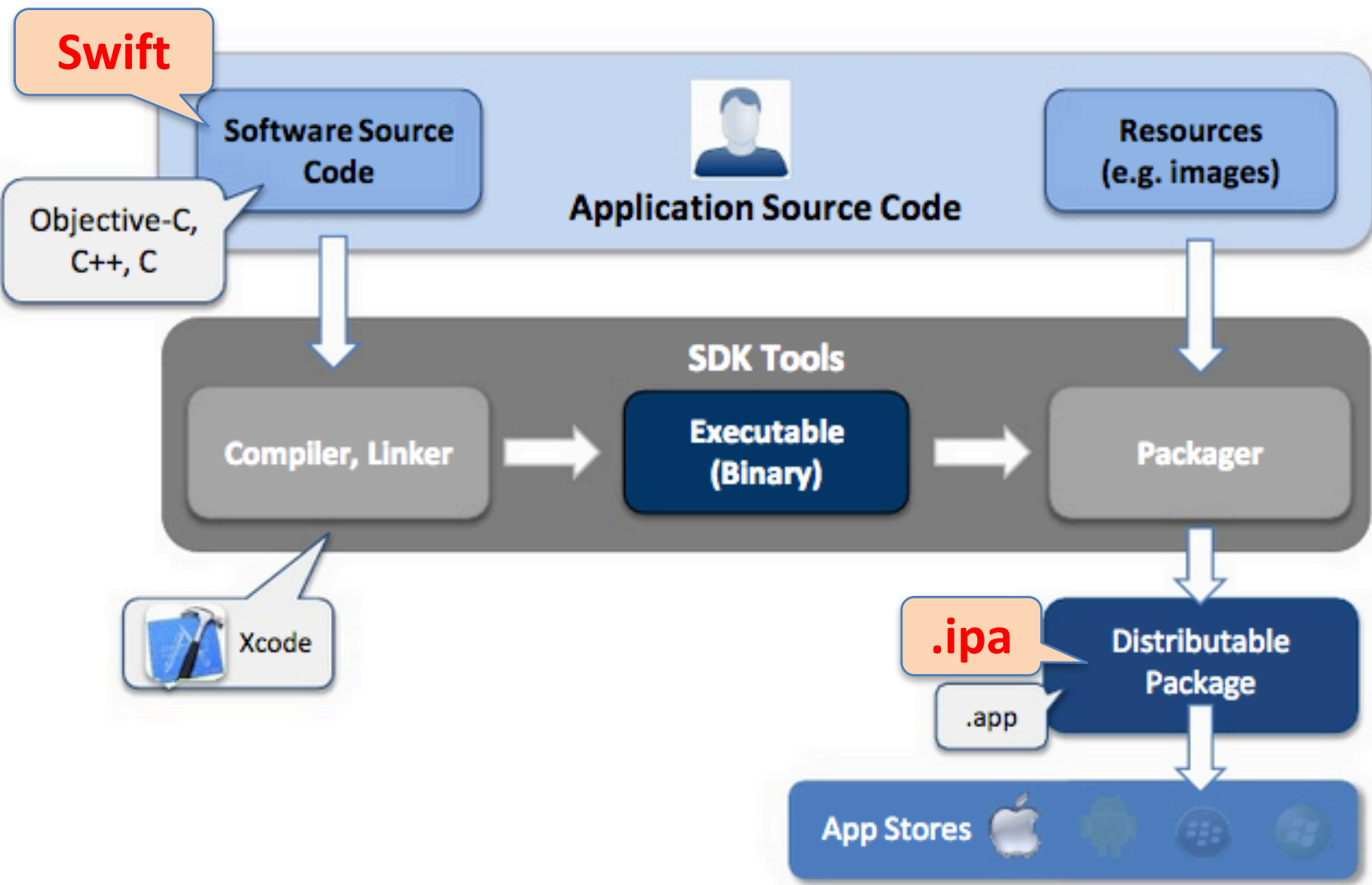
The image shows the Xcode 9 interface. On the left, the iPhone 7 simulator is running the app. The status bar at the top of the simulator shows 'Carrier', signal strength, Wi-Fi, and battery. The time is 10:36 AM. The text input field contains 'Myday'. Below the input field, the text 'Hello, Myday' is displayed in orange, and 'Hello' is displayed in blue below it. A keyboard is visible at the bottom of the simulator, with 'Myday' in the search bar.

In the center, the storyboard is visible, showing a white background with a text input field containing 'Hello World' and a blue 'Hello' label below it.

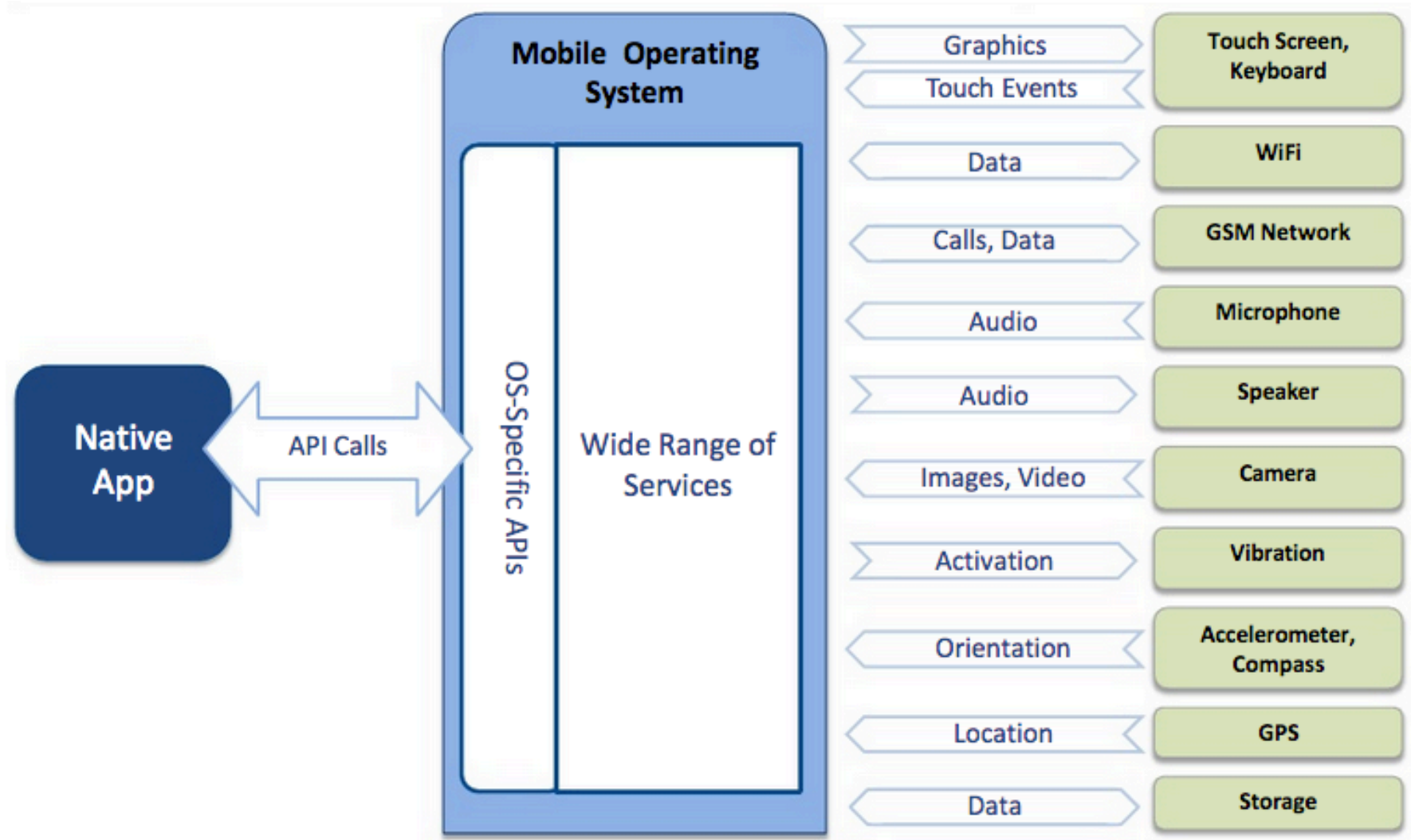
On the right, the Swift code editor is open to 'ViewController.swift'. The code is as follows:

```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
    @IBOutlet weak var txtYourName: UITextField!  
  
    @IBOutlet weak var myLabel: UILabel!  
  
    @IBAction func btnHello(_ sender: AnyObject) {  
        let strYourName:String! = txtYourName.text  
        myLabel.text = "Hello, " + strYourName  
        txtYourName.text = ""  
    }  
}
```

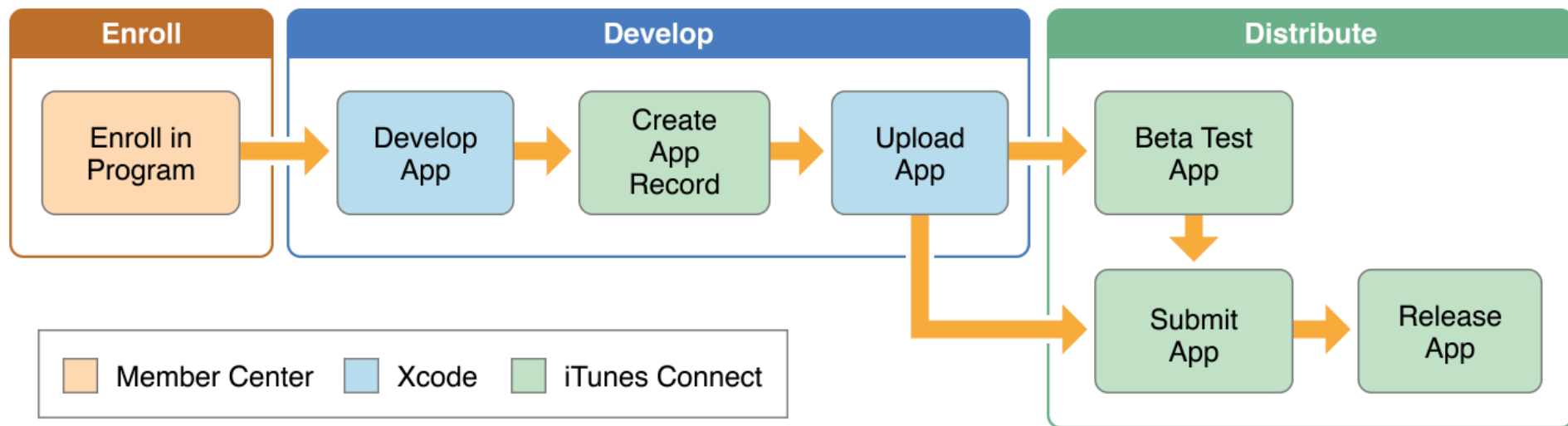
iOS - Native App Development



Native App – Interaction with Mobile Device



Apple App Distribution Workflows



developer.apple.com

 Developer

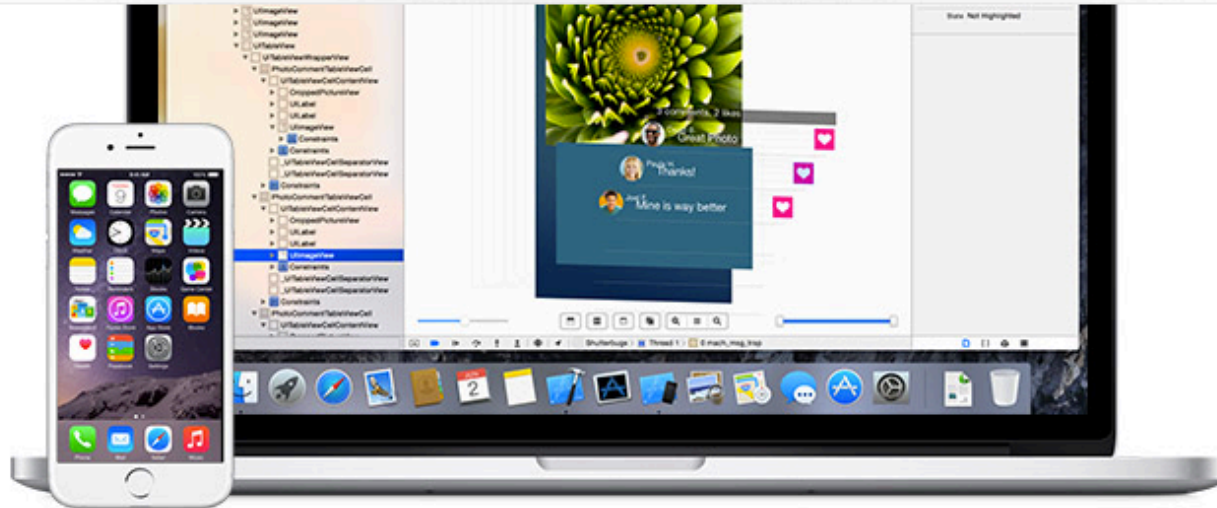
Technologies

Resources

Programs

Support

Member Center



See what's new for developers.

Learn about all the new technologies and powerful capabilities available in iOS 8, OS X Yosemite, and the new programming language, Swift, available in Xcode 6.

developer.apple.com

Download the latest development tools and SDKs.



iOS Apps



Mac Apps



Xcode

developer.apple.com

Become a member.

Enroll in an Apple Developer Program and get everything you need to develop and distribute your apps.

[Learn more >](#)



Apress
For professionals by professionals

Your Account | Your Cart | Sign In

Hot Topics: .NET | iPhone | Android | Web Development | Show All Categories

Search apress.com... Go

Home | Alpha Program | Springer | Source Code | About Apress | Support | Interact with Apress

Twitter | Facebook

Home » iPhone and iPad Apps for Absolute Beginners

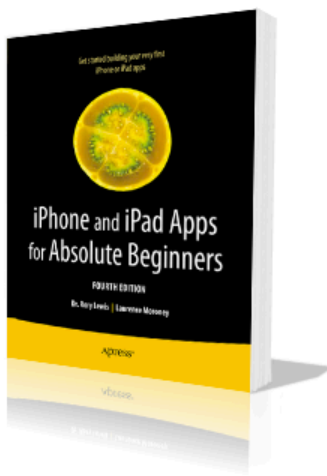
iPhone and iPad Apps for Absolute Beginners

4th Edition

By **Rory Lewis, Laurence Moroney**


This update of an Apress bestseller teaches you how to create your first iOS 7 app to run on iPhone or iPad, using plain English and practical examples. It cuts through the jargon that surrounds iPhone and iPad app development with simple, step-by-step instructions to get you started.

Full Description



ISBN13: 978-1-4302-6361-6
400 Pages
User Level: Beginner
Publishing October 23, 2013, but available now as part of the **Alpha Program**

Your Cart



You don't have any books in your shopping cart yet!

Deal of the Day




Pro **SharePoint 2013**
Branding and Responsive
Web Development

Pro SharePoint 2013 Branding and Responsive Web Development

Start Developing iOS Apps Today

The screenshot shows the Apple Developer Library page for "Start Developing iOS Apps Today". The page is divided into several sections:

- Introduction:**
 - Setup (selected)
 - Tutorial: Basics
- Structuring an App:**
 - App Development Process
 - Designing a User Interface
 - Defining the Interaction
 - Tutorial: Storyboards
- Implementing an App:**
 - Incorporating the Data
 - Using Design Patterns
 - Working with Foundation
 - Writing a Custom Class
 - Tutorial: Add Data
- Next Steps:**
 - iOS Technologies
 - Finding Information
 - Where to Go from Here

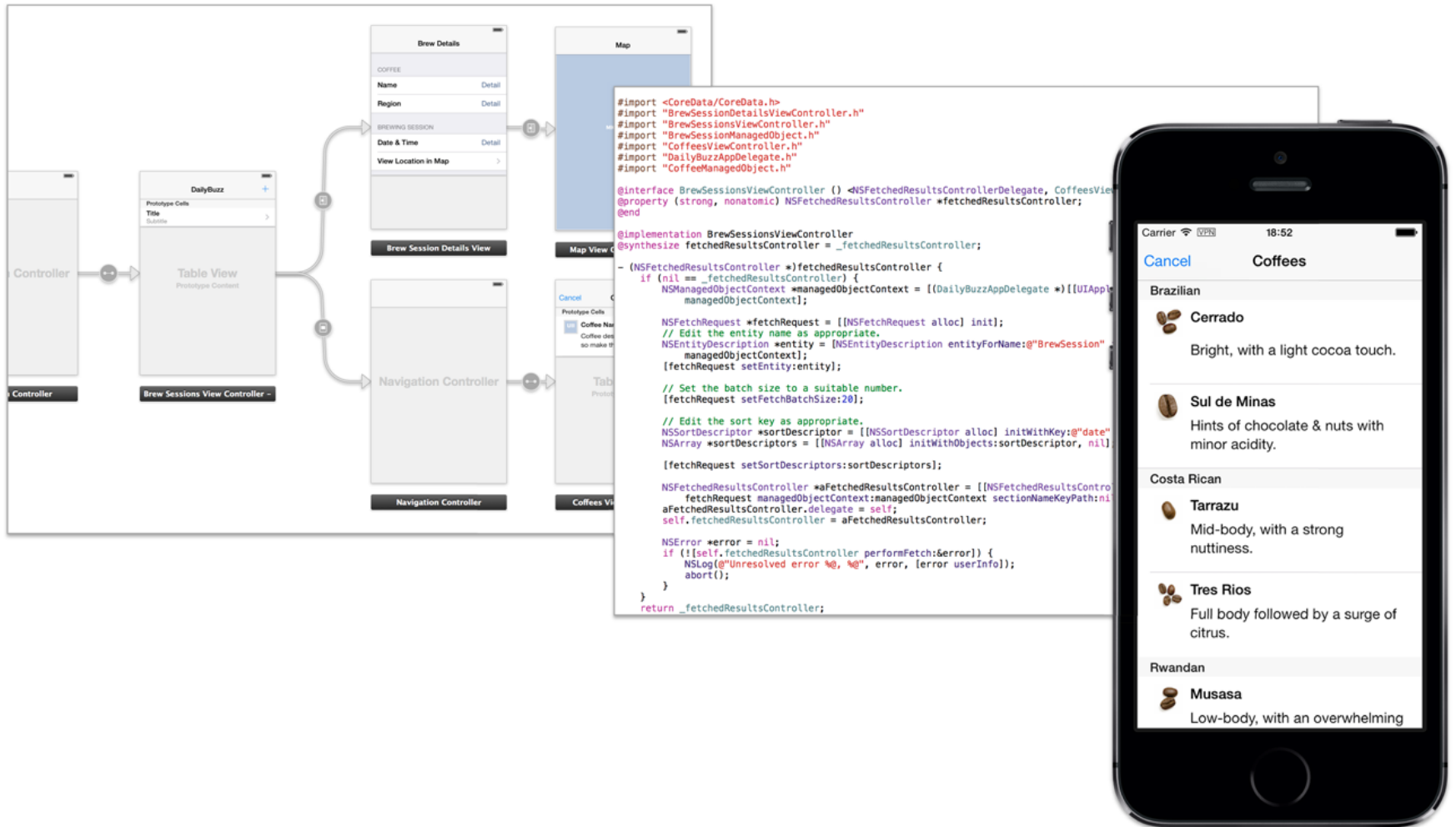
The main content area is titled "Setup" and contains the following text:

Start Developing iOS Apps Today provides the perfect starting point for iOS development. On your Mac, you can create iOS apps that run on iPad, iPhone, and iPod touch. View this guide's four short modules as a gentle introduction to building your first app—including the tools you need and the major concepts and best practices that will ease your path.

The "Setup" section includes a diagram of a storyboard showing various view controllers (Controller, TableView, Navigation Controller, etc.) and a code snippet for a `CoffeeView` class. The code snippet shows a `UITableViewDataSource` implementation that fetches coffee data from a remote source using `NSURLSession` and `JSONSerialization`. A smartphone mockup on the right displays a list of coffee types with their descriptions.

The footer of the page contains the text: "The first three modules each end with a tutorial, where you'll implement what you've learned. At the end of the last tutorial, you'll have created a simple to-do list app." and a "Provide Feedback" button.

Start Developing iOS Apps Today



To develop iOS apps, you need:

- A **Mac computer** running OS X 10.8 or later
 - Mac OS X 10.8 (Mountain Lion)
 - Mac OS X 10.9 (Mavericks)
 - Mac OS X 10.10 (Yosemite)
 - Mac OS X 10.11 (El Capitan)
 - **macOS Sierra (10.12)**
 - **macOS High Sierra (10.13)**
- Xcode
 - Xcode 6
 - Xcode 7
 - **Xcode 8**
 - **Xcode 9**
- iOS SDK



Mac

iPad

iPhone

Watch

TV

Music

Support



MacBook



MacBook Air



MacBook Pro



iMac



Mac Pro



Mac mini



Accessories



macOS Sierra



Compare



New

MacBook

Light. Years ahead.

[Learn more >](#) [Buy >](#)



Store

Mac

iPhone

Watch

iPad

iPod

iTunes

Support



MacBook Air



MacBook Pro



Mac mini



iMac



Mac Pro



OS X Mavericks

Mac

Apps

Pro Apps

Accessories

Server

The notebook people love.

MacBook Air



MacBook and MacBook Air



New

MacBook
from \$1299

- 12-inch (diagonal) LED-backlit Retina display
- 1.2GHz dual-core Intel Core m3, 1.3GHz dual-core Intel Core i5, or 1.4GHz dual-core Intel Core i7 processor
Turbo Boost up to 3.6GHz
- Up to 10 hours battery life¹
- Up to 512GB SSD²
- Force Touch trackpad
- 2.03 pounds³



MacBook Air 13-inch
from \$999

- 13.3-inch (diagonal) LED-backlit widescreen display
- 1.8GHz dual-core Intel Core i5 or 2.2GHz dual-core Intel Core i7 processor
Turbo Boost up to 3.2GHz
- Up to 12 hours battery life¹
- Up to 512GB SSD²
- Multi-Touch trackpad
- 2.96 pounds³



MacBook Pro 15-inch
from \$1999

- 15.4-inch (diagonal) LED-backlit Retina display
- 2.2GHz, 2.5GHz, or 2.8GHz quad-core Intel Core i7 processor
Turbo Boost up to 4.0GHz
- Up to 9 hours battery life¹
- Up to 1TB SSD²
- Force Touch trackpad
- 4.49 pounds³

MacBook Pro



New

MacBook Pro 13-inch
from \$1299

- 13.3-inch (diagonal) LED-backlit Retina display
- 2.3GHz dual-core Intel Core i5 or 2.5GHz dual-core Intel Core i7 processor
Turbo Boost up to 4.0GHz
- Up to 10 hours battery life¹
- Up to 1TB SSD²
- Force Touch trackpad
- 3.02 pounds³



New

MacBook Pro 13-inch
from \$1799

- Touch Bar and Touch ID
- 13.3-inch (diagonal) LED-backlit Retina display
- 3.1GHz or 3.3GHz dual-core Intel Core i5 or 3.5GHz dual-core Intel Core i7 processor
Turbo Boost up to 4.0GHz
- Up to 10 hours battery life¹
- Up to 1TB SSD²
- Force Touch trackpad
- 3.02 pounds³



New

MacBook Pro 15-inch
from \$2399

- Touch Bar and Touch ID
- 15.4-inch (diagonal) LED-backlit Retina display
- 2.8GHz, 2.9GHz, or 3.1GHz quad-core Intel Core i7 processor
Turbo Boost up to 4.1GHz
- Up to 10 hours battery life¹
- Up to 2TB SSD²
- Force Touch trackpad
- 4.02 pounds³

iMac



New

iMac 21.5-inch
from \$1099

- 21.5-inch (diagonal) LED-backlit display
- 2.3GHz dual-core Intel Core i5 processor
Turbo Boost up to 3.6GHz
- 1TB 5400-rpm hard drive; 1TB Fusion Drive; or 256GB SSD²
- Magic Keyboard and



New

iMac 21.5-inch
from \$1299

- 21.5-inch (diagonal) LED-backlit Retina 4K display
- 3.0GHz or 3.4GHz quad-core Intel Core i5 or 3.6GHz quad-core Intel Core i7 processor
Turbo Boost up to 4.2GHz
- 1TB 5400-rpm hard drive; 1TB Fusion Drive; or up to 1TB SSD²



New

iMac 27-inch
from \$1799

- 27-inch (diagonal) LED-backlit Retina 5K display
- 3.4GHz, 3.5GHz, or 3.8GHz quad-core Intel Core i5 or 4.2GHz quad-core Intel Core i7 processor
Turbo Boost up to 4.5GHz
- 1TB, 2TB, or 3TB Fusion Drive; or up to 2TB SSD²

Mac mini



Mac mini
from \$499

- Highly energy efficient⁴
- Up to 3.0GHz dual-core Intel Core i7 processor
Turbo Boost up to 3.5GHz
- Up to 1TB hard drive; 256GB, 512GB, or 1TB SSD; or 1TB or 2TB Fusion Drive²



Mac Pro
from \$2999

- 3.5GHz 6-core, 3.0GHz 8-core, or 2.7GHz 12-core Intel Xeon E5 processor
Up to 30MB of L3 cache
- Up to 1TB SSD²

macOS High Sierra

macOS

[Overview](#) [How to Upgrade](#)

[Upgrade now](#)

macOS High Sierra

Your Mac. Elevated.



Mac

iPad

iPhone

Watch

TV

Music

Support



OS X

Design

Continuity

Better Apps

What is OS X

How to Upgrade

Upgrade



OS X Yosemite

Every bit as powerful as it looks.

An elegant design that feels entirely fresh, yet inherently familiar. The apps you use every day, enhanced with new features. And a completely new relationship between your Mac and iOS devices. OS X Yosemite changes how you see your Mac.

And what you can do with it. Upgrade for free at the Mac App Store.

OS X El Capitan

A refined experience and improved performance for your Mac.

FREE UPGRADE

[Learn More >](#)

Best New Apps & Games



OS X El Capitan
Utilities

★★★★★ 165 Ratings

GET ▾

macOS Sierra

macOS Sierra

What can your Mac do now? Just ask.

FREE UPGRADE

[Learn More >](#)



macOS Sierra
Utilities

DOWNLOAD ▾

macOS Sierra

macOS Sierra

By Apple

Open the Mac App Store to buy and download apps.



[View in Mac App Store](#)

Free

Category: [Utilities](#)

Updated: Sep 20, 2016

Version: 10.12

Size: 4.77 GB

Languages: English, Arabic, Catalan, Croatian, Czech, Danish, Dutch, Finnish, French, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Japanese, Korean, Malay, Norwegian, Polish, Portuguese, Romanian, Russian, Simplified Chinese, Slovak, Spanish, Swedish, Thai, Traditional Chinese, Turkish, Ukrainian, Vietnamese

Seller: Apple Inc.

© 2016 Apple, Inc.

Rated 4+

Description

Siri makes its debut on Mac, with new features designed just for the desktop. Your Mac works with iCloud and your Apple devices in smart new ways, and intelligent capabilities make your photos, music, and messaging even more enjoyable.

[Apple Web Site](#) ▶ [macOS Sierra Support](#) ▶

[...More](#)

Screenshots



Search for information and images just by asking Siri.
Then drag results right into an email or document.



macOS Sierra

To set up the installation of macOS Sierra, click Continue.



Continue

macOS High Sierra



macOS High Sierra 4+

New technologies at the heart of the system make your Mac more reliable, capable and responsive — and lay the foundation for future innovations. macOS High Sierra also refines the features and apps you use every day. It's macOS at its highest level yet.

Easily organise, edit and view your photos in Photos.

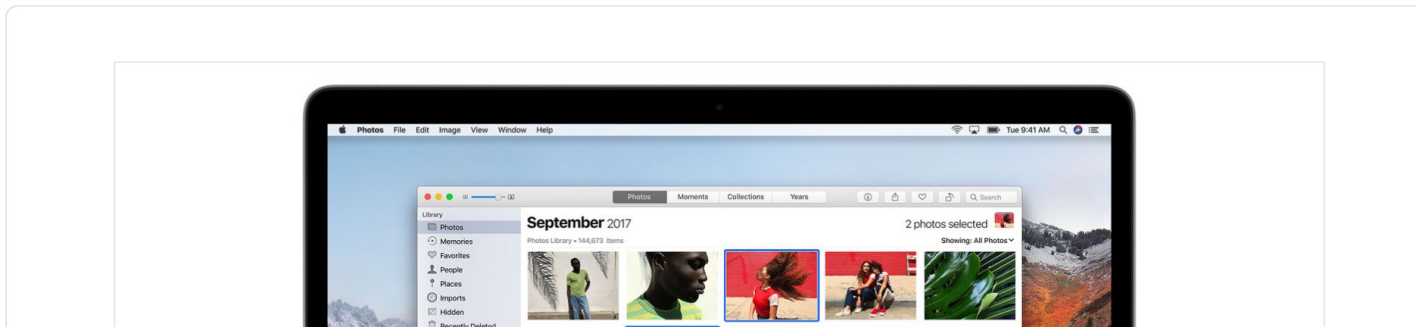
- Make short videos from your Live Photos using new Loop and Bounce effects.
- Easily locate and organise your content with the new sidebar.
- Conveniently access all of your editing tools in the redesigned Edit View.
- Fine-tune colour and contrast in your photos with new Curves and Selective Color tools.
- Access third-party apps directly from Photos and save the edited images back to your Photos library. ...

- [Apple Web Site](#) >
- [macOS High Sierra Support](#) >
- [App License Agreement](#) >
- [Privacy Policy](#) >

...More

Information

Category: **Utilities**
Updated: 12 October 2017
Version: 10.13
Price: Free
Size: 4.80 GB
Family Sharing: Yes
Languages: English, Arabic, Catalan, Croatian, Czech, Danish, Dutch, Finnish, French, German, Greek, Hebrew, Hindi, Hungarian, Indonesian, Italian, Japanese, Korean, Malay, Norwegian, Polish, Portuguese, Romanian, Russian, Simplified Chinese, Slovak, Spanish, Swedish, Thai, Traditional Chinese, Turkish, Ukrainian, Vietnamese



Xcode 6



Xcode

The complete toolset for building great apps.



Xcode 7



Xcode

The complete toolset for building great apps.



Swift 2

Xcode 8



Xcode

The complete toolset for building great apps.



Swift 3

Xcode 9



Xcode

The complete toolset for building great apps.



Swift 4

iOS 9 for Developers



Developing for iOS 9

iOS 9 SDK includes new APIs and services that enable new categories of apps and features. Multitasking and gaming APIs help enhance app functionality and create immersive games. Expanded search capabilities, and new support for CloudKit, HomeKit, HealthKit, and MapKit extend iOS to more places than ever before.

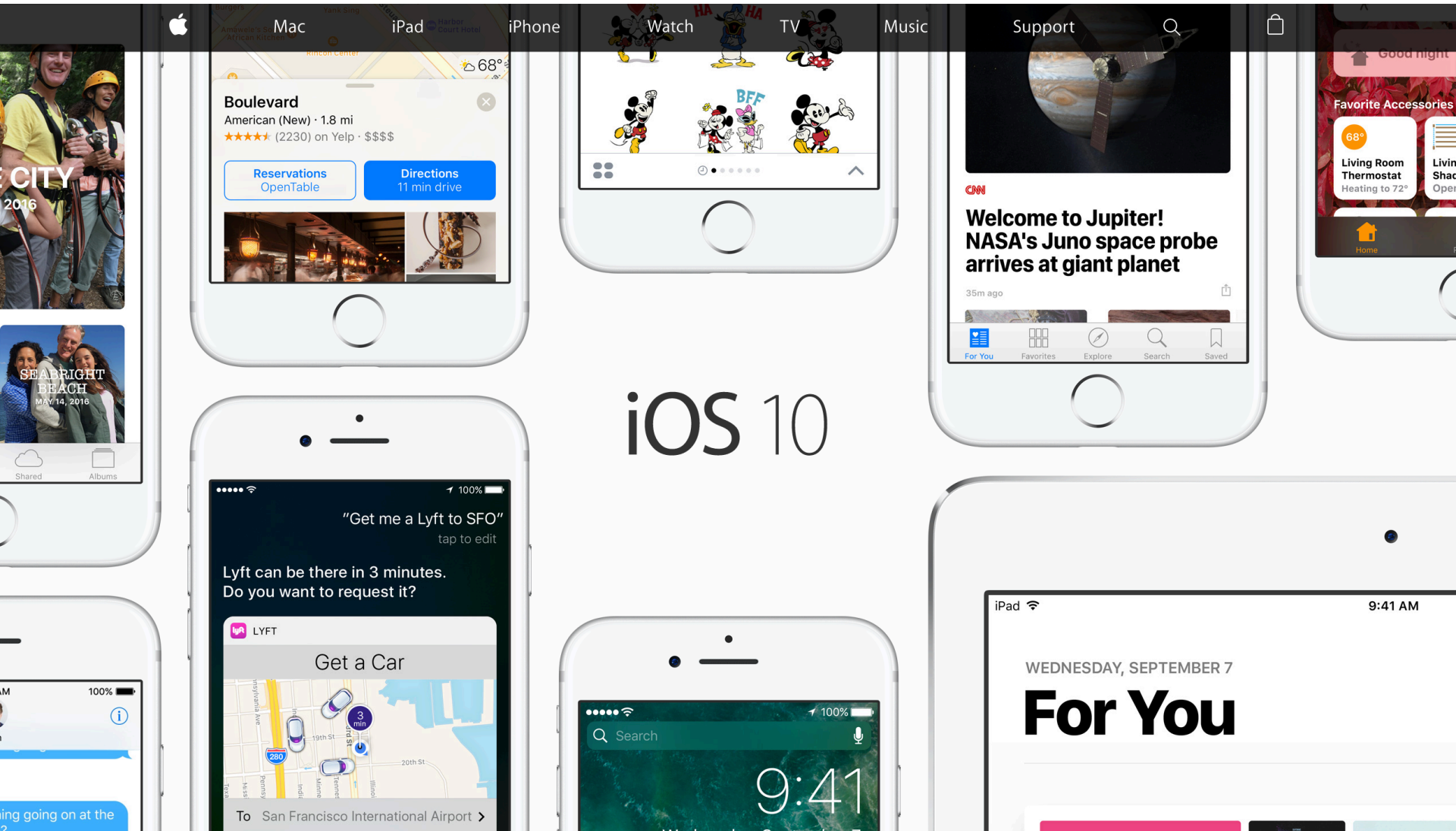
iOS 10 for Developers



Developing for iOS 10

The iOS 10 SDK includes new APIs and services that enable new categories of apps and features. Your apps can now extend to Messages, Siri, Phone, and Maps to provide more engaging functionality like never before.

iOS 10



iOS 10

iOS 10

The world's most advanced mobile operating system.

iOS 10 is compatible with these devices.

iPhone



- iPhone 7
- iPhone 7 Plus
- iPhone 6s
- iPhone 6s Plus
- iPhone 6
- iPhone 6 Plus
- iPhone SE
- iPhone 5s
- iPhone 5c
- iPhone 5

iPad



- iPad Pro 12.9-inch
- iPad Pro 9.7-inch
- iPad Air 2
- iPad Air
- iPad 4th generation
- iPad mini 4
- iPad mini 3
- iPad mini 2

iPod



- iPod touch 6th generation



Xcode 6

The screenshot displays the Xcode 6 development environment. The top menu bar includes Xcode, File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, and Help. The status bar shows the project name 'Shutterbugs', target 'iPhone 5s', and build status 'Succeeded'.

The interface is divided into several panes:

- Left Pane (Project Navigator):** Shows the project structure for 'Shutterbugs', including Views, Controllers, and Resources.
- Storyboard:** Displays a table view prototype cell with a green flower image and text '3 comments, 2 likes' and 'Comment'. A 'Table View Prototype Content' label is visible at the bottom.
- Right Pane (Code Editor):** Shows the Swift code for 'GamesTableViewController.swift'. The code includes imports for UIKit and CoreShutterbugs, and defines the 'GamesTableViewController' class with methods for game list retrieval and table view data source.
- Bottom Pane (Inspector):** Shows the 'Add New Constraints' dialog, which is currently displaying constraints for a table view cell, including Width (320), Height (316), and Spacing to nearest neighbor (63).

The dock at the bottom contains various system icons, including the Spotlight search icon, Launchpad, Safari, Photos, Mail, Calendar, Notes, Xcode, and the Trash.

Get the Tools

Mac App Store



Xcode **FREE** ▼



Download



Xcode 7



Xcode 7

This release includes the Xcode IDE, Swift 2 compiler, Instruments, Simulator, and latest SDKs for OS X, iOS, and watchOS.

Build

7A220

Posted Date

Sep 16, 2015

SDK

iOS 9

OS X v10.11

watchOS 2

Xcode 8



Xcode 8 requires a Mac running macOS Sierra 10.12 or OS X El Capitan 10.11.5 or later. Xcode 8 includes SDKs for iOS 10.0, watchOS 3.0, macOS Sierra 10.12, and tvOS 10.0.

Build
8A218a
Posted Date
Sep 13, 2016
SDKs
iOS 10
macOS 10.12
watchOS 3
tvOS 10

Xcode 8



Get the latest beta releases of Xcode, iOS, macOS, watchOS, tvOS, and more.



Xcode 8.1 beta 2

[Download Xcode 8.1](#)

[Release Notes](#)

Build
8T46g

Posted Date
Oct 04, 2016

SDKs
iOS 10.1
macOS 10.12.1
watchOS 3.1
tvOS 10.0.1



Xcode 8

[Download Xcode 8](#)

[Release Notes](#)

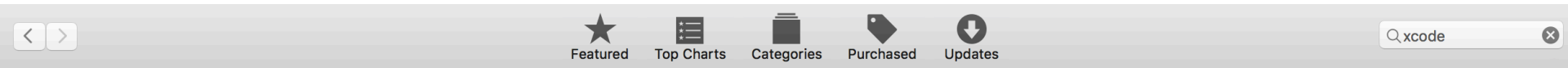
Build
8A218a

Posted Date
Sep 13, 2016

SDKs
iOS 10
macOS 10.12
watchOS 3
tvOS 10

Source: <https://developer.apple.com/xcode/download/>

Xcode 8



Xcode



Install ▾

Xcode **4+**

Essentials

Xcode includes everything developers need to create great applications for Mac, iPhone, iPad, Apple TV, and Apple Watch. Xcode provides developers a unified workflow for user interface design, coding, testing, and debugging. The Xcode IDE combined with the Swift programming language make developing apps easier and more fun than ever before.

...

What's New in Version 8.0

Xcode 8 includes Swift 3, and SDKs for iOS 10, watchOS 3, tvOS 10, and macOS Sierra.

...

...More

...More

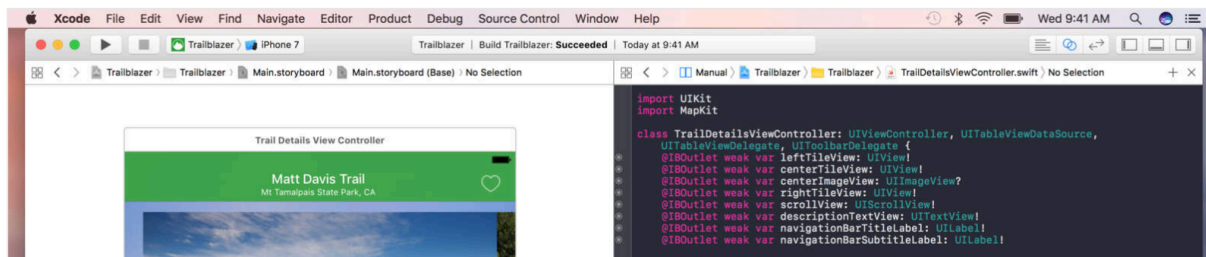
- [Apple Web Site](#) >
- [Xcode Support](#) >
- [App License Agreement](#) >
- [Privacy Policy](#) >

Information

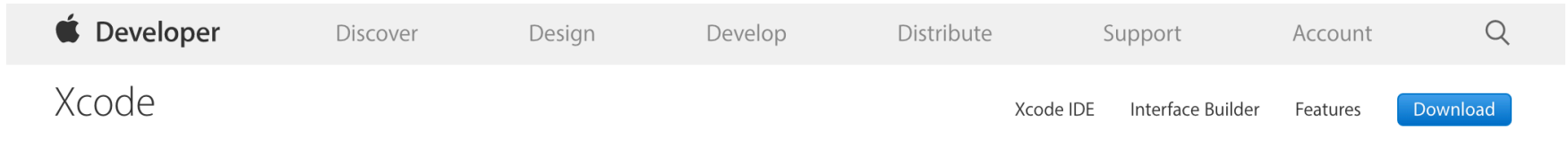
Category: Developer Tools
Updated: 13 September 2016
Version: 8.0
Price: Free
Size: 4.43 GB
Family Sharing: Yes
Language: English
Developer: iTunes S.a.r.l.
© 1999–2016 Apple Inc.

Rated 4+

Compatibility:
OS X 10.11.5 or later

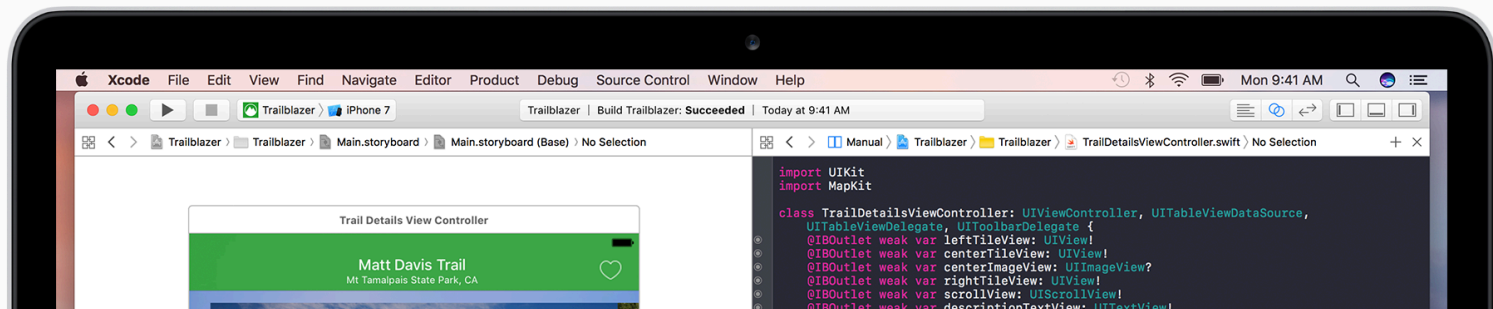


Xcode 8



Xcode 8

Xcode 8 includes everything you need to create amazing apps for iPhone, iPad, Mac, Apple Watch, and Apple TV. This radically faster version of the IDE features new editor extensions that you can use to completely customize your coding experience. New runtime issues alert you to hidden bugs by pointing out memory leaks, and a new Memory Debugger dives deep into your object graph. Swift 3 includes more natural and consistent API naming, which you can experiment with in the new Swift Playgrounds app for iPad.

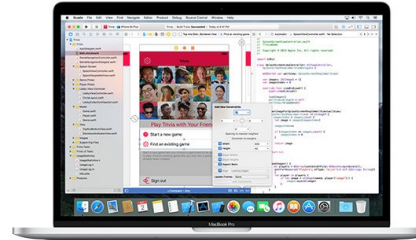


Source: <https://developer.apple.com/xcode/>

Xcode 9



Xcode



Open

Xcode **4+**

Essentials

Xcode includes everything developers need to create great applications for Mac, iPhone, iPad, Apple TV, and Apple Watch. Xcode provides developers a unified workflow for user interface design, coding, testing, and debugging. The Xcode IDE combined with the Swift programming language make developing apps easier and more fun than ever before.

...

What's New in Version 9.0

Xcode 9 includes Swift 4 and SDKs for iOS 11, watchOS 4, tvOS 11, and macOS High Sierra 10.13

...

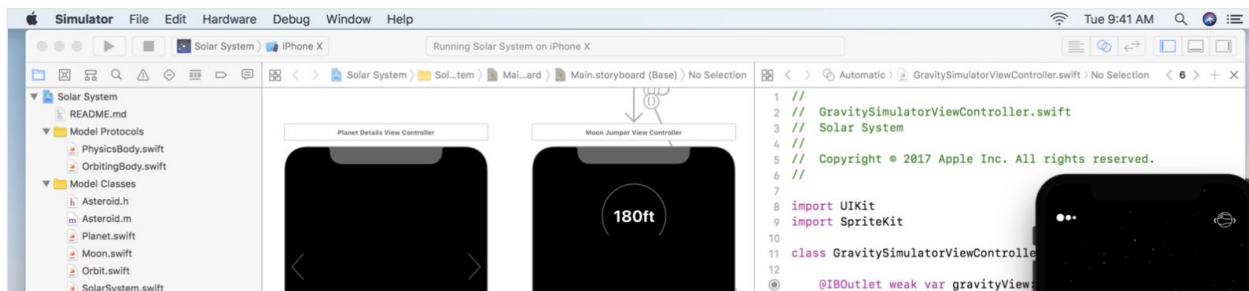
...More

...More

- [Apple Web Site](#) >
- [Xcode Support](#) >
- [App License Agreement](#) >
- [Privacy Policy](#) >

Information

Category: Developer Tools
Updated: 19 September 2017
Version: 9.0
Price: Free
Size: 5.39 GB
Family Sharing: Yes
Language: English
Developer: Apple Distribution International
© 1999–2017 Apple Inc.
Rated 4+
Compatibility:
macOS 10.12.6 or later



Xcode 8

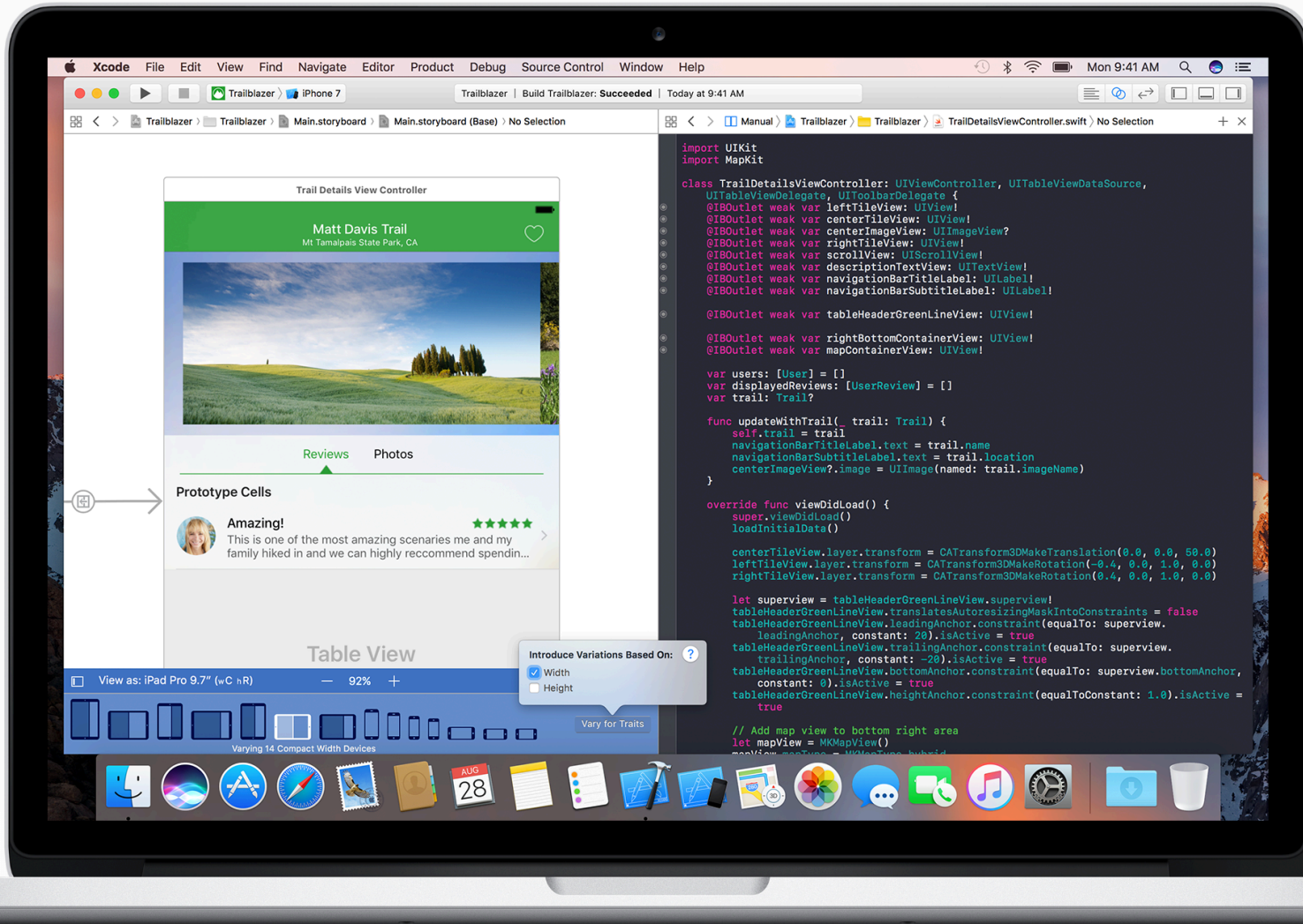
Xcode

Xcode IDE

Interface Builder

Features

Download



Swift 3



Swift 3 is the first major release of the innovative programming language built completely in the open with the community of developers at Swift.org.

This release unifies core API naming rules under a new public API Naming Guidelines document that makes writing Swift code feel even more natural.

You can also experiment with Swift 3 in the new Swift Playgrounds app for iPad.

Swift 3



ABOUT SWIFT

BLOG

DOWNLOAD

Releases

Previews

Snapshots

Using Downloads

GETTING STARTED

DOCUMENTATION

MIGRATING TO SWIFT 3

SOURCE CODE

COMMUNITY

CONTRIBUTING

CONTINUOUS
INTEGRATION

PROJECTS

Download Swift

Releases

Swift 3.0


Download	Date
Xcode 8* (Toolchain) (Debugging Symbols)	September 13, 2016
Ubuntu 15.10 (Signature)	September 13, 2016
Ubuntu 14.04 (Signature)	September 13, 2016

*Swift 3.0 is available as part of [Xcode 8](#).

Source: <https://swift.org/download/#releases>

Swift 4

→ ↻ Apple Inc. [US] | <https://swift.org/download/#releases>



Swift

- ABOUT SWIFT
- BLOG
- DOWNLOAD**
 - Releases**
 - [Snapshots](#)
 - [Using Downloads](#)
- GETTING STARTED
- DOCUMENTATION
- MIGRATING TO SWIFT 4
- SOURCE CODE
- COMMUNITY
- CONTRIBUTING
- CONTINUOUS INTEGRATION
- SOURCE COMPATIBILITY

FOCUS AREAS

Download Swift

Releases

Swift 4.0

Download	Date
Xcode 9.0* (Toolchain) (Debugging Symbols)	September 19, 2017
Ubuntu 16.10 (Signature)	September 19, 2017
Ubuntu 16.04 (Signature)	September 19, 2017
Ubuntu 14.04 (Signature)	September 19, 2017



Xcode

What's New

Xcode IDE

Interface Builder

Features

Download



Xcode





Toolbar

Editor area

The screenshot shows the Xcode IDE interface with the following components:

- Navigator area (left):** A sidebar showing a project tree for 'Adventure'. It includes folders like 'Adventure Shared', 'Scene', 'Sprites', 'AI', 'Utilities', 'Adventure - OS X', 'Adventure iOS', 'Assets', 'Sounds', 'UI', and 'Particles'. The 'Scene' folder is expanded, showing 'AdventureScene.swift', 'LayeredCharacterScene.swift', and 'Player.swift'.
- Editor area (center):** A code editor displaying Swift code for 'AdventureScene.swift'. The code includes methods like 'addSpawnPoints()', 'addBackgroundTiles()', and 'addCollisionWalls()'. A breakpoint is set at line 1.1 of the 'addSpawnPoints()' method. The code is as follows:


```

addSpawnPoints()
addTrees()
addCollisionWalls()
}

func addBackgroundTiles() {
    for tileNode in sBackgroundTiles {
        addNode(tileNode, atWorldLayer: .Ground)
    }
}

func addSpawnPoints() {
    for y in 0..

```
- Debug area (bottom center):** A console window showing the current state of the application. It displays:


```

self = (Adventure_IOS.AdventureScene) 0x7bf477b0 (lldb)
self: SyGenerator (RangeGenerator<Int>)
self: y (Int)
self: $x$generator (RangeGenerator<Int>)
self: x (Int)
self: location (CGPoint)
self: spot (Adventure_IOS.SpriteLocation)

```
- Utilities area (right):** A panel containing several sections:
 - Identity and Type:** Shows target membership for 'Adventure' and 'Adventure iOS'.
 - Text Settings:** Includes options for Text Encoding (Unicode (UTF-8)), Line Endings (Default - OS X / Unix (LF)), Indent Using (Spaces), and Wrap lines (checked).
 - Source Control:** Shows repository, type, and current branch information.
 - View Controller:** A controller that supports the fundamental view-management model in iOS.
 - Navigation Controller:** A controller that manages navigation through a hierarchy of views.
 - Table View Controller:** A controller that manages a table view.

Navigator area

Debug area

Utilities area



Xcode

Inspector pane

Identity and Type Show

Interface Builder Document

Opens in ▼

Builds for ▼

View as ▼

Use Auto Layout

Use Size Classes

Inspector bar

Library pane

View Controller - A controller that supports the fundamental view-management model in iOS.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Table View Controller - A controller that manages a table view.

Library bar

Filter bar

Getting Started with Xcode 9 Development

1. Get a Mac
2. Register as an Apple Developer
3. Install Xcode
4. Enroll in the Apple Developer Program
(Optional)



Swift.

A modern programming language
that is safe, fast, and interactive.



Swift

- Swift is a powerful and intuitive **programming language** for **iOS**, **OS X**, and **watchOS**.
- Writing Swift code is interactive and fun, the syntax is concise yet expressive, and apps run lightning-fast.
- Swift is ready for your next project — or addition into your current app — because Swift code works side-by-side with Objective-C.

Swift Language



A new programming language for iOS and OS X.

- Swift is a new **object-oriented programming language** for **iOS and OS X development**.
- Swift is **modern, powerful, expressive,** and **easy to use**.

Swift Language



- **Unified**
A **complete replacement** for both the **C** and **Objective-C** languages.
 - Swift provides full object-oriented features, and includes low-level language primitives such as types, flow control, and operators.
- **Fast**
- **Complete platform**
- **Safe by design**
- **Modern**
- **Interactive**





```
print("Hello world")
```

```
println("Hello world")
```

```
println("Hello World")
```

```
println("Hello World")
```

'println' has been renamed to 'print'

Fix-it Replace "println" with "print"



Swift 2

```
print("Hello world")
```



Swift

```
let count = 10
var price = 23.55

let firstMessage = "Swift is awesome. "
let secondMessage = "What do you think?"
var message = firstMessage + secondMessage

print(message)
```

Objective-C

```
const int count = 10;
double price = 23.55;

NSString *firstMessage = @"Swift is awesome. ";
NSString *secondMessage = @"What do you think?";
NSString *message = [NSString stringWithFormat:@"%s%s",
firstMessage, secondMessage];

NSLog(@"%@", message);
```

```
var s = "Hello" + " World"
```



```
var myVariable = 82  
myVariable = 90  
let myConstant = 82
```

```
let individualScores = [75, 43, 93, 87, 12]
var teamScore = 0
for score in individualScores {
    if score > 60 {
        teamScore += 3
    } else {
        teamScore += 1
    }
}
print(teamScore)
```

```
let individualScores = [75, 43, 93, 87, 12]
var teamScore = 0
for score in individualScores {
    if score > 60 {
        teamScore += 3
    } else {
        teamScore += 1
    }
}
print(teamScore)
```

```
[75, 43, 93, 87, 12]
0
```

```
(3 times)
```

```
(2 times)
```

```
"11\n"
```

The Swift Programming Language (Swift 4)

```
print( "Hello, world!" )
```

Xcode 9



Welcome to Xcode

Version 9.0 (9A235)



Get started with a playground

Explore new ideas quickly and easily.



Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.

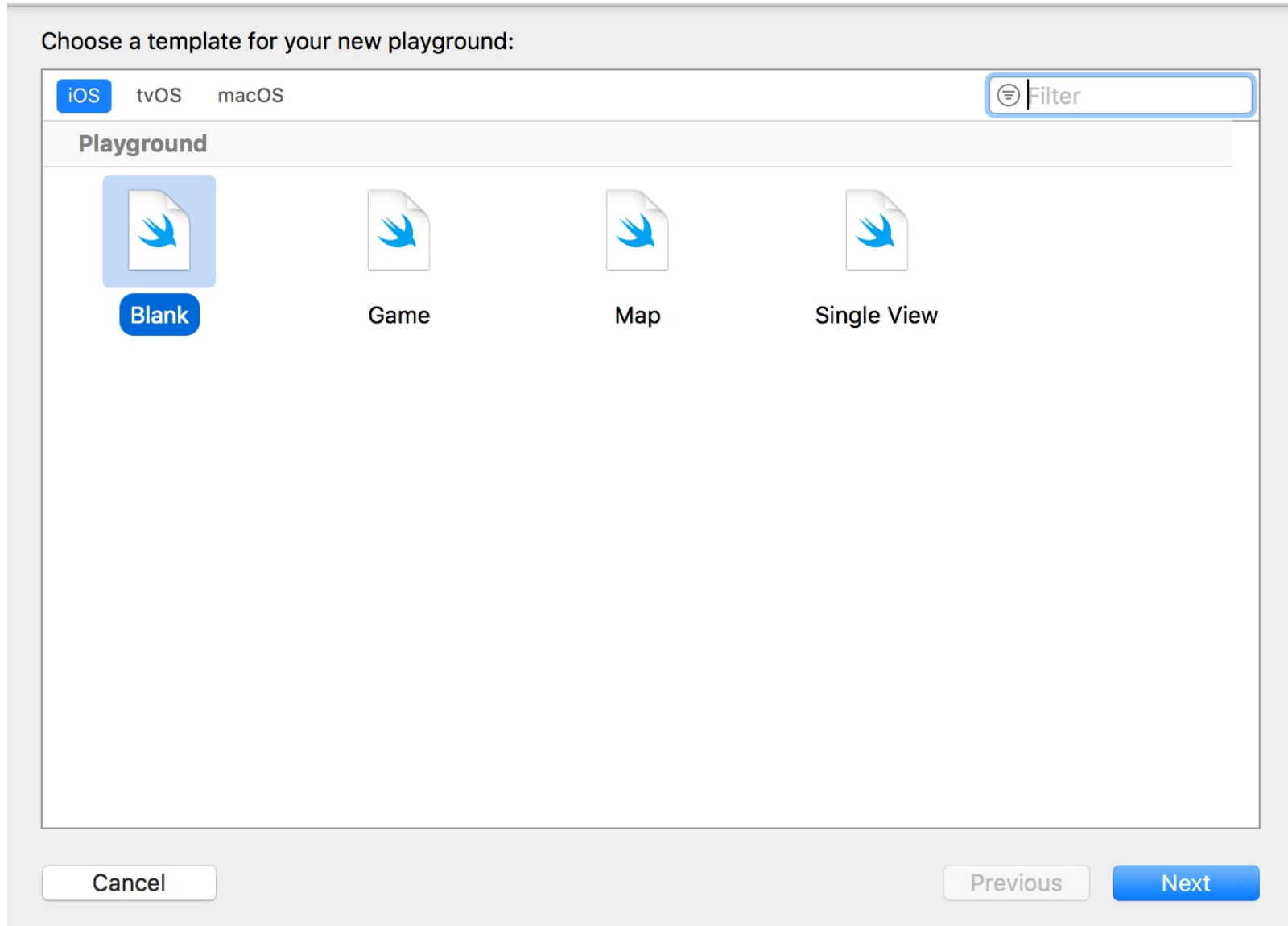


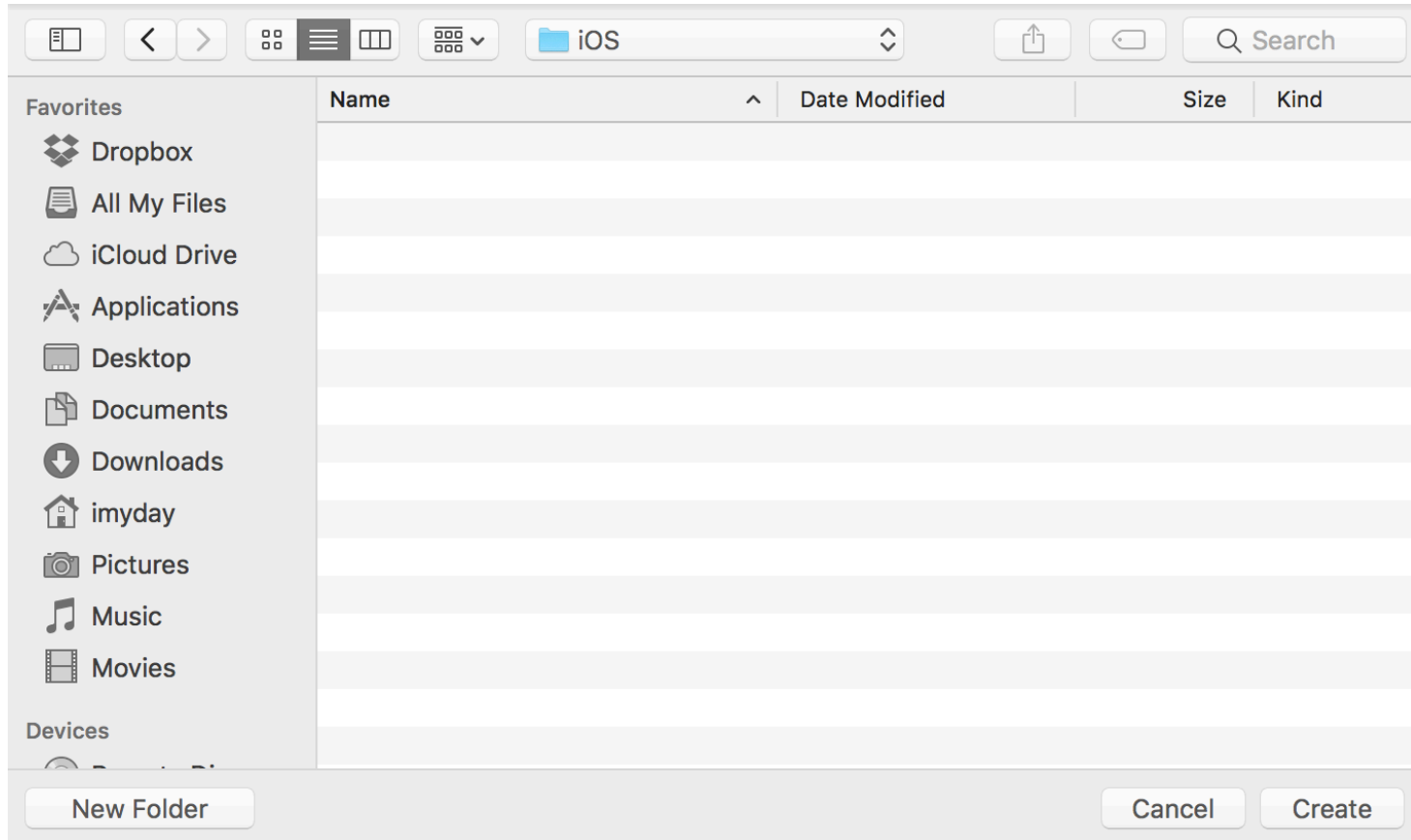
Clone an existing project

Start working on something from an SCM repository.

Show this window when Xcode launches

Xcode 9 Playground





Swift 4 in Xcode 9 Playground

A screenshot of the Xcode 9 Playground interface. The window title is "Ready | Today at 1:53 PM". The playground name is "MyPlayground1". The code editor shows the following Swift 4 code:

```
1  //: Playground - noun: a place where people can play
2
3  import UIKit
4
5  var str = "Hello, playground"
6
```

The output area on the right shows the string "Hello, playground". A callout box highlights the code lines 1 through 6.

Swift 4 in Xcode 9 Playground

A screenshot of the Xcode 9 Playground interface. The window title is "MyPlayground". The status bar shows "Ready" and "Today at 8:04 AM". The code editor contains the following Swift code:

```
//: Playground - noun: a place where people can play  
  
import UIKit  
  
var str = "Hello, playground"  
print(str)
```

The right-hand pane shows the output of the code:

```
"Hello, playground"  
"Hello, playground\n"
```

At the bottom, there is a play button icon and the output text "Hello, playground".

Swift 4 in Xcode 9 Playground



```
var message = "Hello World"
print(message)
var a = 7
var b = 2
var total = a + b
if (total < 10)
{
    print("Hello Swift 3")
}
```

Swift 4 in Xcode 9 Playground



```
import UIKit

var str = "Hello, playground"
print(str)

var message = "Hello World"
print(message)
var a = 7
var b = 2
var total = a + b
if (total < 10)
{
    print("Hello Swift 3")
}
```

```
"Hello, playground"
"Hello, playground\n"

"Hello World"
"Hello World\n"
7
2
9

"Hello Swift 3\n"
```

Hello, playground
Hello World
Hello Swift 3

Swift 4 in Xcode 9 Playground



```
var strS = "Your score is "  
var score = 90  
var yourScore = strS + String(score)  
print(yourScore)  
if (score >= 60) {  
    print("Pass")  
}  
else {  
    print("Fail")  
}
```

Swift 4 in Xcode 9 Playground



MyPlayground

```
//: Playground - noun: a place where people can play
```

```
import UIKit
```

```
var str = "Hello, playground"
```

```
var strS = "Your score is "
```

```
var score = 90
```

```
var yourScore = strS + String(score)
```

```
print(yourScore)
```

```
if (score >= 60) {
```

```
    print("Pass")
```

```
}
```

```
else {
```

```
    print("Fail")
```

```
}
```

```
"Hello, playground"
```

```
"Your score is "
```

```
90
```

```
"Your score is 90"
```

```
"Your score is 90\n"
```

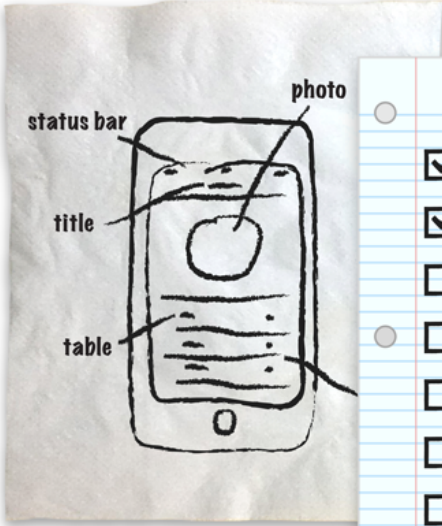
```
"Pass\n"
```



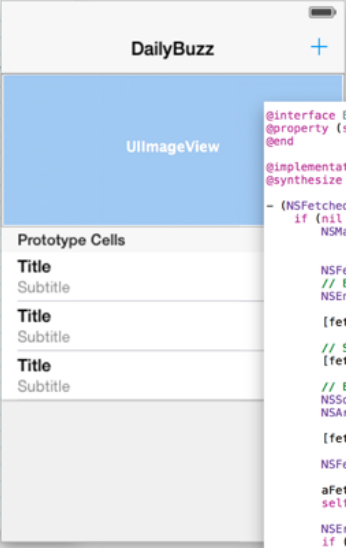
```
Your score is 90
```

```
Pass
```

iOS App Development Process



- Focus: Primary Target
- Think top down
- Consistent UI
- Gestures
- Orientation?
- Check target size
- Reduce settings



```
@interface BrewSessionsViewController () <NSFetchResultsControllerDelegate>
@property (strong, nonatomic) NSFetchResultsController *fetchResultsController;
@end

@implementation BrewSessionsViewController
@synthesize fetchedResultsController = _fetchedResultsController;

- (NSFetchResultsController *)fetchedResultsController {
    if (nil == _fetchedResultsController) {
        NSManagedObjectContext *managedObjectContext =
            [self managedObjectContext];

        NSFetchRequest *fetchRequest = [[NSFetchRequest alloc]
            initWithEntityName:@"BrewSession"];
        // Edit the entity name as appropriate.
        NSEntityDescription *entity = [NSEntityDescription
            descriptionWithClassName:@"BrewSession"
            managedObjectContext:managedObjectContext];
        [fetchRequest setEntity:entity];

        // Set the batch size to a suitable number.
        [fetchRequest setFetchBatchSize:20];

        // Edit the sort key as appropriate.
        NSSortDescriptor *sortDescriptor = [[NSSortDescriptor
            alloc] initWithKey:@"date" ascending:YES];
        NSArray *sortDescriptors = [[NSArray alloc] initWithObjects:
            sortDescriptor, nil];

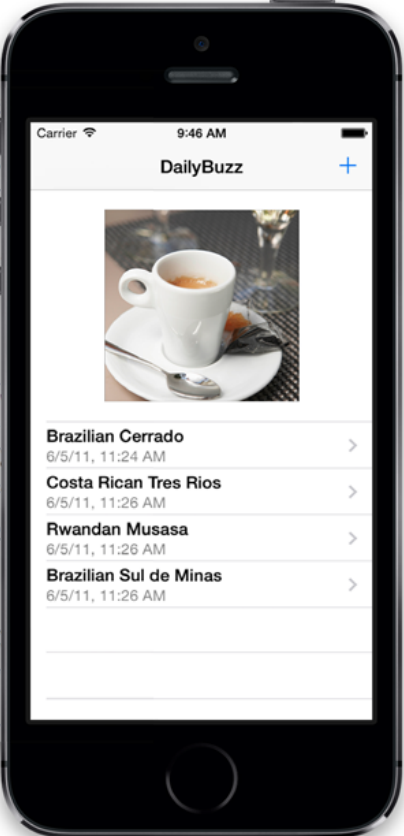
        [fetchRequest setSortDescriptors:sortDescriptors];

        NSFetchResultsController *aFetchedResultsController =
            [[NSFetchResultsController alloc] initWithFetchRequest:fetchRequest
            managedObjectContext:managedObjectContext
            sectionNameKeyPath:nil
            delegate:self
            cacheName:nil];
        self.fetchedResultsController = aFetchedResultsController;

        NSError *error = nil;
        if (![self.fetchedResultsController performFetch]
            || [error != nil]) {
            NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
            abort();
        }
    }
    return _fetchedResultsController;
}

- (void)addBrewSessionForCoffee:(NSManagedObject *)coffee
    NSManagedObjectContext *context = [self managedObjectContext];
    NSManagedObjectContext *session = [NSEntityDescription insertNewObjectForContext:context];
    [session setValue:coffee forKey:@"coffee"];
    [session setValue:date forKey:@"date"];
    [[DailyBuzzAppDelegate sharedInstance] applicationDidEnterBackground];
}

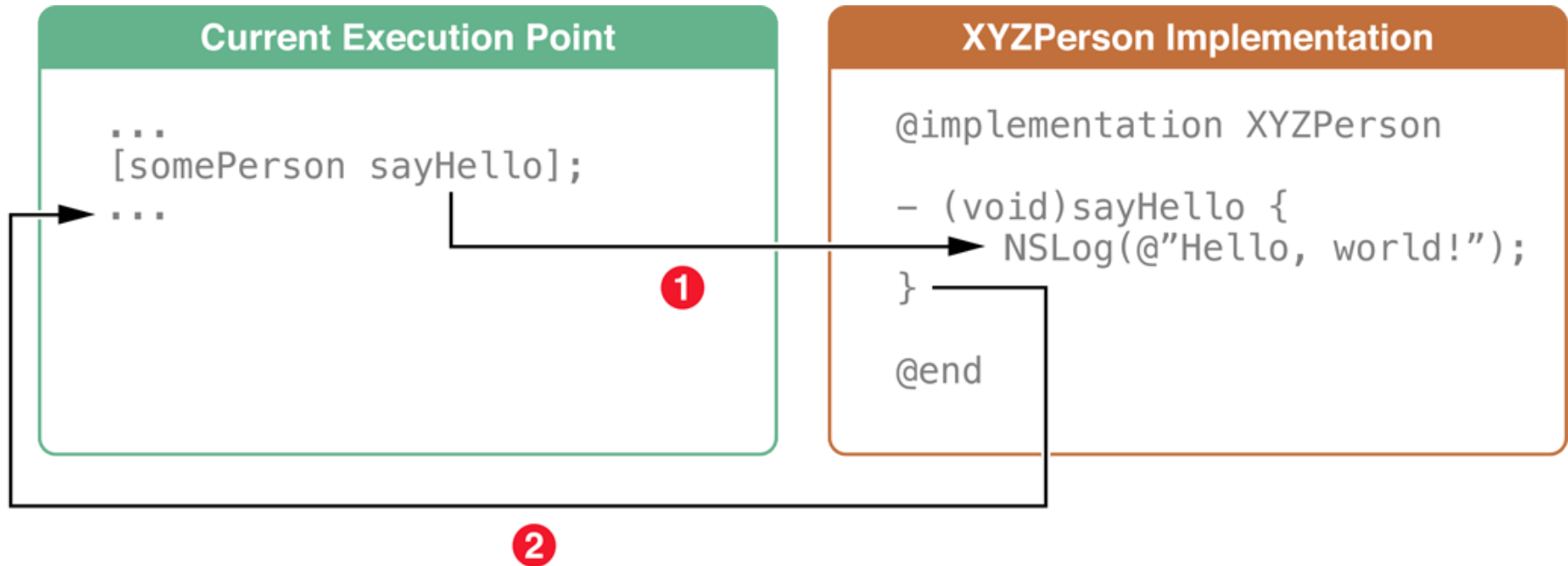
@end
```



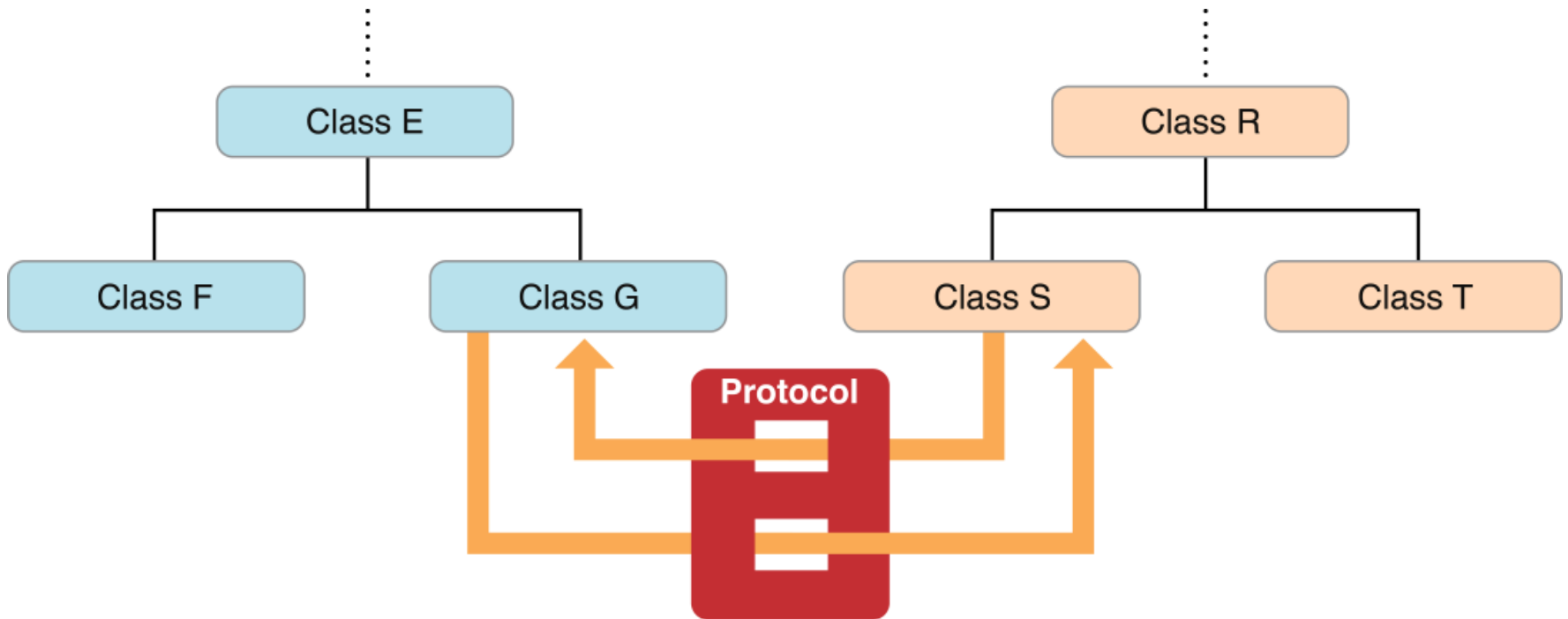
iOS App Development Process

- Defining the Concept
- Designing a User Interface
- Defining the Interaction
- Implementing the Behavior

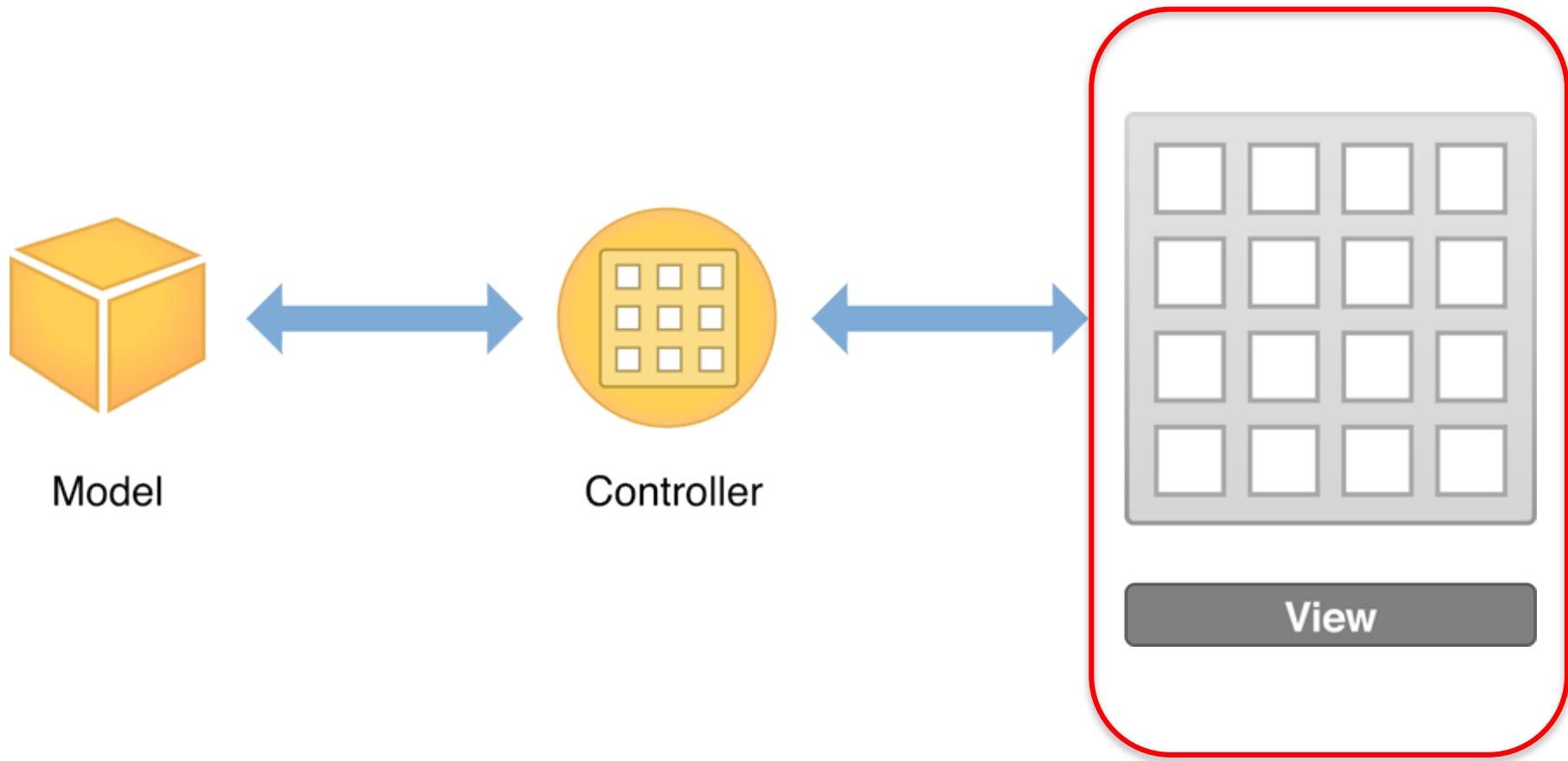
Objects Communicate Through Messages



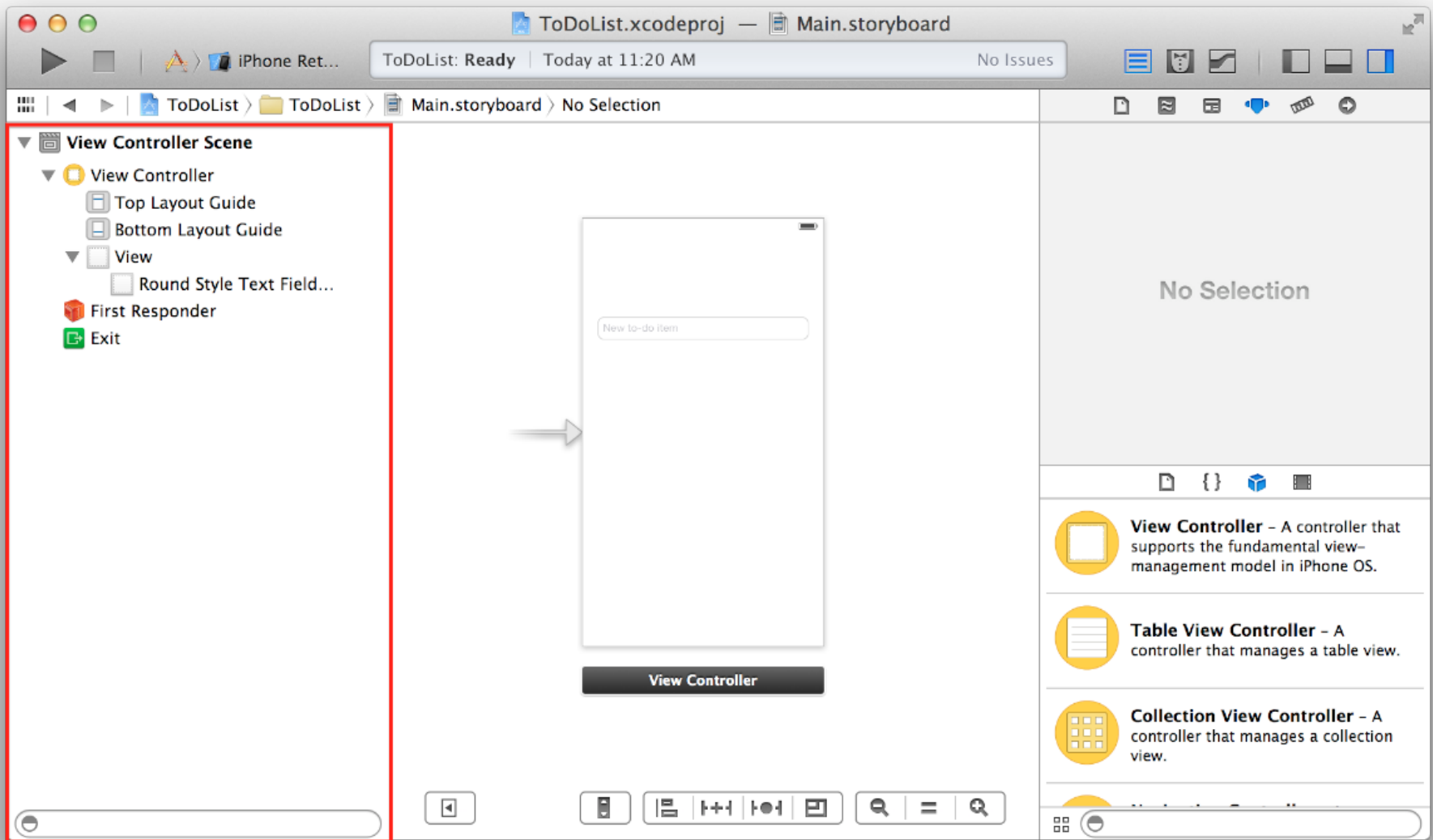
Protocols Define Messaging Contracts



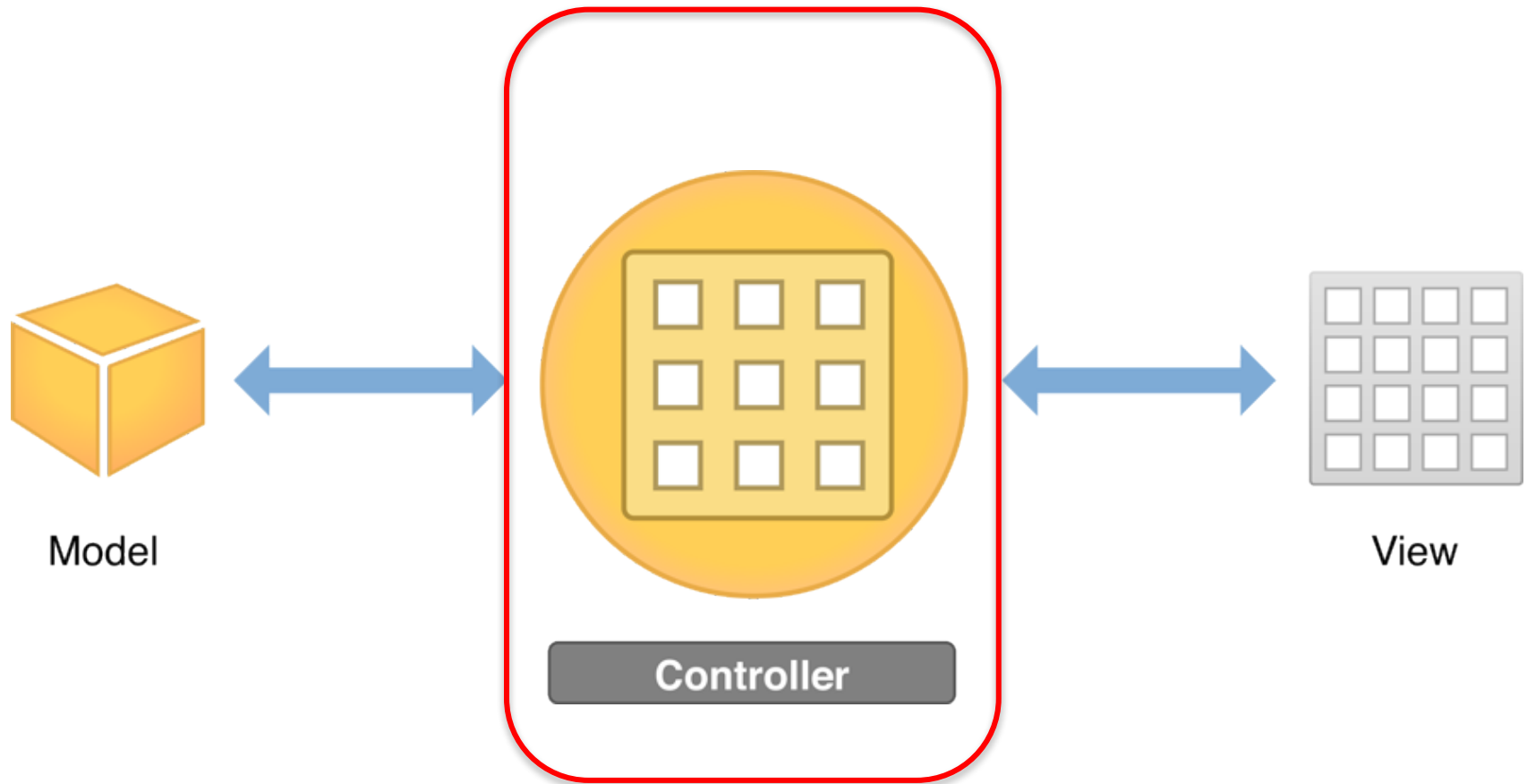
Designing a User Interface



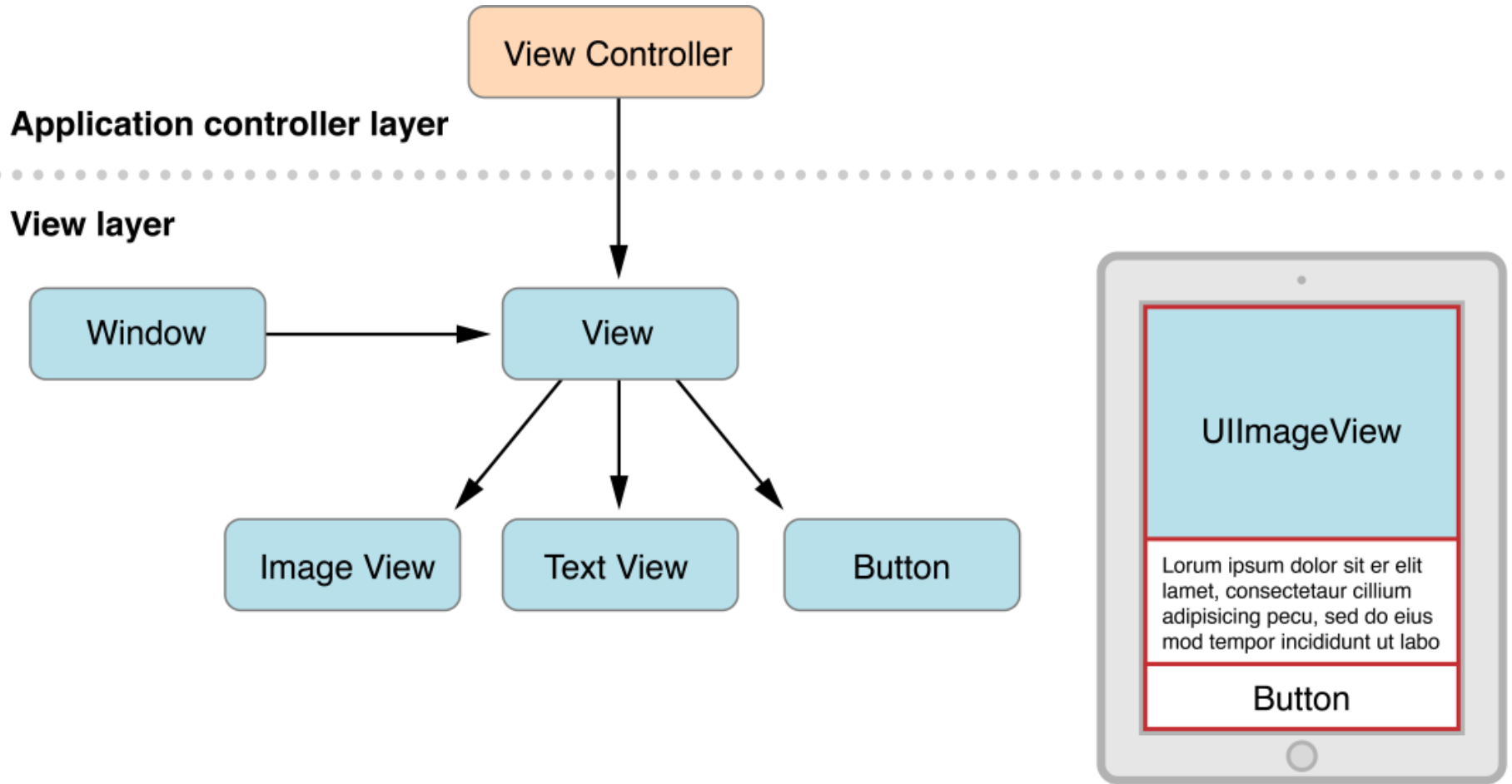
Use Storyboards to Lay Out Views



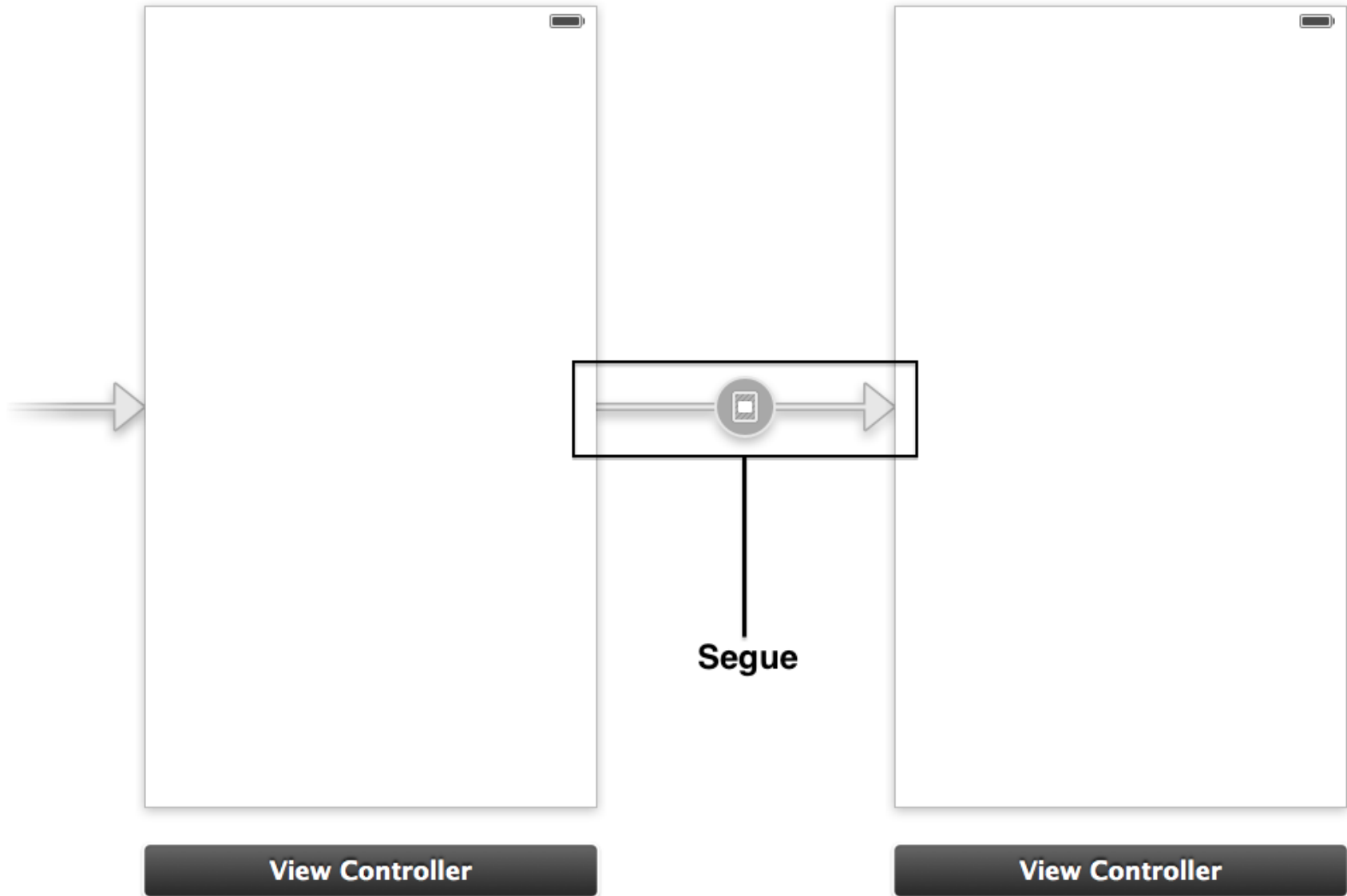
Defining the Interaction



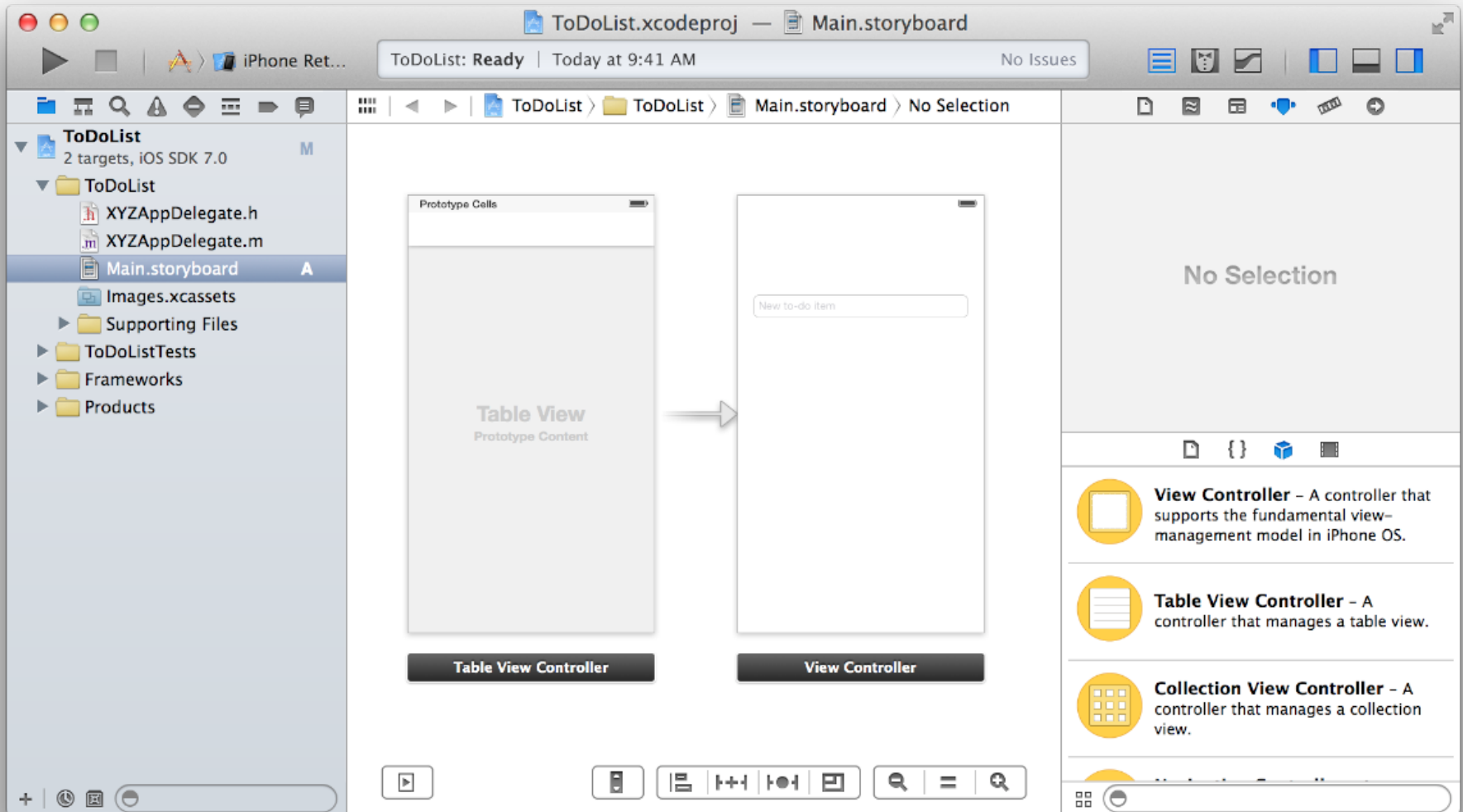
View Controllers



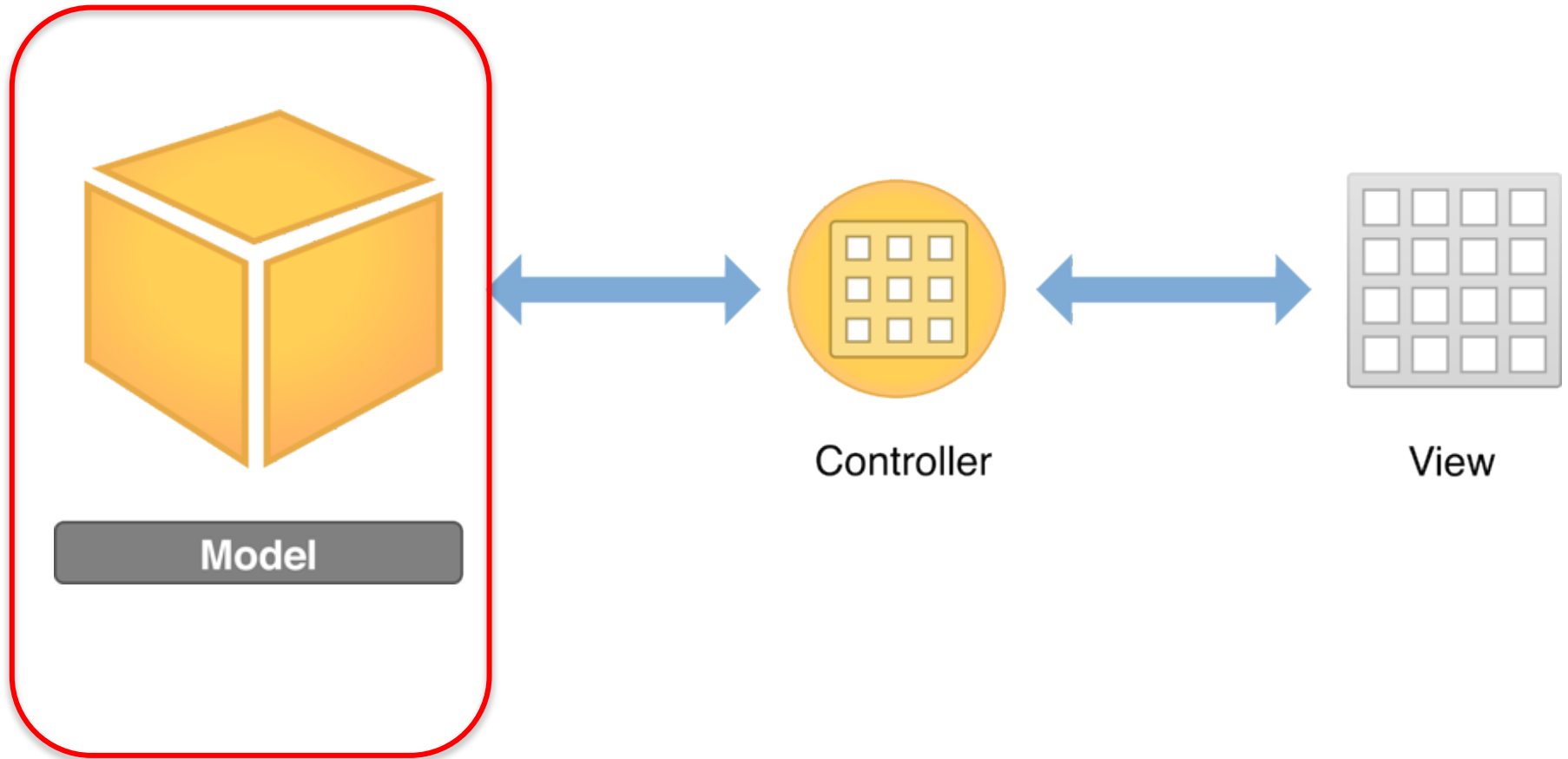
Use Storyboards to Define Navigation



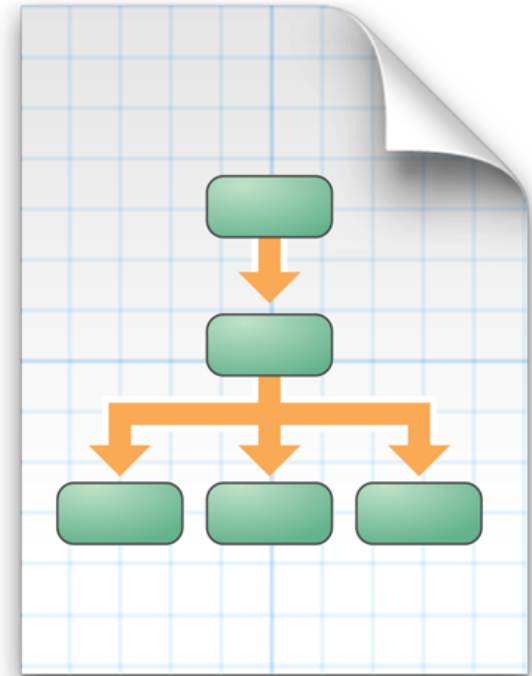
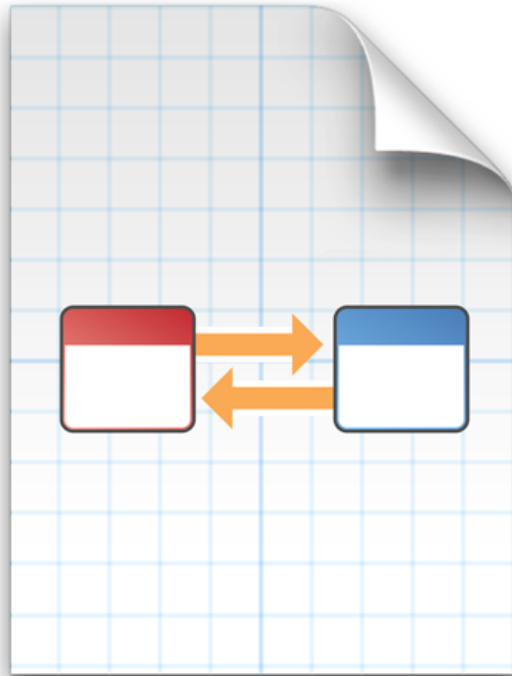
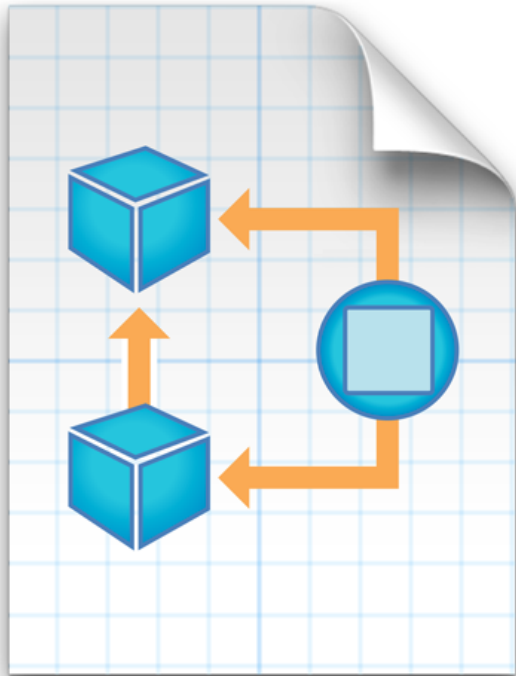
Storyboards



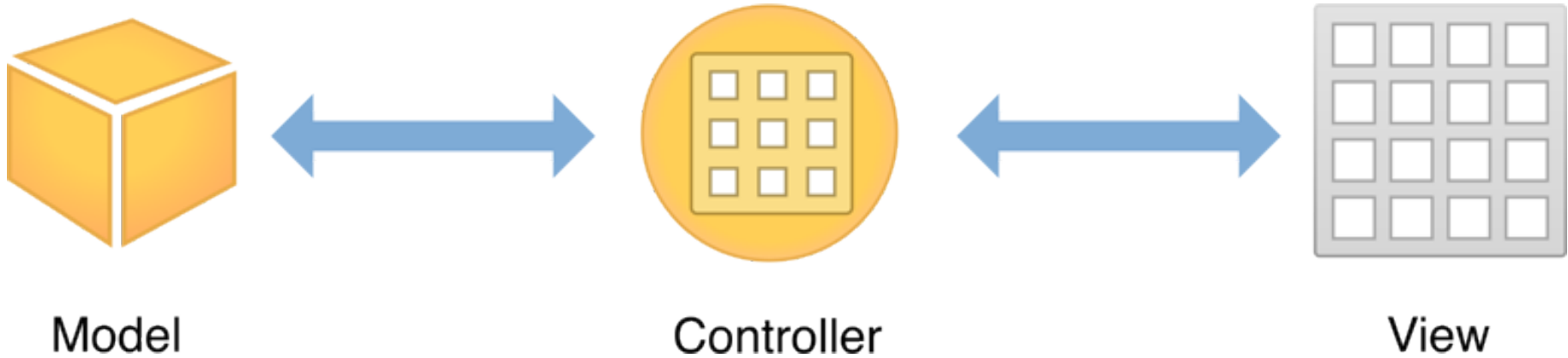
Incorporating the Data



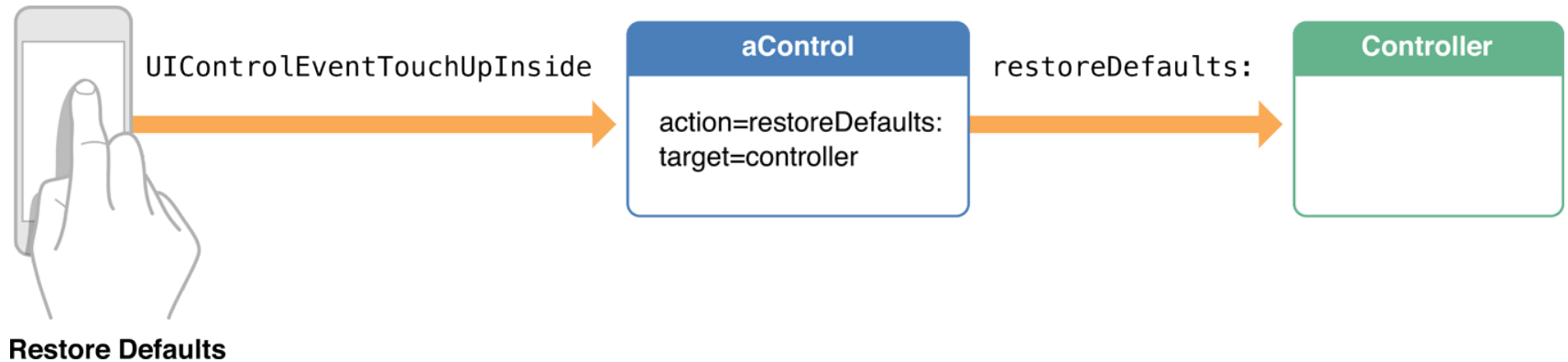
Using Design Patterns



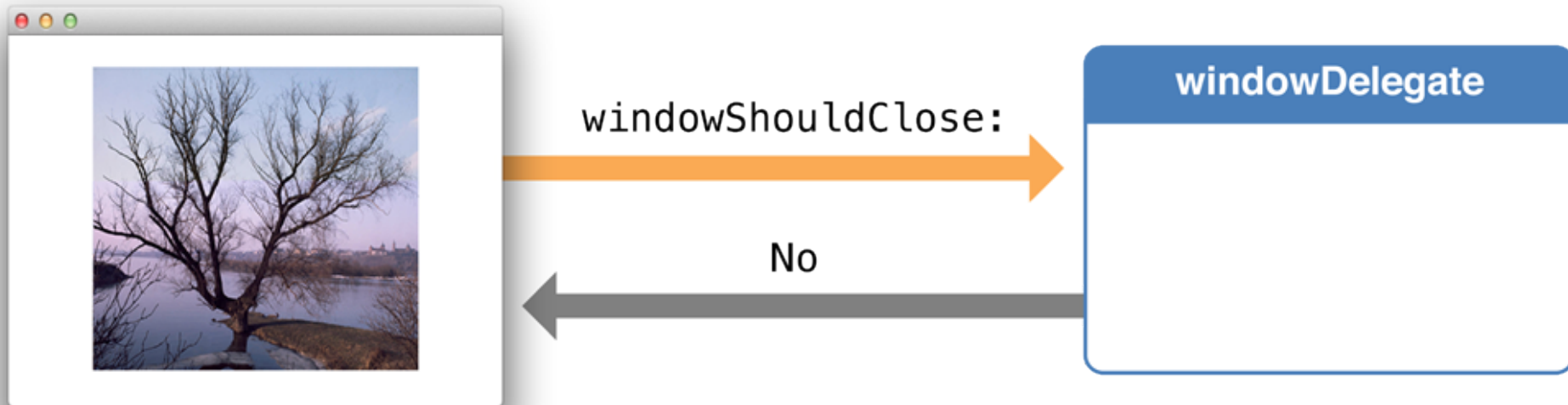
Model-View-Controller (MVC)



Target-Action



Delegation



IBOutlet and IBAction

- IBOutlet
 - Interface Builder Outlet
- IBAction
 - Interface Builder Action

Demo:
**Building Your First iOS App with
Xcode 9 (Swift 4)**

Building Your First iOS App with Xcode 9

The image shows the Xcode 9 interface. On the left, the iPhone 7 simulator is running the app. The status bar at the top of the simulator shows 'Carrier', signal strength, Wi-Fi, and battery. The time is 10:36 AM. The text input field contains 'Myday'. Below the input field, the text 'Hello, Myday' is displayed in orange, and 'Hello' is displayed in blue. A keyboard is visible at the bottom of the simulator, with 'Myday' in the search bar and the letters 'q w e r t y u i o p' on the top row.

In the center, the storyboard is visible, showing a white background with a text input field containing 'Hello World' and a blue 'Hello' label below it.

On the right, the Swift code editor is open to 'ViewController.swift'. The code is as follows:

```
//
//  ViewController.swift
//  HelloWorld
//
//  Created by iMyday on 10/5/16.
//  Copyright © 2016 imtku. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

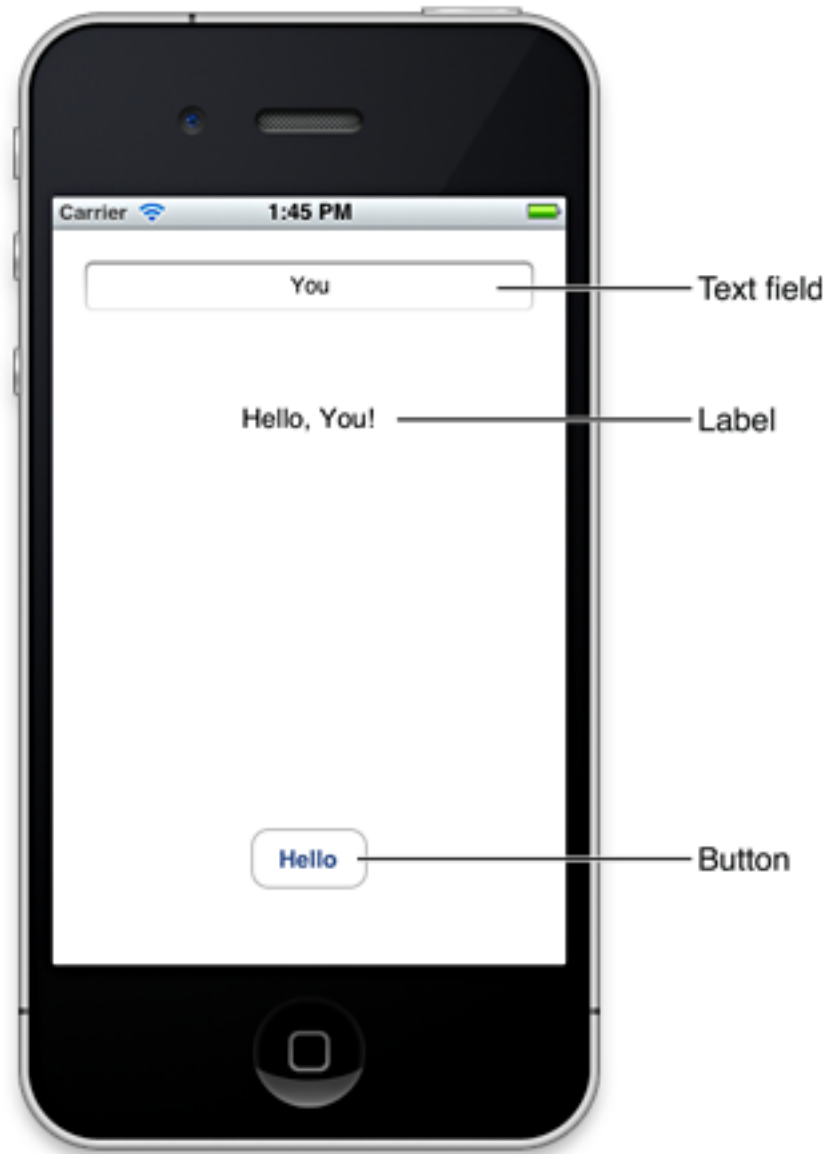
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var txtYourName: UITextField!

    @IBOutlet weak var myLabel: UILabel!

    @IBAction func btnHello(_ sender: AnyObject) {
        let strYourName:String! = txtYourName.text
        myLabel.text = "Hello, " + strYourName
        txtYourName.text = ""
    }
}
```

Your First iOS App



Xcode 8 with Swift 3



Xcode 8



Swift 3

Xcode 9 with Swift 4



Xcode 9



Swift 4

Launchpad → Xcode



Safari



Mail



Contacts



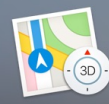
Calendar



Reminders



Notes



Maps



Messages



FaceTime



Photo Booth



Photos



iMovie



GarageBand



iTunes



iBooks



App Store



Pages



Numbers



Keynote



Preview



Dictionary



Calculator



Other



Mission Control



Dashboard



System Preferences



Google Chrome



Dropbox



Microsoft OneNote



Microsoft PowerPoint



Microsoft Excel



Microsoft Outlook



Microsoft Word



Siri



Xcode



Xcode 9



Welcome to Xcode

Version 9.0 (9A235)



Get started with a playground

Explore new ideas quickly and easily.



Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.



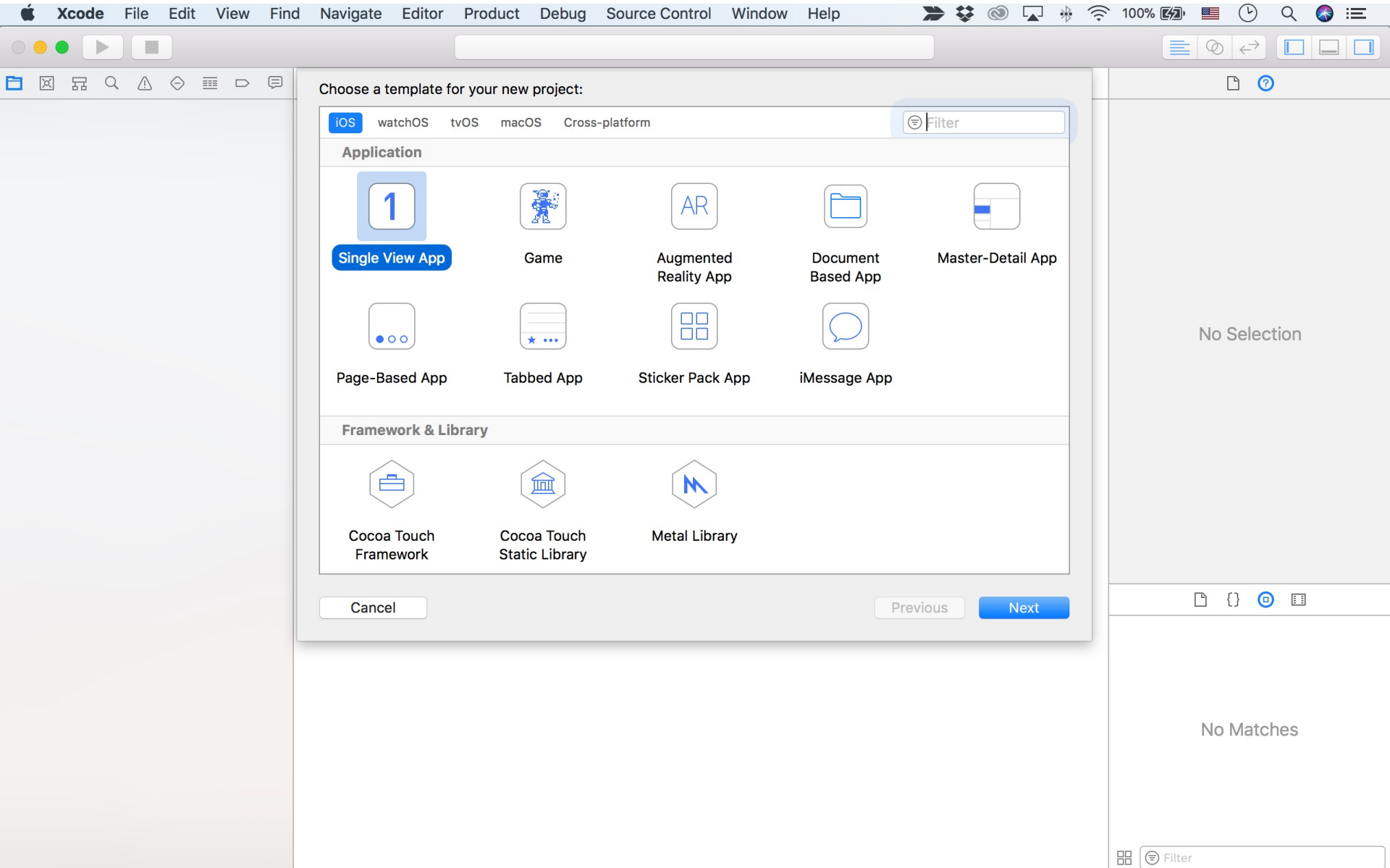
Clone an existing project

Start working on something from an SCM repository.

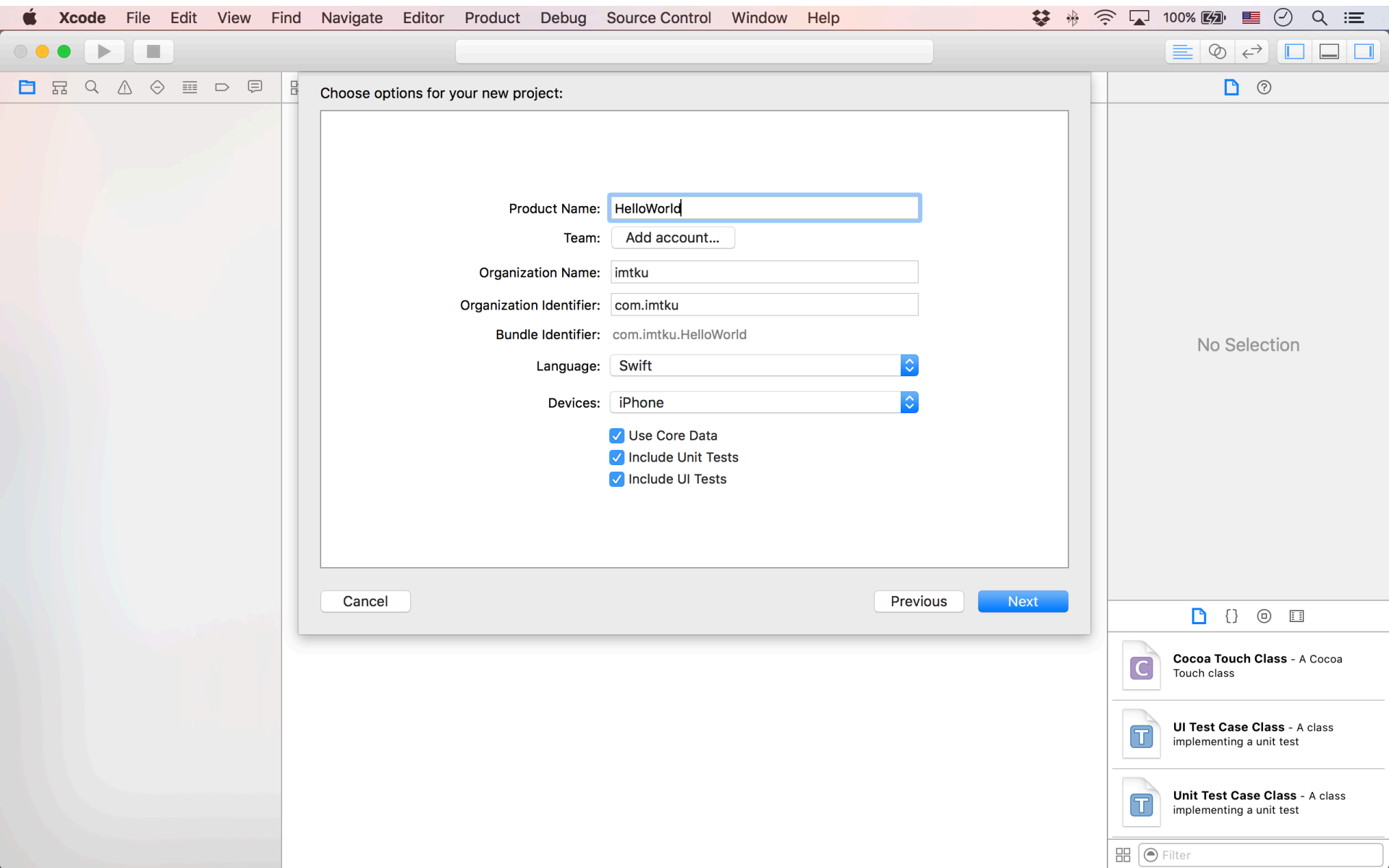


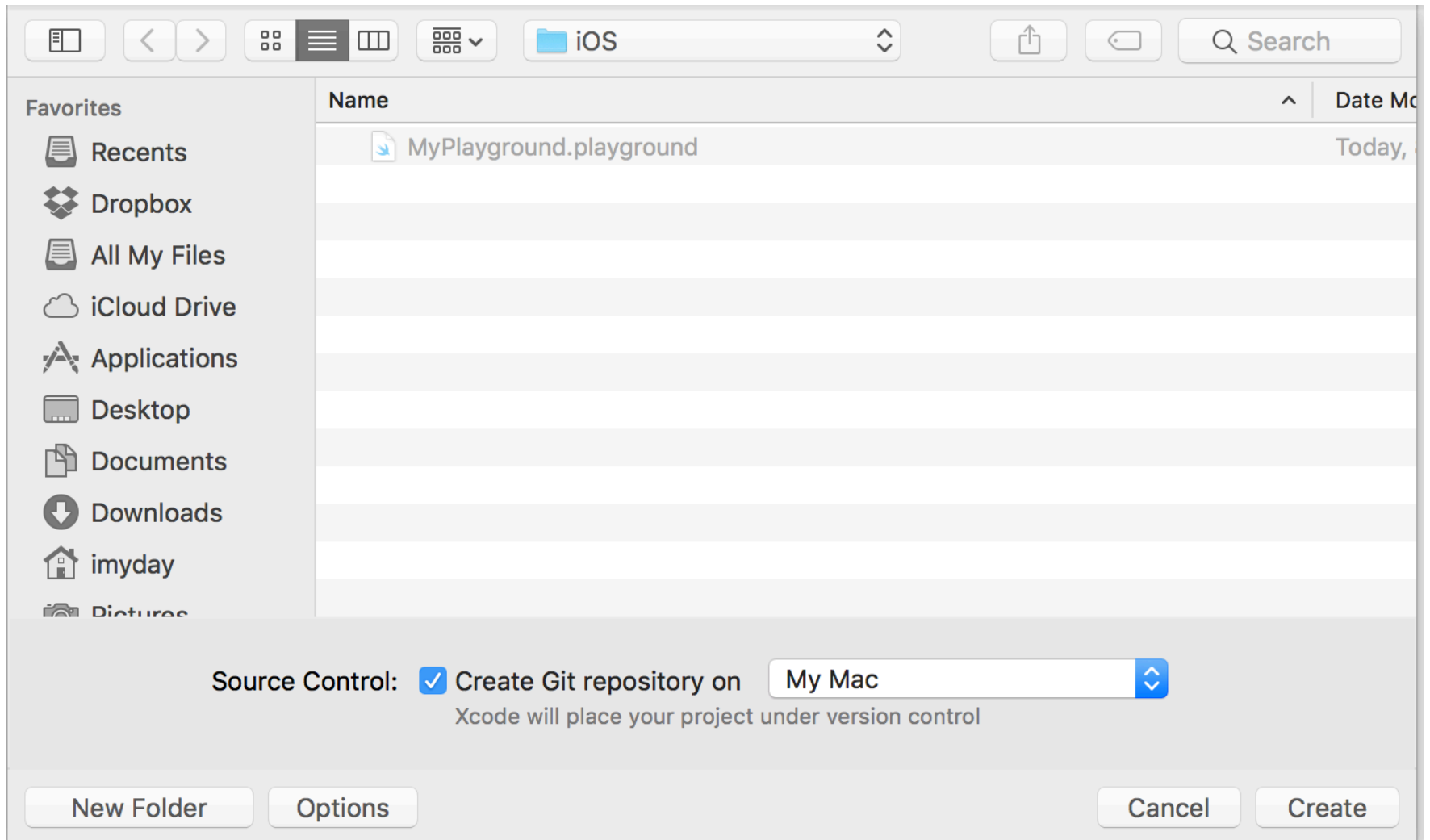
Show this window when Xcode launches

Xcode Single View App



Swift Language





Project Navigator showing the file structure for HelloWorld:

- AppDelegate.swift
- ViewController.swift
- Main.storyboard
- Assets.xcassets
- LaunchScreen.storyboard
- Info.plist
- HelloWorld.xcdatamodeld
- HelloWorldTests
- HelloWorldUITests
- Products

Target Configuration for HelloWorld (iPhone7,1):

- Identity**
 - Display Name: HelloWorld
 - Bundle Identifier: com.imtku.HelloWorld
 - Version: 1.0
 - Build: 1
- Signing**
 - No accounts found. Add a developer account to sign your app.
 - Add Account...
- Deployment Info**
 - Deployment Target: 10.0
 - Devices: iPhone
 - Main Interface: Main
 - Device Orientation: Portrait, Upside Down, Landscape Left, Landscape Right
 - Status Bar Style: Default
 - Hide status bar
 - Requires full screen
- App Icons and Launch Images**
 - App Icons Source: Applcon

Inspector showing Identity and Type, Project Document, and Text Settings:

- Identity and Type**
 - Name: HelloWorld
 - Location: Absolute
 - Full Path: /Users/imyday/Documents/SMAP/iOS/HelloWorld/HelloWorld.xcodeproj
- Project Document**
 - Project Format: Xcode 3.2-compatible
 - Organization: imtku
 - Class Prefix:
- Text Settings**
 - Indent Using: Spaces
 - Widths: 4 (Tab), 4 (Indent)
 - Wrap lines

Project Navigator showing the file structure for the HelloWorld project:

- ▼ HelloWorld
 - AppDelegate.swift
 - ViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - HelloWorld.xcdatamodeld
 - ▼ HelloWorldTests
 - ▼ HelloWorldUITests
 - Products

Inspector showing the configuration for the HelloWorld target:

PROJECT

- ▼ Hello World
 - ▼ TARGETS
 - Hello World
 - HelloWorldTests
 - HelloWorldUITests

Identity

- Display Name: HelloWorld
- Bundle Identifier: com.imtku.HelloWorld
- Version: 1.0
- Build: 1

Signing

No accounts found
Add a developer account to sign your app.
[Add Account...](#)

Deployment Info

- Deployment Target: 10.0
- Device: iPhone (selected), iPad, Universal
- Main Interface: [dropdown]
- Device Orientation:
 - Portrait
 - Upside Down
 - Landscape Left
 - Landscape Right
- Status Bar Style: Default
 - Hide status bar
 - Requires full screen

App Icons and Launch Images

- App Icons Source: Applcon

Inspector showing project and document settings:

Identity and Type

- Name: HelloWorld
- Location: Absolute
- Full Path: /Users/imyday/Documents/SMAP/iOS/HelloWorld/HelloWorld.xcodeproj

Project Document

- Project Format: Xcode 3.2-compatible
- Organization: imtku
- Class Prefix: [empty]

Text Settings

- Indent Using: Spaces
- Widths: Tab: 4, Indent: 4
- Wrap lines

Class Library Reference Section:

- Cocoa Touch Class** - A Cocoa Touch class
- UI Test Case Class** - A class implementing a unit test
- Unit Test Case Class** - A class implementing a unit test

Main.storyboard (UI)

The screenshot displays the Xcode IDE interface for editing a storyboard. The top menu bar includes File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, and Help. The status bar at the top indicates the project is 'HelloWorld: Ready' and the time is 'Today at 9:12 AM'. The left sidebar shows the project structure for 'HelloWorld', with 'Main.storyboard' selected. The main editor area shows a 'View Controller Scene' containing a 'View Controller' widget. The right sidebar contains the 'Identity and Type' panel, showing the storyboard's name, type, location, and full path. Below this are sections for 'On Demand Resource Tags', 'Interface Builder Document' (with options for 'Opens in', 'Builds for', and checkboxes for 'Use Auto Layout', 'Use Trait Variations', and 'Use as Launch Screen'), 'Global Tint', 'Localization' (with 'Base' selected), and a list of storyboard elements: 'View Controller', 'Storyboard Reference', and 'Navigation Controller'. The bottom status bar shows 'View as: iPhone 6s (w C h R)' and a zoom level of '75%'.

Main.storyboard (UI)

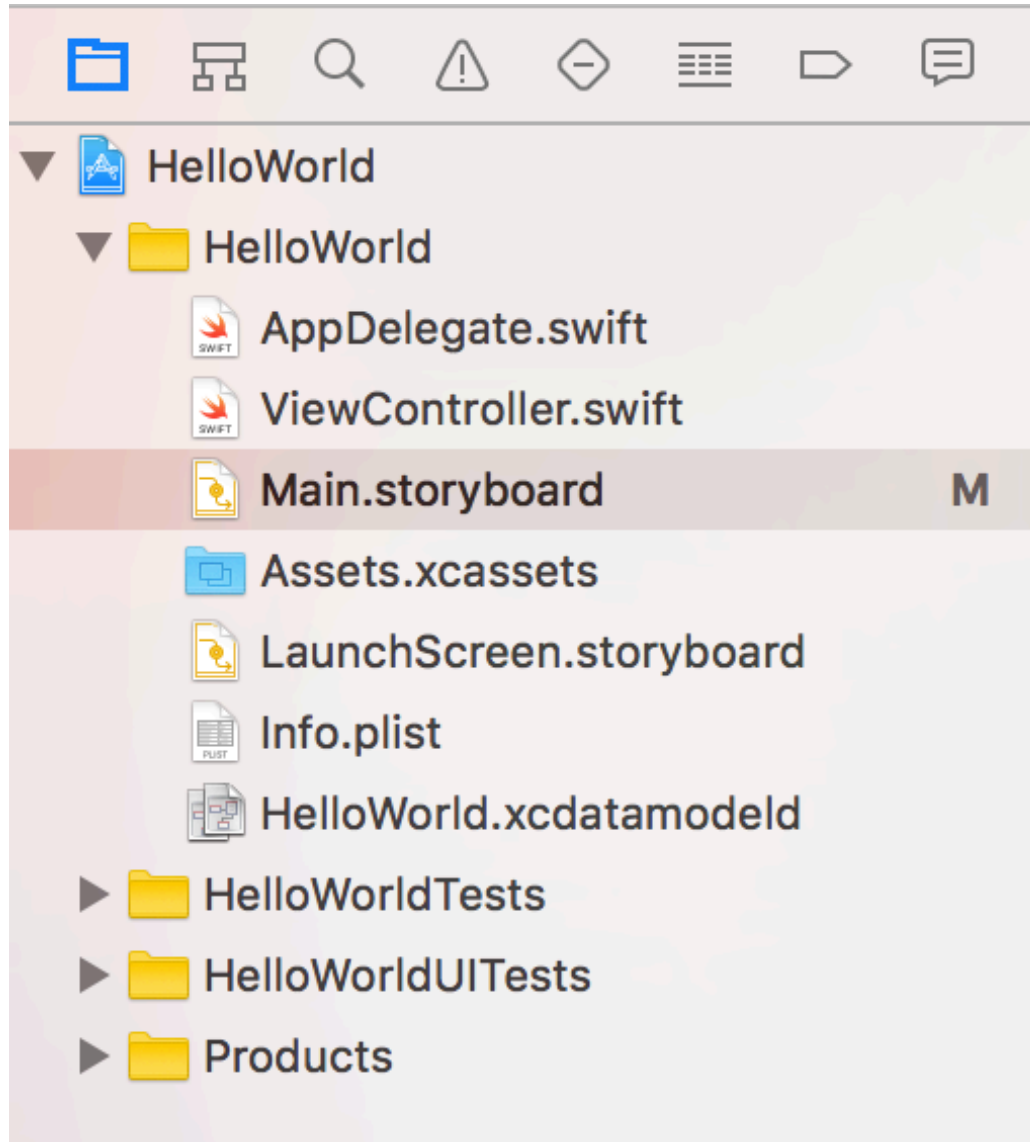
The screenshot displays the Xcode IDE with the Main.storyboard editor open. The top menu bar includes Xcode, File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, and Help. The toolbar below the menu bar contains various icons for navigation and editing. The project navigator on the left shows the project structure, with the Main.storyboard file selected. The storyboard canvas in the center shows a single blue view. The properties inspector on the right shows the following settings for the selected view:

- Content Mode: Scale To Fill
- Semantic: Unspecified
- Tag: 0
- Interaction: User Interaction Enabled, Multiple Touch
- Alpha: 1
- Background: [Color Picker]
- Tint: [Color Picker] Default
- Drawing: Opaque, Hidden, Clears Graphics Context, Clip To Bounds, Autocomplete Subviews
- Stretching: X: 0, Y: 0, Width: 1, Height: 1

At the bottom of the screen, the status bar shows "View as: iPhone 6s (w C h R)" and "75%".

Main.storyboard (UI)

ViewController.swift (Code)



Main.storyboard (UI)

ViewController.swift (Code)

The screenshot displays the Xcode interface for editing a storyboard scene. The main canvas shows a 'View Controller' scene with a 'View Controller' object. The right-hand sidebar contains the 'Simulated Metrics' and 'View Controller' property inspectors. The 'Simulated Metrics' section is highlighted with a red dashed border and includes settings for Size (Inferred), Status Bar (Inferred), Top Bar (Inferred), and Bottom Bar (Inferred). The 'View Controller' section includes settings for Title, Is Initial View Controller (checked), Layout (Adjust Scroll View Insets checked, Hide Bottom Bar on Push unchecked, Resize View From NIB checked, Use Full Screen (Deprecated) unchecked), Extend Edges (Under Top Bars checked, Under Bottom Bars checked, Under Opaque Bars unchecked), Transition Style (Cover Vertical), Presentation (Full Screen), Defines Context (unchecked), Provides Context (unchecked), and Content Size (Use Preferred Explicit Size unchecked, Width 375, Height 667). The bottom status bar shows 'View as: iPhone 6s (wC hR)' and a zoom level of 73%. A red dashed border also highlights the bottom status bar area, which includes device and orientation icons and a 'Vary for Traits' button.

Simulated Metrics

- Size: Inferred
- Status Bar: Inferred
- Top Bar: Inferred
- Bottom Bar: Inferred

View Controller

- Title: []
- Is Initial View Controller
- Layout:
 - Adjust Scroll View Insets
 - Hide Bottom Bar on Push
 - Resize View From NIB
 - Use Full Screen (Deprecated)
- Extend Edges:
 - Under Top Bars
 - Under Bottom Bars
 - Under Opaque Bars
- Transition Style: Cover Vertical
- Presentation: Full Screen
- Defines Context
- Provides Context
- Content Size:
 - Use Preferred Explicit Size
 - Width: 375
 - Height: 667

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

File Explorer:

- HelloWorld
 - AppDelegate.swift
 - ViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - HelloWorld.xcdatamodeld
 - HelloWorldTests
 - HelloWorldUITests
 - Products

View Controller Scene:

- View Controller
 - Top Layout Guide
 - Bottom Layout Guide
 - View
 - First Responder
 - Exit
 - Storyboard Entry Point

Canvas:

View as: iPhone 6s (w C h R) — 73% +

Device: [Icons for various iPhone and iPad models]

Orientation: [Icons for portrait and landscape orientations]

Vary for Traits

Simulated Metrics

- Size: **Inferred** (dropdown menu open)
 - Freeform
 - Page Sheet
 - Form Sheet
 - Detail Master
- Status Bar: Freeform
- Top Bar: Page Sheet
- Bottom Bar: Form Sheet
- View Controller: Detail Master
- Title:
 - Is Initial View Controller
- Layout:
 - Adjust Scroll View Insets
 - Hide Bottom Bar on Push
 - Resize View From NIB
 - Use Full Screen (Deprecated)
- Extend Edges:
 - Under Top Bars
 - Under Bottom Bars
 - Under Opaque Bars
- Transition Style: Cover Vertical
- Presentation: Full Screen
 - Defines Context
 - Provides Context
- Content Size:
 - Use Preferred Explicit Size
 - Width: 375
 - Height: 667

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

Navigation Controller - A controller that manages navigation through a hierarchy of views.

Label

The screenshot shows the Xcode IDE with the 'HelloWorld' project open on an iPhone 7 Plus. The interface is divided into several panels:

- Left Panel (Project Navigator):** Shows the project structure for 'HelloWorld', including 'Main.storyboard' and 'View Controller Scene'.
- Center Panel (Canvas):** Displays the storyboard scene for 'View Controller'. A 'Label' widget is being added to the scene, and a documentation popup is visible over it.
- Right Panel (Inspector):** Shows the 'Simulated Metrics' and 'View Controller' settings. The 'View Controller' section includes options like 'Is Initial View Controller', 'Adjust Scroll View Insets', and 'Extend Edges'. A red dashed box highlights the 'Label' widget icon in the bottom toolbar.

The 'Label' widget documentation popup contains the following text:

Label UILabel

Implements a read-only text view. A label can contain an arbitrary amount of text, but UILabel may shrink, wrap, or truncate the text, depending on the size of the bounding rectangle and properties you set. You can control the font, text color, alignment, highlighting, and shadowing of the text in the label.

Done

Button

The screenshot displays the Xcode IDE interface. On the left, the Project Navigator shows the 'HelloWorld' project with a 'View Controller Scene' containing a 'View Controller' with a 'View'. The right-hand pane shows the 'Simulated Metrics' and 'View Controller' settings. The 'View Controller' settings include: Title (empty), Is Initial View Controller (checked), Layout (Adjust Scroll View Insets checked, Hide Bottom Bar on Push unchecked, Resize View From NIB checked, Use Full Screen (Deprecated) unchecked), Extend Edges (Under Top Bars checked, Under Bottom Bars checked, Under Opaque Bars unchecked), Transition Style (Cover Vertical), and Presentation (Full Screen). Below these settings, a list of widgets is shown, with 'Button' highlighted in a red dashed box. The 'Button' widget is described as: 'Button - Intercepts touch events and sends an action message to a target object when it's tapped.' The 'Text' widget is described as: 'Text Field - Displays editable text and sends an action message to a target object when Return is tapped.'

Button UIButton

Implements a button that intercepts touch events and sends an action message to a target object when it's tapped. You can set the title, image, and other appearance properties of a button. In addition, you can specify a different appearance for each button state.

Done

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text **Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.

Text Field

The screenshot shows the Xcode IDE with a storyboard open. The storyboard is titled "HelloWorld: Ready" and is for an iPhone 7 Plus. The storyboard is currently in "View as: iPhone 6s" mode. The storyboard is showing a "View Controller Scene" with a "View Controller" containing a "View". A "Text Field" widget is being added to the view. A tooltip for the "Text Field" widget is visible, showing its name "Text Field" and class "UITextField", along with a description: "Displays a rounded rectangle that can contain editable text. When a user taps a text field, a keyboard appears; when a user taps Return in the keyboard, the keyboard disappears and the text field can handle the input in an application-specific way. UITextField supports overlay views to display additional information, such as a bookmarks icon. UITextField also provides a clear text control a user taps to erase the contents of the text field." The right-hand side of the interface shows the "Simulated Metrics" and "View Controller" settings. The "View Controller" settings include: Title (empty), Is Initial View Controller (checked), Adjust Scroll View Insets (checked), Hide Bottom Bar on Push (unchecked), Resize View From NIB (checked), Use Full Screen (Deprecated) (unchecked), Extend Edges (Under Top Bars, Under Bottom Bars, Under Opaque Bars checked), Transition Style (Cover Vertical), and Presentation (Full Screen). The "Simulated Metrics" section shows: Size (Inferred), Status Bar (Inferred), Top Bar (Inferred), and Bottom Bar (Inferred). The "Label" and "Button" sections are also visible, with "Label" described as "A variably sized amount of static text." and "Button" described as "Intercepts touch events and sends an action message to a target object when it's tapped." The "Segmented Control" section is also visible, with a description: "Displays multiple segments, each of which functions as a discrete button." The "Text Field" section is highlighted with a red dashed box, showing a description: "Text Field - Displays editable text and sends an action message to a target object when Return is tapped." The "Done" button is visible at the bottom right of the tooltip.

Text Field

The screenshot displays the Xcode IDE with a storyboard for an iPhone 7 Plus. The storyboard shows a 'Round Style Text Field' on a 'View' within a 'View Controller Scene'. A blue arrow points from the 'Text Field' property in the right-hand panel to the text field in the storyboard. The right-hand panel shows the following properties for the 'Text Field':

- Text: Plain
- Color: Default
- Font: System 14.0
- Alignment: Left
- Placeholder: Placeholder Text
- Background: Background Image
- Disabled: Disabled Background Image
- Border Style: ROUNDED_RECTANGLE
- Clear Button: Never appears
- Min Font Size: 17
- Adjust to Fit: checked
- Capitalization: None
- Correction: Default
- Spell Checking: Default

Below the properties, there is a list of UI components with their descriptions:

- Label** - A variably sized amount of static text.
- Button** - Intercepts touch events and sends an action message to a target object when it's tapped.
- Segmented Control** - Displays multiple segments, each of which functions as a discrete button.
- Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.

Text Field

The screenshot displays the Xcode interface for editing a storyboard. The top menu bar includes File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, and Help. The status bar shows 'HelloWorld: Ready' and 'Today at 9:24 AM'. The left sidebar shows the project structure for 'HelloWorld', with 'Main.storyboard' selected. The middle-left pane shows the 'View Controller Scene' hierarchy, with 'View Controller' > 'View' > 'Round Style Text...' selected. The main canvas shows a storyboard with a 'Round Style Text Field' widget. The right-hand pane shows the 'Text Field' properties, including Text (Plain), Color (Default), Font (System 14.0), Alignment, Placeholder (Placeholder Text), Background (Background Image), Disabled (Disabled Background Image), Border Style (Rounded), Clear Button (Never appears), Min Font Size (17), Adjust to Fit (checked), Capitalization (None), Correction (Default), and Spell Checking (Default). Below the properties pane, there are descriptions for 'Label', 'Button', 'Segmented Control', and 'Text Field'.

Text Field

Text: Plain

Color: Default

Font: System 14.0

Alignment: [Left, Center, Right, Justified]

Placeholder: Placeholder Text

Background: Background Image

Disabled: Disabled Background Image

Border Style: [None, Solid, Dashed, Rounded]

Clear Button: Never appears

Clear when editing begins

Min Font Size: 17

Adjust to Fit

Capitalization: None

Correction: Default

Spell Checking: Default

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

Label

The screenshot displays the Xcode interface for editing a storyboard. On the left, the Project Navigator shows the 'HelloWorld' project with 'Main.storyboard' selected. The middle-left pane shows the 'View Controller Scene' hierarchy, with 'View' > 'Label' selected. The main canvas shows a storyboard for an iPhone 6s, with a 'Label' widget placed on the screen. A blue arrow points from the 'Label' widget in the canvas to the 'Label' property list in the right-hand pane. The property list includes settings for Text, Color, Font, Alignment, Lines, Behavior, Baseline, Line Break, Autoshrink, Highlighted, Shadow, and Shadow Offset. Below the property list, a 'View' section provides a brief description of the Label widget: 'Label - A variably sized amount of static text.' Other widget descriptions for Button, Segmented Control, and Text are also visible.

Label

Text Plain

Label

Color Default

Font System 17.0

Alignment

Lines 1

Behavior Enabled Highlighted

Baseline Align Baselines

Line Break Truncate Tail

Autoshrink Fixed Font Size Tighten Letter Spacing

Highlighted Highlighted

Shadow Default

Shadow Offset 0 -1

Width Height

View

Label - A variably sized amount of static text.

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text **Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.

Label

The screenshot displays the Xcode IDE with a storyboard for an iPhone 7 Plus. The storyboard shows a single view containing a label with the text "Hello World". The right-hand pane shows the properties for the selected Label widget. The properties are as follows:

- Text:** Plain (font style), Hello World (text)
- Color:** Default (color)
- Font:** System 17.0 (font size)
- Alignment:** Center (text alignment)
- Lines:** 1 (number of lines)
- Behavior:** Enabled (checkbox), Highlighted (checkbox)
- Baseline:** Align Baselines (baseline)
- Line Break:** Truncate Tail (line break)
- Autoshrink:** Fixed Font Size (autoshrink)
- Tighten Letter Spacing:** (checkbox)
- Highlighted:** Default (highlighted color)
- Shadow:** Default (shadow)
- Shadow Offset:** 0 (width), -1 (height)

The bottom of the screen shows the status bar with "View as: iPhone 6s (wC hR)", "100%", and various tool icons.

Project Navigator showing the file structure of the HelloWorld app, including AppDelegate.swift, ViewController.swift, Main.storyboard, and various test files.

Storyboard Editor showing the layout of the View Controller Scene, including the View Controller, Top Layout Guide, Bottom Layout Guide, and the Hello World label.

Inspector for the selected Label, showing properties such as Text (Plain), Color (Orange), Font (System 24.0), Alignment (Center), Lines (1), and Behavior (Enabled).

Library and Documentation pane showing the Label class and its properties, along with a list of other UI components like Button, Segmented Control, and Text Field.

Button

The screenshot displays the Xcode IDE with a storyboard for an iPhone 7 Plus. The storyboard shows a "Hello World" label and a "Button" below it. The right-hand pane shows the "Button" properties, including "Type" (System), "State Config" (Default), "Title" (Plain), "Font" (System 15.0), and "Text Color" (Default). The "Button" property is highlighted with a red dashed box. A blue arrow points from the "Button" property in the right-hand pane to the "Button" in the storyboard.

Button

Type: System

State Config: Default

Title: Plain

Button

Font: System 15.0

Text Color: Default

Shadow Color: Default

Image: Default Image

Background: Default Background Image

Shadow Offset: 0 (Width), 0 (Height)

Reverses On Highlight:

Shows Touch On Highlight:

Highlighted Adjusts Image:

Disabled Adjusts Image:

Line Break: Truncate Middle

Control

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

Button

The screenshot displays the Xcode IDE with a storyboard for an iPhone 7 Plus. The storyboard shows a 'Hello World' label and a 'Hello' button. The right-hand inspector shows the properties for the selected 'Hello' button, including its type, state configuration, title, font, and shadow settings. Below the inspector, there is a 'Control' section with a list of UI controls and their descriptions.

Button

- Type: System
- State Config: Default
- Title: Plain
- Text: Hello
- Font: System 15.0
- Text Color: Default
- Shadow Color: Default
- Image: Default Image
- Background: Default Background Image
- Shadow Offset: Width 0, Height 0
- Reverses On Highlight:
- Shows Touch On Highlight:
- Highlighted Adjusts Image:
- Disabled Adjusts Image:
- Line Break: Truncate Middle

Control

- Label** - A variably sized amount of static text.
- Button** - Intercepts touch events and sends an action message to a target object when it's tapped.
- Segmented Control** - Displays multiple segments, each of which functions as a discrete button.
- Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.

Set the active scheme

iOS Simulator: iPhone

The screenshot shows the Xcode IDE with the following components:

- Menu Bar:** Xcode, File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, Help.
- Toolbar:** Includes play, stop, and other simulation controls. The 'iPhone 7 Plus' simulator is selected and highlighted with a red dashed box.
- Project Navigator (Left):** Shows the project structure for 'HelloWorld', including 'Main.storyboard'.
- Document Navigator (Middle-Left):** Shows the 'View Controller Scene' hierarchy, with 'Hello' selected under the 'View'.
- Canvas (Center):** Displays the 'Hello World' app running on an iPhone simulator. The text 'Hello World' is visible in orange. A grey arrow points from the 'Hello' element in the Document Navigator to the simulator.
- Inspector (Right):** Shows the 'Button' inspector with various settings like Type (System), Title (Hello), and Font (System 15.0).
- Bottom Bar:** Shows the active scheme as 'iPhone 6s (w C h R)' and the zoom level at 100%.

Button

Type: System

State Config: Default

Title: Plain

Text: Hello

Font: System 15.0

Text Color: Default

Shadow Color: Default

Image: Default Image

Background: Default Background Image

Shadow Offset: Width 0, Height 0

Reverses On Highlight

Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break: Truncate Middle

Control

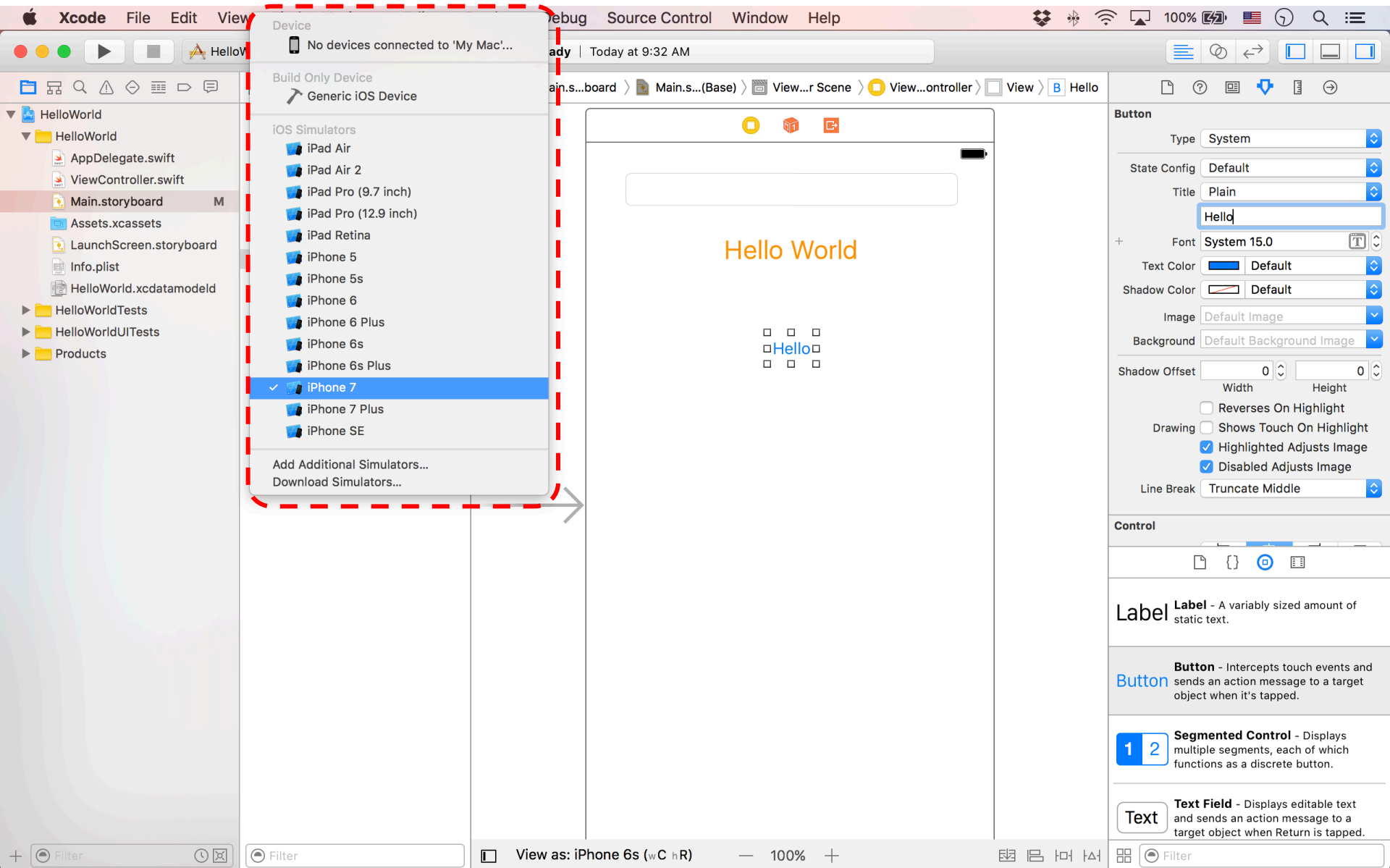
Label - A variably sized amount of static text.

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text - Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

iOS Simulators: iPhone 7



Build and Run

The screenshot shows the Xcode IDE with the 'HelloWorld' project open. The main canvas displays a simulated iPhone 7 with the text 'Hello World' and a 'Build Succeeded' message. The right sidebar lists various UI components:

- Triggered Segues**
 - action
- Outlet Collections**
 - gestureRecognizers
- Sent Events**
 - Did End On Exit
 - Editing Changed
 - Editing Did Begin
 - Editing Did End
 - Primary Action Triggered
 - Touch Cancel
 - Touch Down
 - Touch Down Repeat
 - Touch Drag Enter
 - Touch Drag Exit
 - Touch Drag Inside
 - Touch Drag Outside
 - Touch Up Inside
 - Touch Up Outside
 - Value Changed
- Referencing Outlets**
 - New Referencing Outlet
- Referencing Outlet Collections**
 - New Referencing Outlet Collection

Below the list, there are descriptions for 'Label', 'Button', 'Segmented Control', and 'Text' components.

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

iOS Simulator: iPhone 7 - iOS 10

The image shows the Xcode iOS Simulator interface. The main window displays a Hello World app on an iPhone 7. The app's status bar shows 'Carrier', signal strength, Wi-Fi, and the time '9:40 AM'. The app's content area features a white text field at the top, followed by the text 'Hello World' in orange, and 'Hello' in blue below it. A red dashed box highlights the simulator window, and a grey arrow points from it towards the right sidebar.

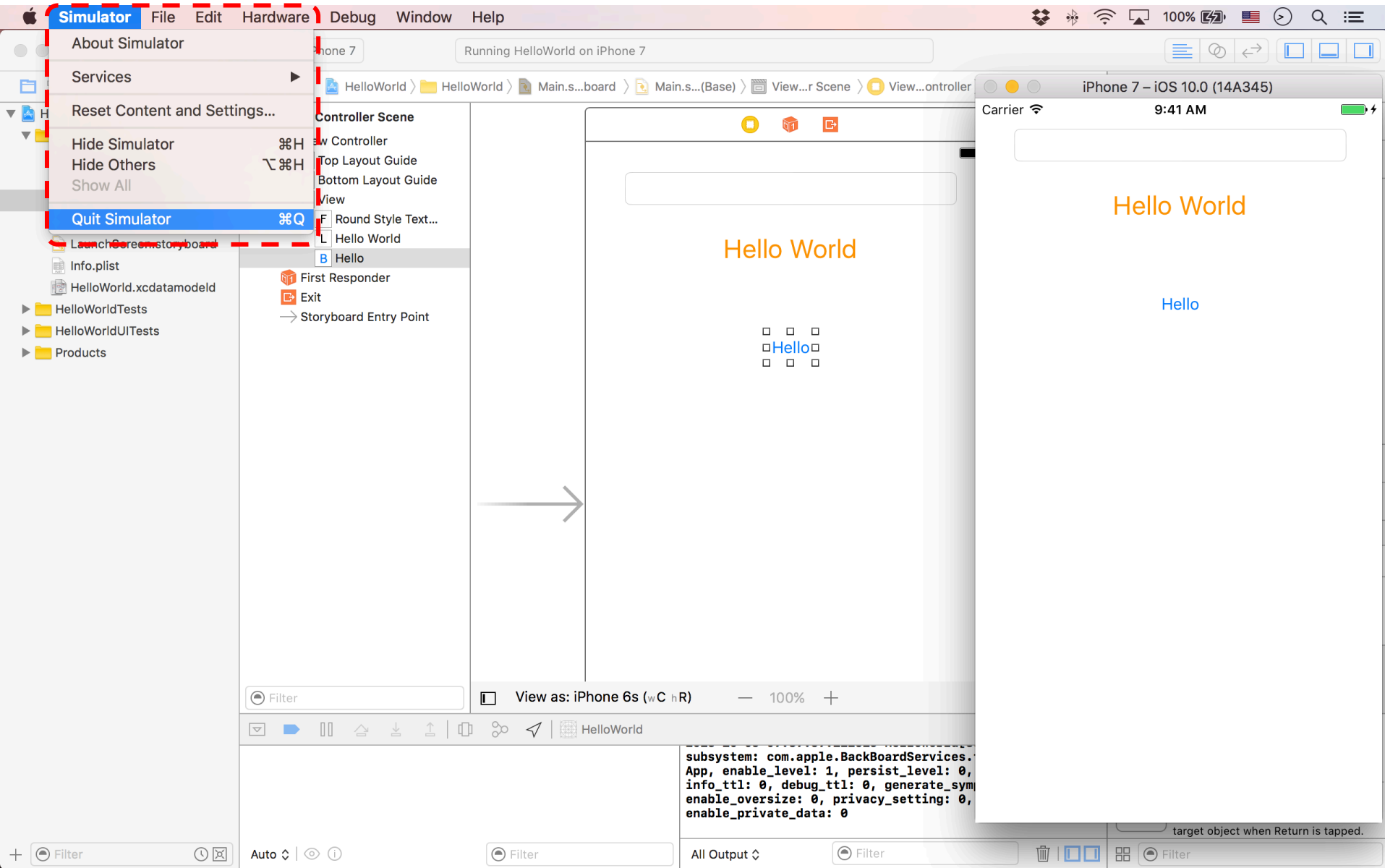
The right sidebar contains a list of UI components with their respective icons and descriptions:

- Triggered Segues**: action
- Outlet Collections**: gestureRecognizers
- Sent Events**: Did End On Exit, Editing Changed, Editing Did Begin, Editing Did End, Primary Action Triggered, Touch Cancel, Touch Down, Touch Down Repeat, Touch Drag Enter, Touch Drag Exit, Touch Drag Inside, Touch Drag Outside, Touch Up Inside, Touch Up Outside, Value Changed
- Referencing Outlets**: New Referencing Outlet
- Referencing Outlet Collections**: New Referencing Outlet Collection
- Label**: Label - A variably sized amount of static text.
- Button**: Button - Intercepts touch events and sends an action message to a target object when it's tapped.
- Segmented Control**: Segmented Control - Displays multiple segments, each of which functions as a discrete button.
- Text**: Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

The bottom of the simulator shows a console window with the following output:

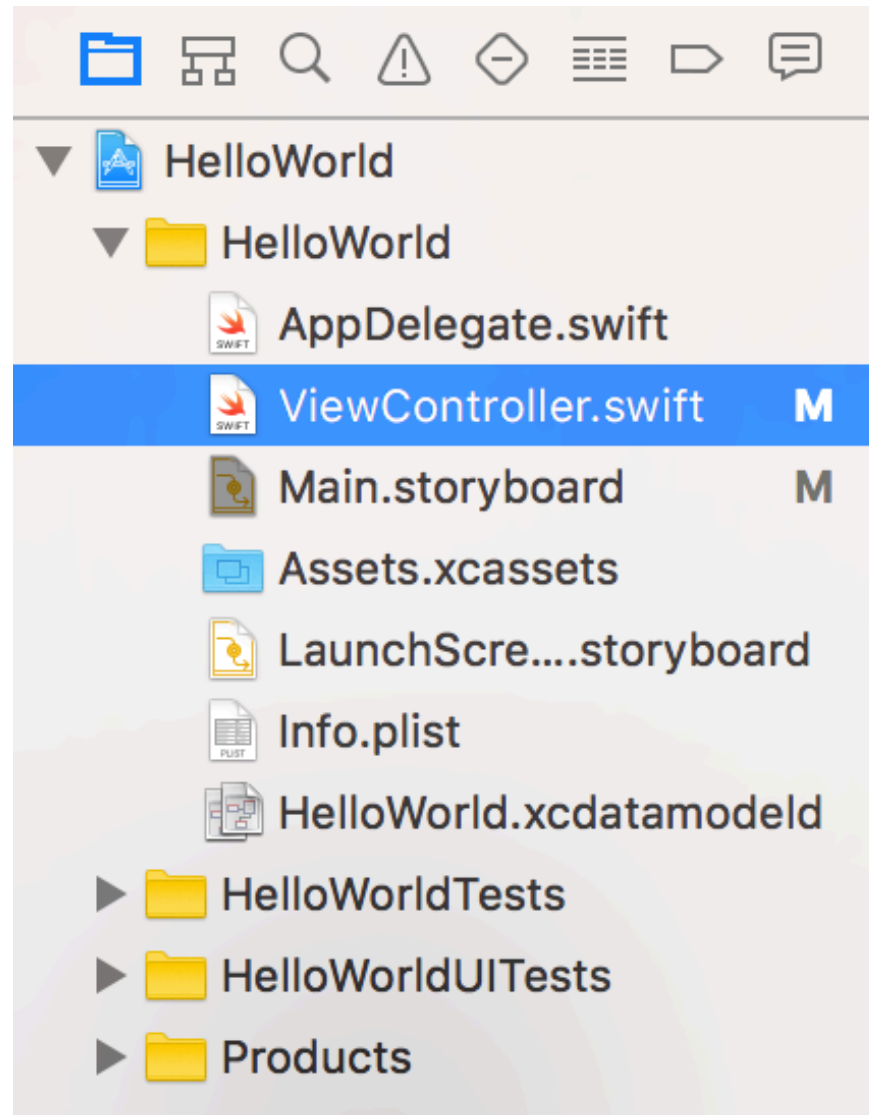
```
subsystem: com.apple.BackBoardServices.fence, category: App, enable_level: 1, persist_level: 0, default_ttl: 0, info_ttl: 0, debug_ttl: 0, generate_symptoms: 0, enable_oversize: 0, privacy_setting: 0, enable_private_data: 0
```

Simulator: Quit Simulator



Main.storyboard (UI)

ViewController.swift (Code)



ViewController.swift (Code)

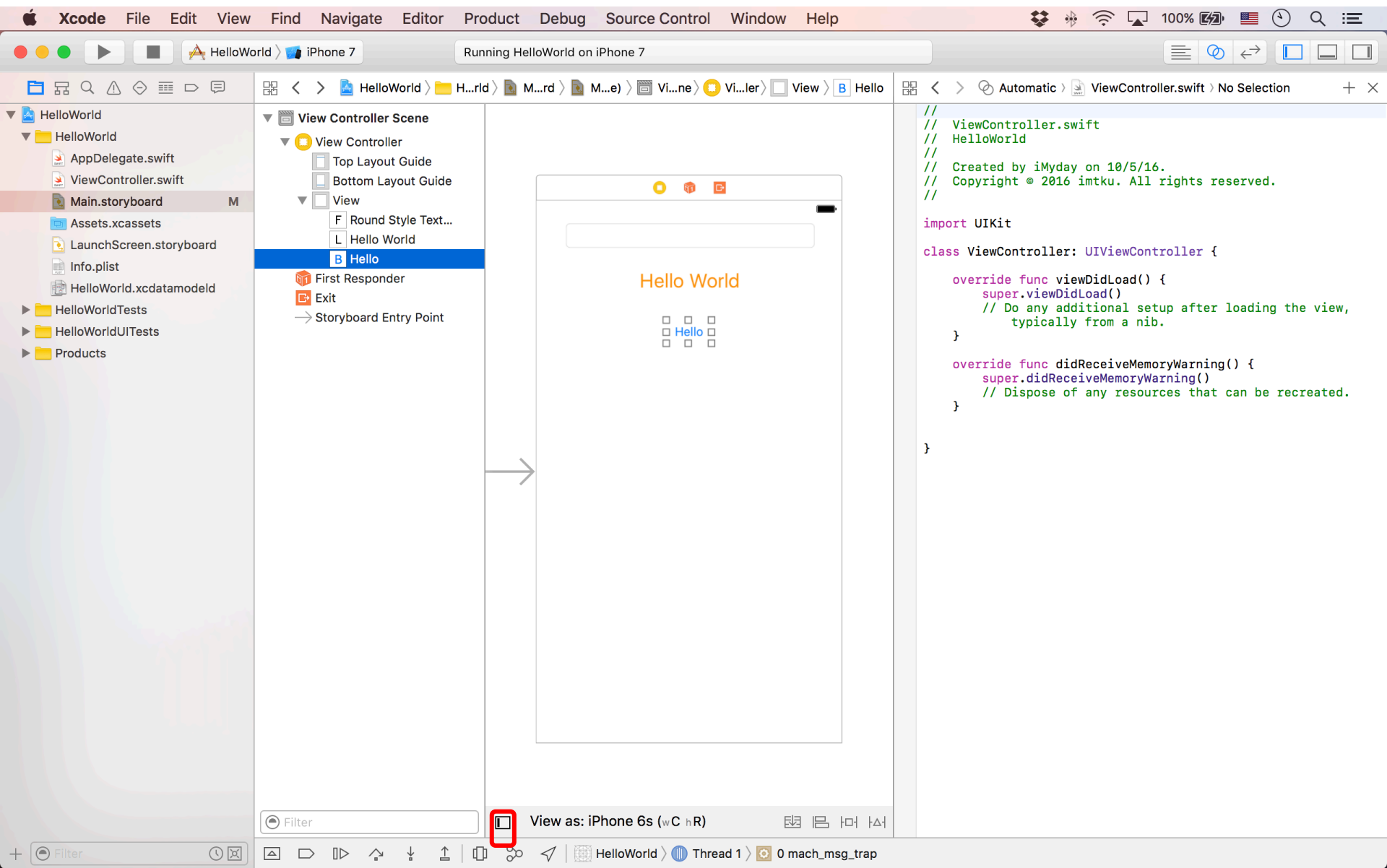
The screenshot displays the Xcode IDE with the following components:

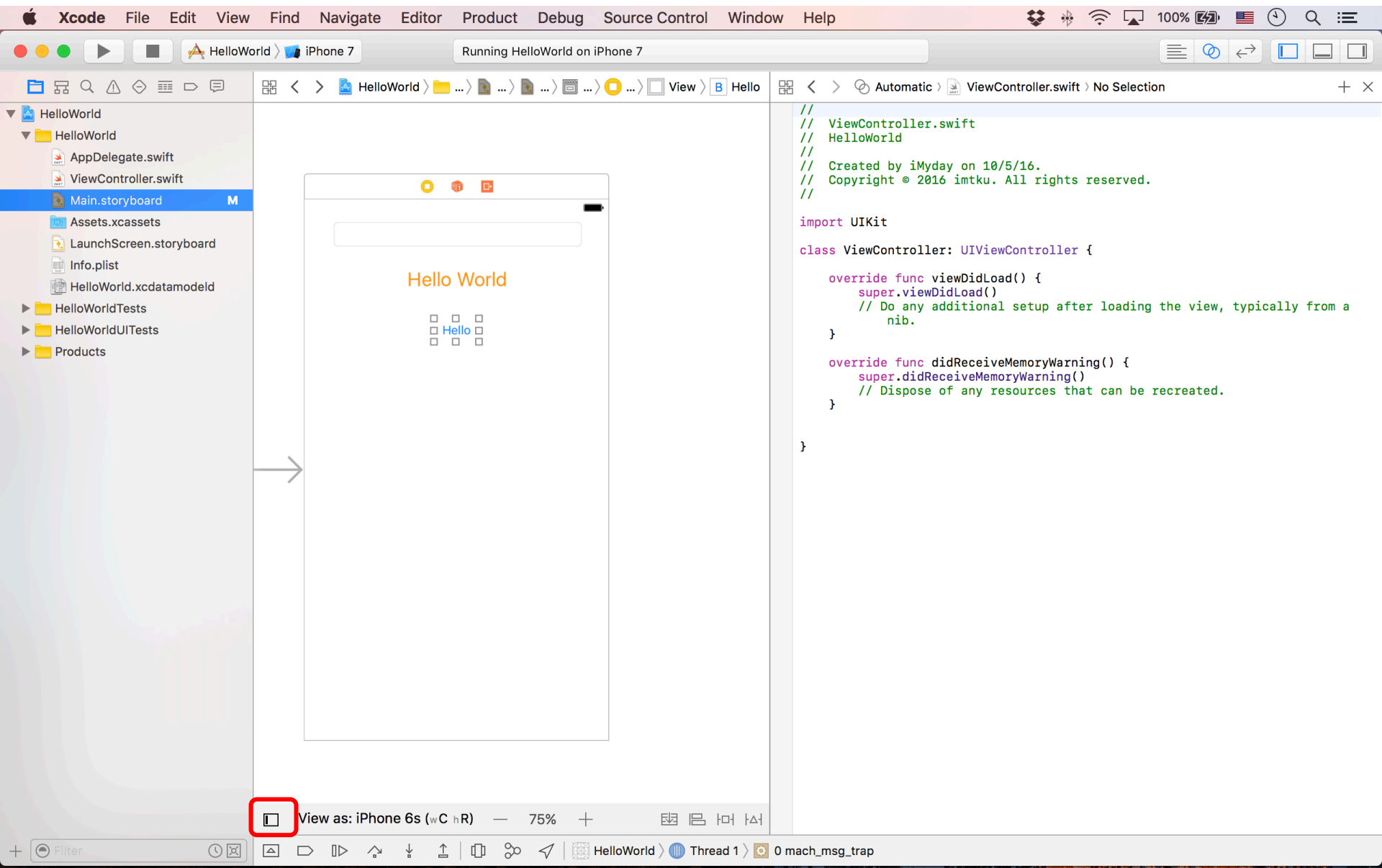
- Project Navigator (Left):** Shows the project structure for 'HelloWorld', including files like AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, and HelloWorld.xcdatamodeld.
- Editor (Center):** Displays the source code for ViewController.swift:

```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
}
```
- Identity and Type Panel (Right):** Shows the file's name as 'ViewController.swift', type as 'Default - Swift Source', and full path as '/Users/imyday/Documents/SMAP/iOS/HelloWorld/HelloWorld/ViewController.swift'.
- Target Membership:** Shows the file is assigned to the 'HelloWorld' target.
- Text Settings:** Shows 'Text Encoding' as 'Default - Unicode (UTF-8)' and 'Line Endings' as 'Default - macOS / Unix (LF)'.
- Component List (Bottom Right):** Lists UI components: Label, Button, Segmented Control, and Text Field, each with a brief description of its function.

IBOutlet and IBAction

- IBOutlet
 - Interface Builder Outlet
- IBAction
 - Interface Builder Action





```
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}
```

 View as: iPhone 6s (w C h R) — 75% +

Project Navigator showing file structure:

- HelloWorld
 - AppDelegate.swift
 - ViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - HelloWorld.xcdatamodeld
 - HelloWorldTests
 - HelloWorldUITests
 - Products

Storyboard preview for iPhone 7 showing a white background with the text "Hello World" in orange and a blue "Hello" button below it.

```
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the
        // view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be
        // recreated.
    }

}
```

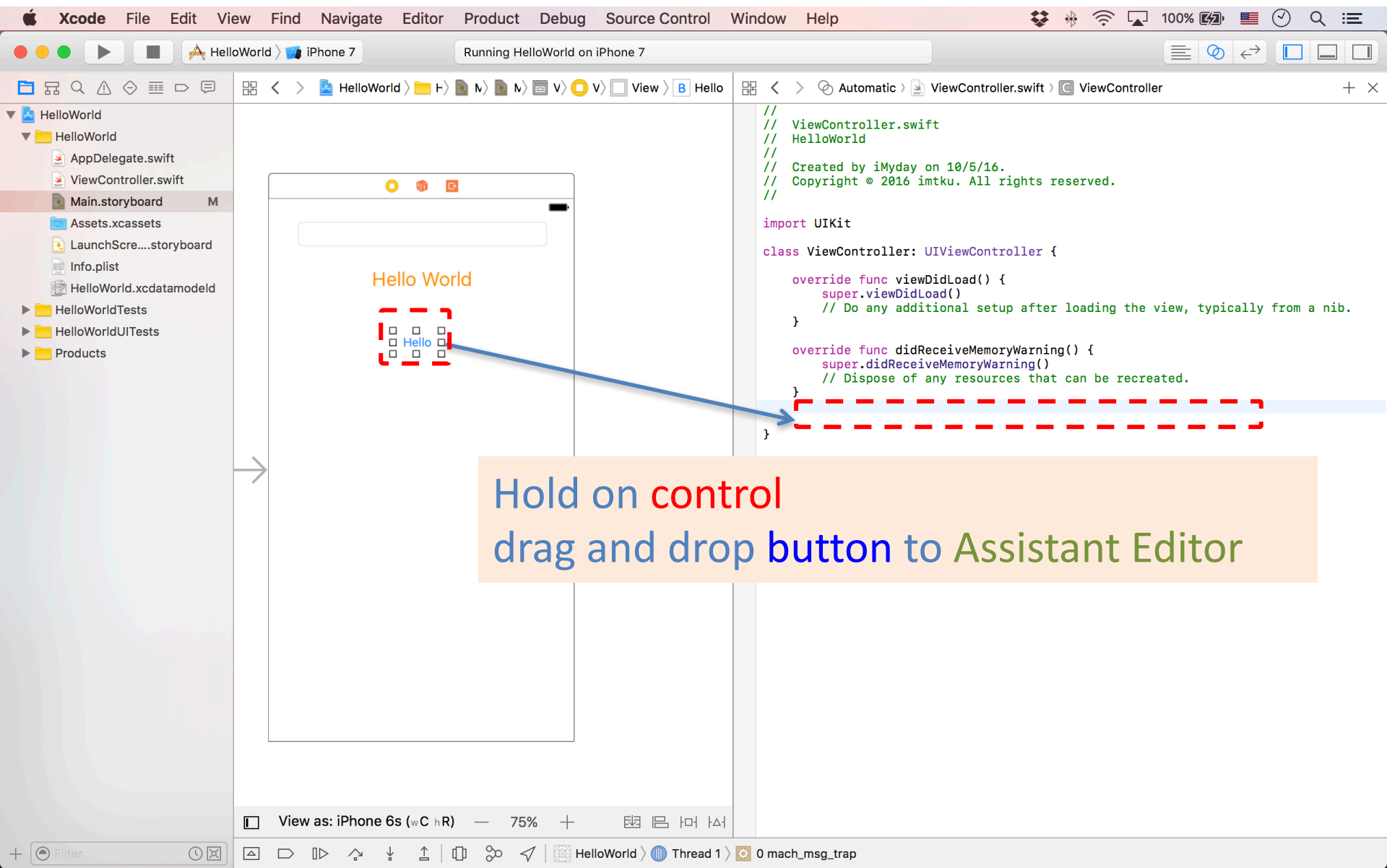
Inspector panel showing properties for a Button:

- Type: System
- State Config: Default
- Title: Plain
- Text: Hello
- Font: System 15.0
- Text Color: Default
- Shadow Color: Default
- Image: Default Image
- Background: Default Background Image
- Shadow Offset: 0 (Width), 0 (Height)
- Reverses On Highlight:
- Shows Touch On Highlight:
- Highlighted Adjusts Image:
- Disabled Adjusts Image:
- Line Break: Truncate Middle

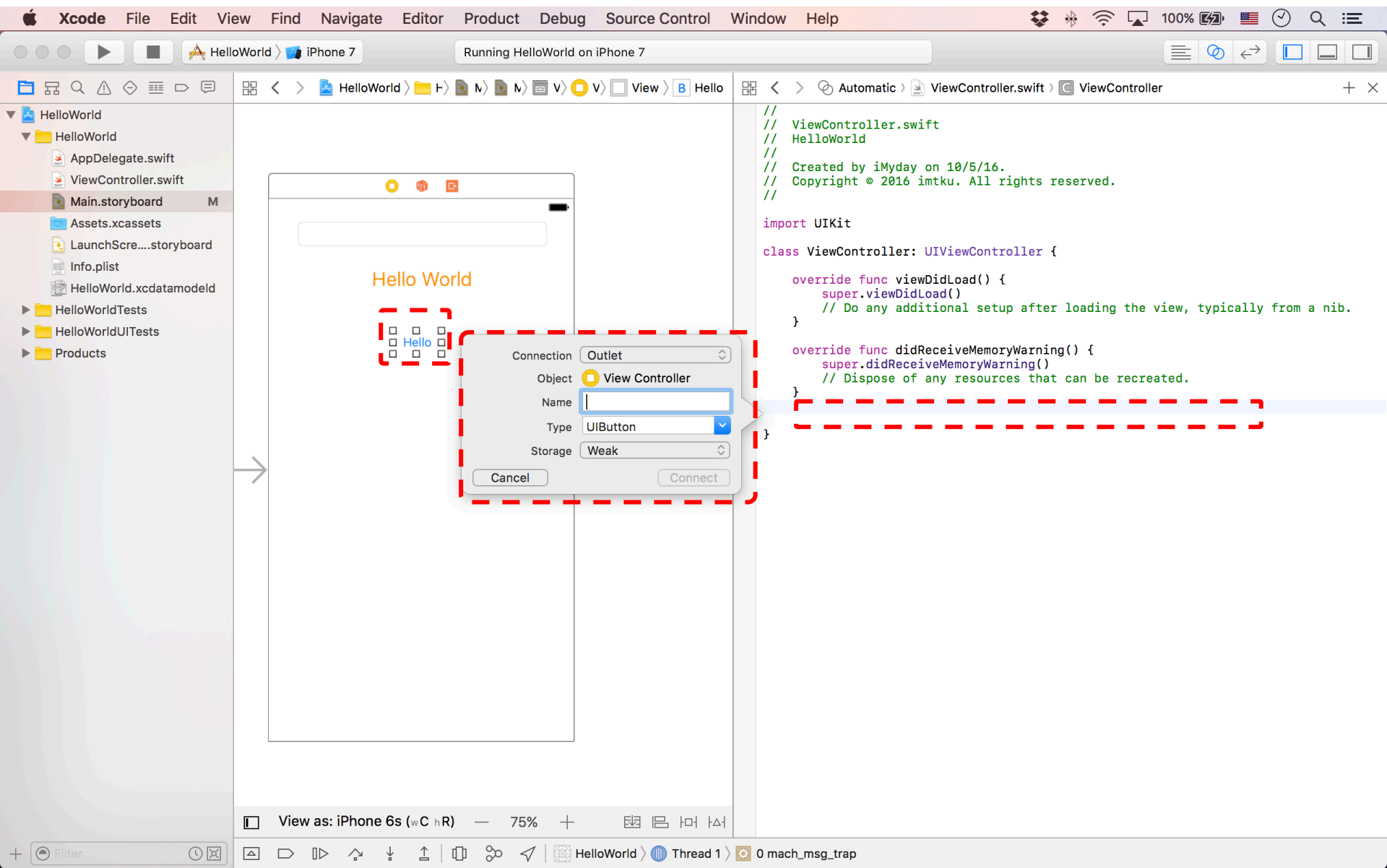
Control panel showing a list of UI controls:

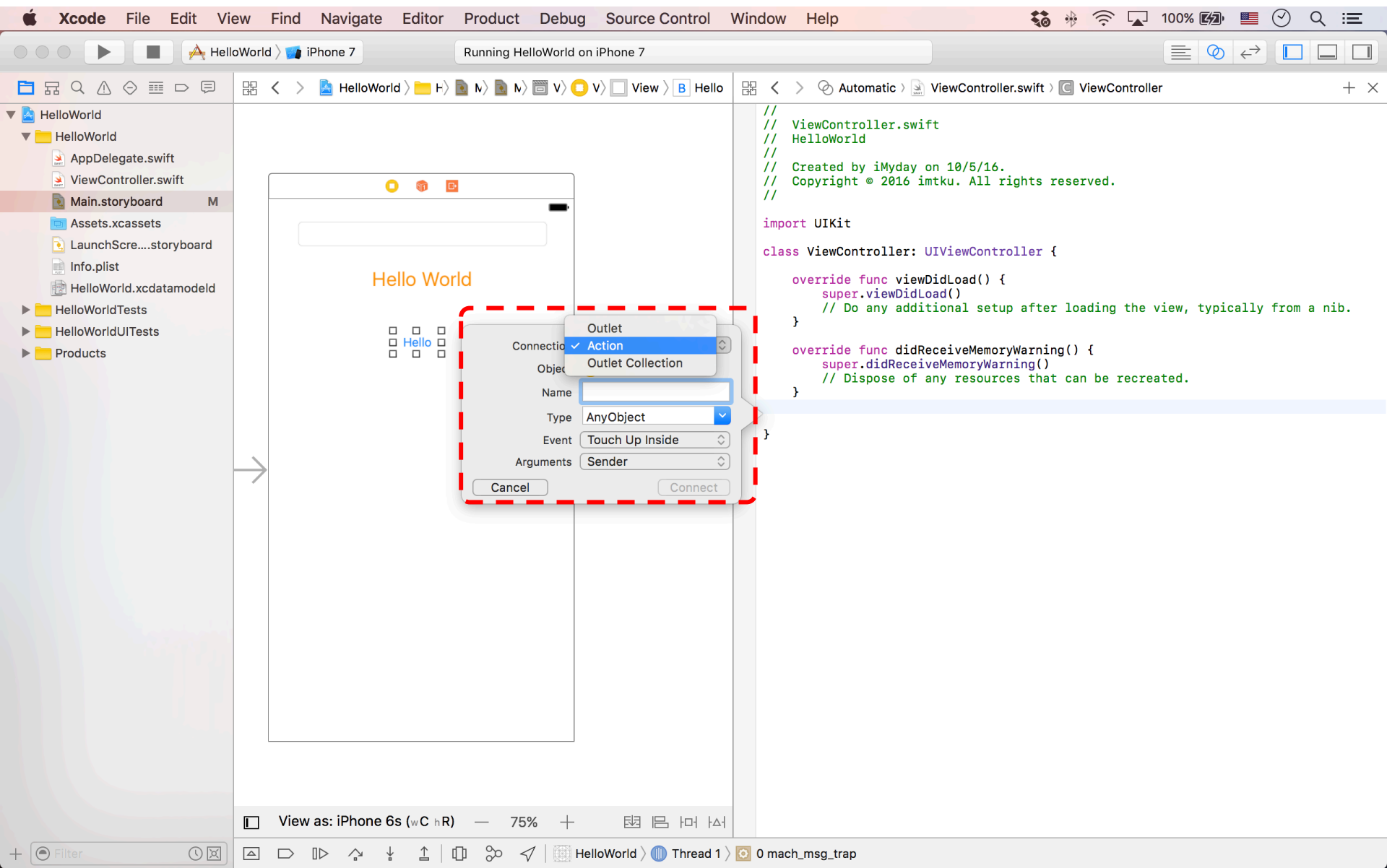
- Label: Label - A variably sized amount of static text.
- Button: Button - Intercepts touch events and sends an action message to a target object when it's tapped.
- Segmented Control: Segmented Control - Displays multiple segments, each of which functions as a discrete button.
- Text: Text Field - Displays editable text and sends an action message to a target object when Return is tapped.

View as: iPhone 6s (w C h R)



Hold on control
drag and drop button to Assistant Editor





btnHello

The screenshot shows the Xcode IDE with a project named 'HelloWorld' for an iPhone 7. The interface is divided into three main panes:

- Left Pane (Project Navigator):** Shows the project structure. The 'Main.storyboard' file is selected.
- Center Pane (Storyboard):** Displays a storyboard for an iPhone 6s. A button is visible with the text 'Hello World'. A red dashed box highlights the button, and a red dashed box highlights the 'Connection Inspector' panel.
- Right Pane (Code Editor):** Shows the Swift code for 'ViewController.swift'. The code includes a class definition for 'ViewController' that inherits from 'UIViewController'. It has two overridden methods: 'viewDidLoad()' and 'didReceiveMemoryWarning()'. The 'viewDidLoad()' method is currently selected.

The 'Connection Inspector' panel shows the following configuration for the selected button:

- Connection: Action
- Object: View Controller
- Name: btnHello
- Type: AnyObject
- Event: Touch Up Inside
- Arguments: Sender

The Swift code in the right pane is as follows:

```
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}
```

btnHello

The screenshot displays the Xcode IDE interface. On the left, the project navigator shows the 'HelloWorld' project with files like AppDelegate.swift, ViewController.swift, Main.storyboard, and Assets.xcassets. The main editor is split into two panes. The left pane shows a storyboard for an iPhone 7 simulator, featuring a white background with the text 'Hello World' in orange and a red button labeled 'Hello'. The right pane shows the ViewController.swift file with the following code:

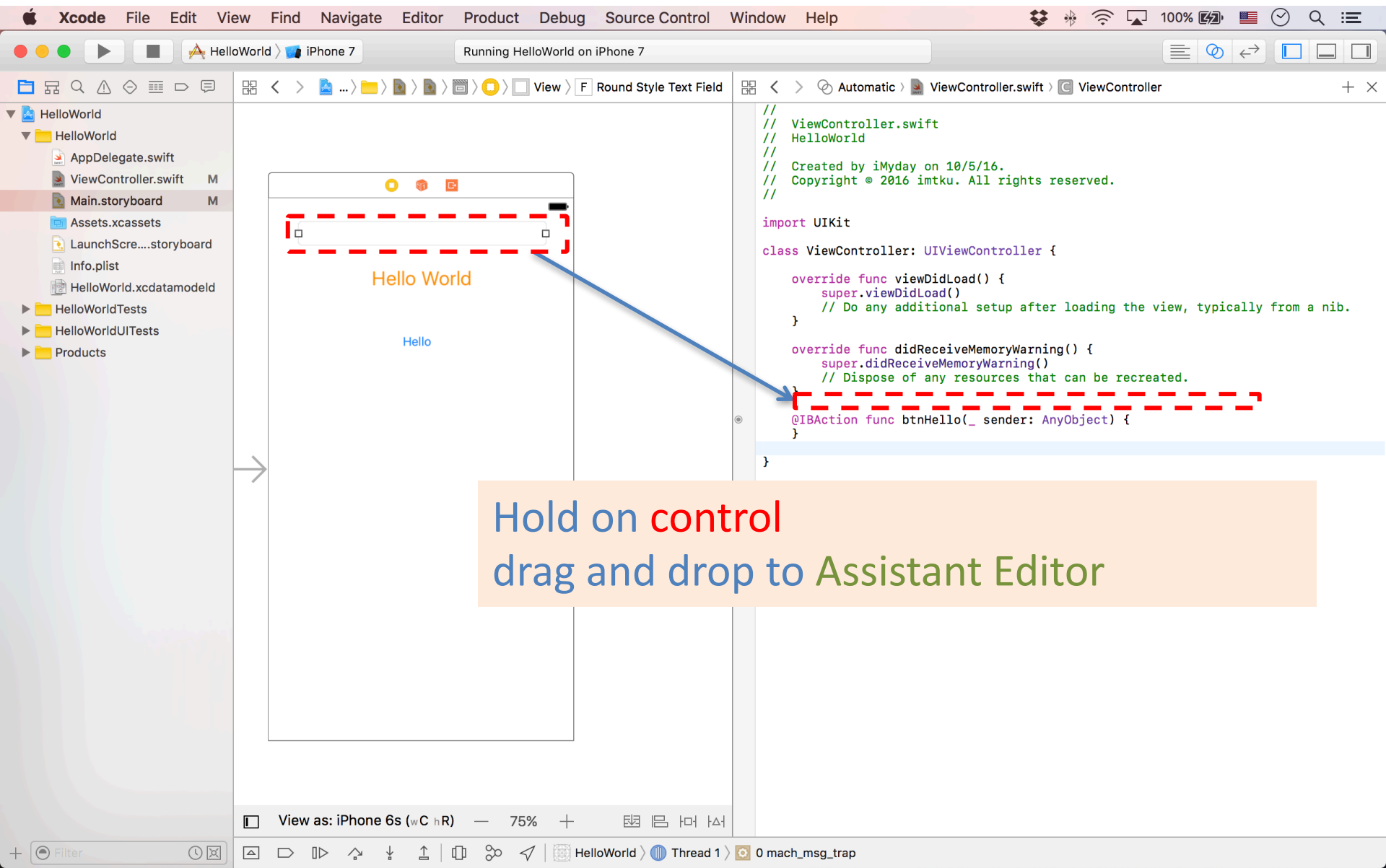
```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
    @IBAction func btnHello(_ sender: AnyObject) {  
    }  
}
```

A red dashed box highlights the @IBAction method signature in the Swift code. Below the storyboard, a red-bordered box contains the following code snippet:

```
@IBAction func btnHello(_ sender: AnyObject) {  
}
```

At the bottom of the image, a large orange box contains the text:

IBAction (Interface Builder Action)



Hold on control
drag and drop to Assistant Editor

txtYourName

The screenshot shows the Xcode IDE with the following components:

- Left Panel (Project Navigator):** Shows the project structure for 'HelloWorld' on an iPhone 7. Files include AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, and HelloWorld.xcdatamodeld.
- Center Panel (Storyboard):** Displays a storyboard for an iPhone 7. A 'Hello World' label is visible. A red dashed box highlights a 'Round Style Text Field' in the storyboard. A 'Hello' label is also present below the text field.
- Right Panel (Code Editor):** Shows the Swift code for 'ViewController.swift'. The code includes comments, imports, and class definitions. A red dashed box highlights the '@IBAction' method 'btnHello'.
- Connection Inspector:** A dialog box is open, showing the connection details for the text field. It is configured as an 'Outlet' connected to the 'View Controller' object, with the name 'txtYourName', type 'UITextField', and storage 'Weak'.
- Bottom Panel (Debug Console):** Shows the current view as 'iPhone 6s (w C h R)' at 75% zoom. The status bar at the bottom indicates 'HelloWorld' and 'Thread 1'.

txtYourName

The screenshot shows the Xcode IDE with a project named 'HelloWorld' running on an iPhone 7. The interface is split into three main areas:

- Left Panel (Project Navigator):** Shows the project structure with files like AppDelegate.swift, ViewController.swift, Main.storyboard, and Assets.xcassets.
- Center Panel (Storyboard):** Displays a storyboard for an iPhone 7. A text field is visible with the text 'Hello World' and 'Hello' below it. A red dashed box highlights the text field.
- Right Panel (Code Editor):** Shows the Swift code for ViewController.swift. The code includes imports, class declarations, and override methods. A red dashed box highlights the following line of code:

```
@IBOutlet weak var txtYourName: UITextField!
```

A red box highlights the code snippet in the right panel, and a red box highlights the code snippet in the center panel. An arrow points from the red box in the center panel to the red box in the right panel.

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

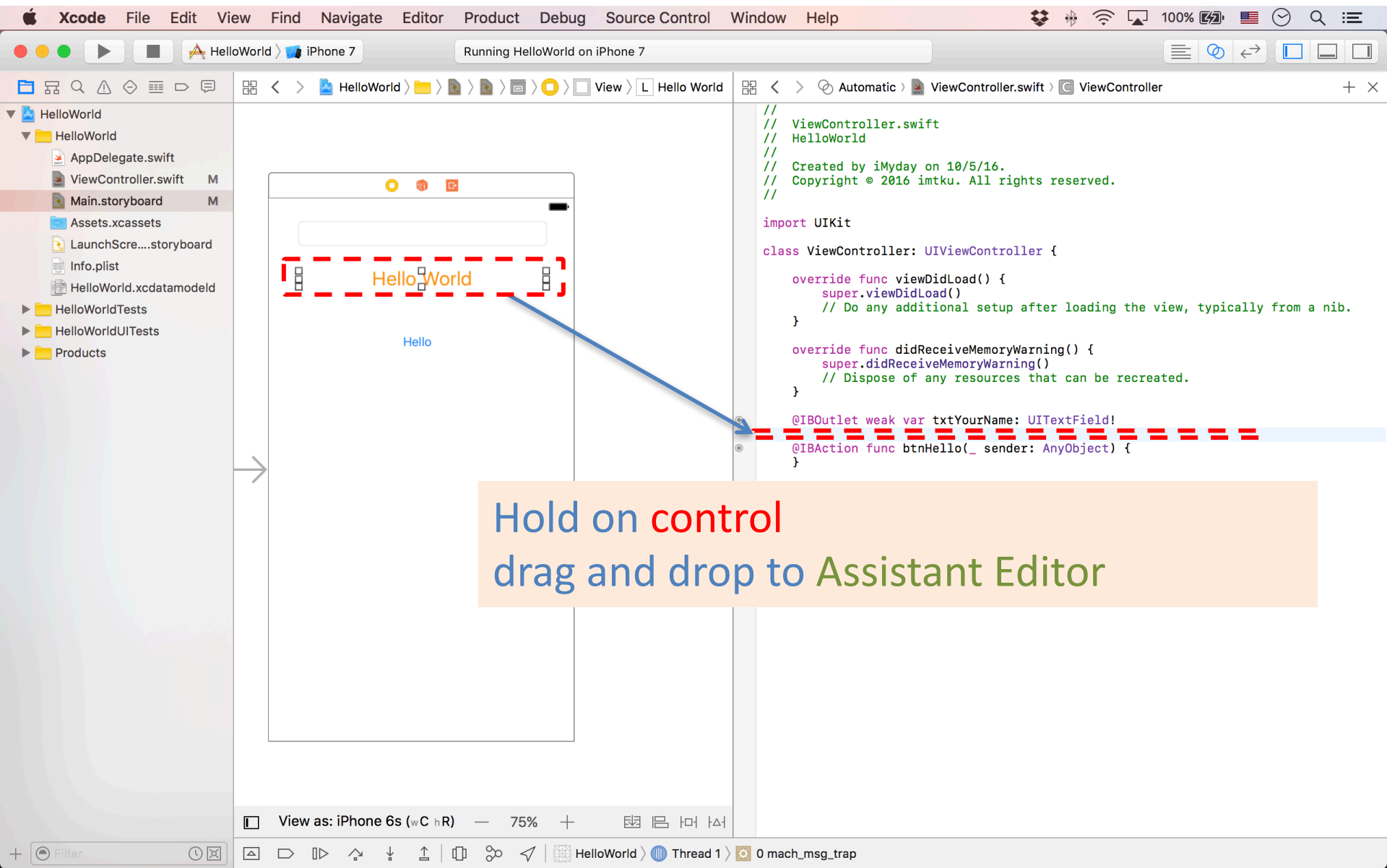
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var txtYourName: UITextField!

    @IBAction func btnHello(_ sender: AnyObject) {
    }
}
```

```
@IBOutlet weak var txtYourName: UITextField!
```

IBOutlet
(Interface Builder Outlet)



myLabel

The screenshot shows the Xcode IDE with the following components:

- Header:** Apple logo, menu bar (File, Edit, View, Find, Navigate, Editor, Product, Debug, Source Control, Window, Help), system status bar (100% battery, Wi-Fi, AirPlay, clock).
- Toolbar:** Play button, HelloWorld target, iPhone 7 simulator, Running HelloWorld on iPhone 7.
- Left Panel (Project Navigator):** Shows the project structure for 'HelloWorld', including AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, HelloWorld.xcdatamodeld, HelloWorldTests, HelloWorldUITests, and Products.
- Center Panel (Storyboard):** Displays a storyboard for an iPhone 7. A red dashed box highlights a 'HelloWorld' label. Below it, the text 'Hello' is visible.
- Right Panel (Code Editor):** Shows the Swift code for 'ViewController.swift'. The code includes comments, imports, and class methods. A red dashed box highlights the following code:

```
@IBOutlet weak var txtYourName: UITextField!  
@IBAction func btnHello(_ sender: AnyObject) {
```
- Dialog Box:** An 'IBOutlet' connection dialog is open, showing:
 - Connection: Outlet
 - Object: View Controller
 - Name: myLabel
 - Type: UILabel
 - Storage: Weak
 - Buttons: Cancel, Connect
- Bottom Panel:** View as: iPhone 6s (w C h R) - 75% +. Status bar: HelloWorld, Thread 1, 0 mach_msg_trap.

myLabel

The image shows the Xcode IDE interface. On the left, the project navigator shows the 'HelloWorld' project with a red dashed box around the 'HelloWorld' label in the storyboard. In the center, the storyboard shows a 'HelloWorld' label and a 'Hello' label. On the right, the code editor shows the 'ViewController.swift' file with the following code:

```
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.

import UIKit

class ViewController: UIViewController {

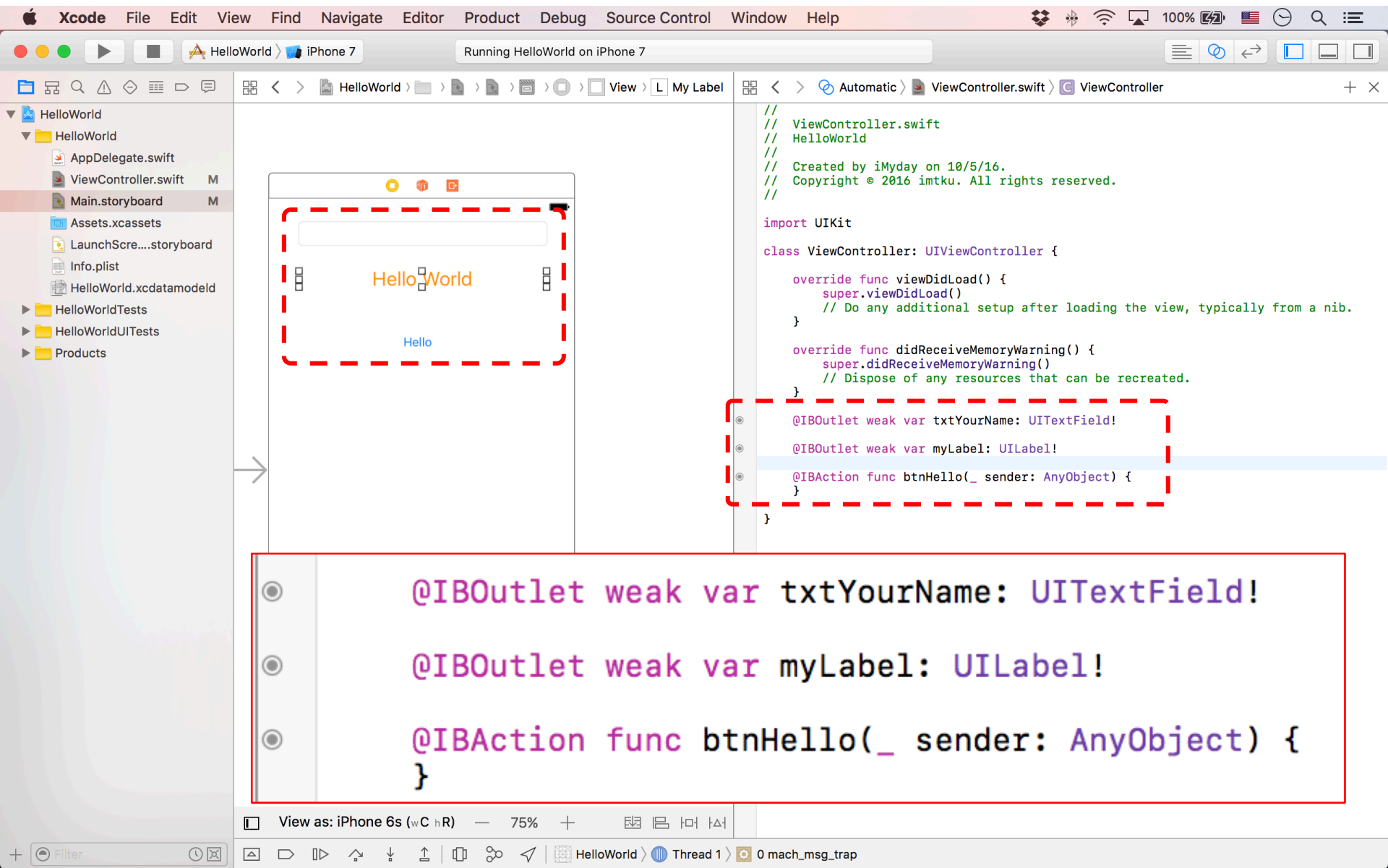
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var txtYourName: UITextField!
    @IBOutlet weak var myLabel: UILabel!
    @IBAction func btnHello(_ sender: AnyObject) {
    }

}
```

The line `@IBOutlet weak var myLabel: UILabel!` is highlighted with a red dashed box. Below the code editor, a blue box contains the text `@IBOutlet weak var myLabel: UILabel!`. Below that, an orange box contains the text `IBOutlet` and `(Interface Builder Outlet)`.



@IBAction func btnHello()

The image shows the Xcode IDE interface. On the left, the project navigator shows the 'HelloWorld' project with files like AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScre...storyboard, Info.plist, and HelloWorld.xcdatamodeld. The main canvas displays a storyboard for an iPhone 7. It features a text input field at the top, a 'Hello World' label in the center, and a 'Hello' label below it. A red dashed box highlights the @IBAction method in the ViewController.swift file:

```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
    @IBOutlet weak var txtYourName: UITextField!  
  
    @IBOutlet weak var myLabel: UILabel!  
  
    @IBAction func btnHello(_ sender: AnyObject) {  
        let strYourName:String! = txtYourName.text  
        myLabel.text = "Hello, " + strYourName  
        txtYourName.text = ""  
    }  
}
```

The status bar at the bottom indicates 'View as: iPhone 6s (wC hR)' at 75% zoom. The system tray shows 'HelloWorld', 'Thread 1', and '0 mach_msg_trap'.

@IBAction func btnHello()

The screenshot displays the Xcode IDE with a project named 'HelloWorld' running on an iPhone 7. The interface is split into three main areas: a file browser on the left, a storyboard in the center, and a code editor on the right.

File Browser (Left): Shows the project structure for 'HelloWorld', including files like AppDelegate.swift, ViewController.swift, Main.storyboard, and various test files.

Storyboard (Center): Shows a visual representation of the app's UI. It features a text input field at the top, a button labeled 'Hello World' in the middle, and a label 'Hello' below it.

Code Editor (Right): Shows the Swift code for 'ViewController.swift'. The code includes the following:

```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
    @IBOutlet weak var txtYourName: UITextField!  
  
    @IBOutlet weak var myLabel: UILabel!  
  
    @IBAction func btnHello(_ sender: AnyObject) {  
        let strYourName:String! = txtYourName.text  
        myLabel.text = "Hello, " + strYourName  
        txtYourName.text = ""  
    }  
}
```

A red box highlights the @IBAction function in the code editor:

```
@IBAction func btnHello(_ sender: AnyObject) {  
    let strYourName:String! = txtYourName.text  
    myLabel.text = "Hello, " + strYourName  
    txtYourName.text = ""  
}
```

The bottom status bar shows 'View as: iPhone 6s (wC hR) - 75% +', 'HelloWorld > Thread 1', and '0 mach_msg_trap'.

@IBAction func btnHello()

```
⦿ @IBOutlet weak var txtYourName: UITextField!  
⦿ @IBOutlet weak var myLabel: UILabel!  
⦿ @IBAction func btnHello(_ sender: AnyObject) {  
    let strYourName:String! = txtYourName.text  
    myLabel.text = "Hello, " + strYourName  
    txtYourName.text = ""  
}
```

Build and Run

The screenshot displays the Xcode IDE interface. At the top, the menu bar includes 'Xcode', 'File', 'Edit', 'View', 'Find', 'Navigate', 'Editor', 'Product', 'Debug', 'Source Control', 'Window', and 'Help'. The status bar shows 'Finished running HelloWorld on iPhone 7'. The Run button (a play icon) is highlighted with a red box. The interface is divided into three main sections:

- Left Panel (Project Navigator):** Shows the project structure for 'HelloWorld', including files like AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, and HelloWorld.xcdatamodeld.
- Center Panel (Canvas):** Displays a preview of the app running on an iPhone 7 simulator. The app shows a text input field at the top, followed by a label 'Hello World' in orange, and a button labeled 'Hello' in blue.
- Right Panel (Code Editor):** Shows the Swift code for ViewController.swift. The code includes comments, imports, and the implementation of UIViewController methods: viewDidLoad(), didReceiveMemoryWarning(), and btnHello().

```
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var txtYourName: UITextField!

    @IBOutlet weak var myLabel: UILabel!

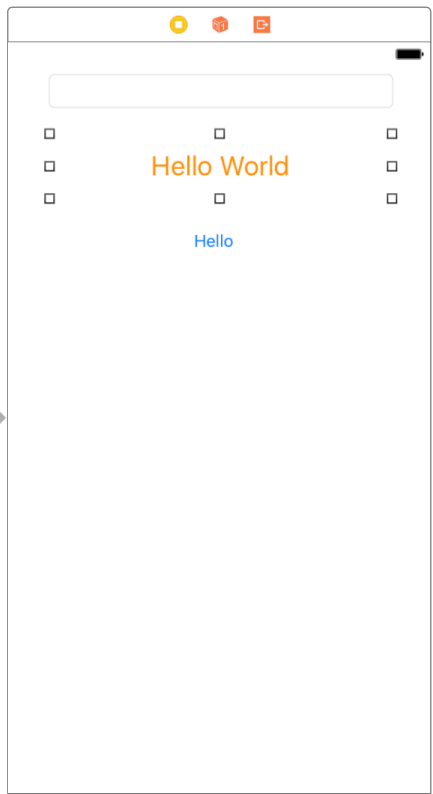
    @IBAction func btnHello(_ sender: AnyObject) {
        let strYourName:String! = txtYourName.text
        myLabel.text = "Hello, " + strYourName
        txtYourName.text = ""
    }

}
```

At the bottom, the status bar indicates 'View as: iPhone 6s (w C h R)' and '75%' zoom level.

- HelloWorld
 - AppDelegate.swift
 - ViewController.swift
 - Main.storyboard
 - Assets.xcassets
 - LaunchScre...storyboard
 - Info.plist
 - HelloWorld.xcdatamodeld
 - HelloWorldTests
 - HelloWorldUITests
 - Products

View > L My Label



Storyboard preview showing a text field, a label "Hello World", and a button "Hello".

```
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

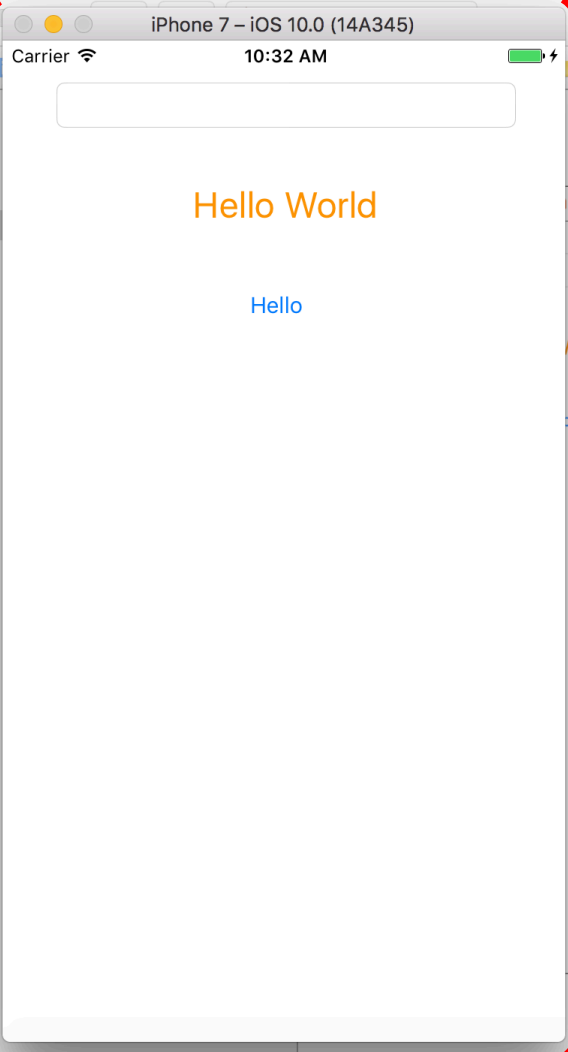
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var txtYourName: UITextField!

    @IBOutlet weak var myLabel: UILabel!

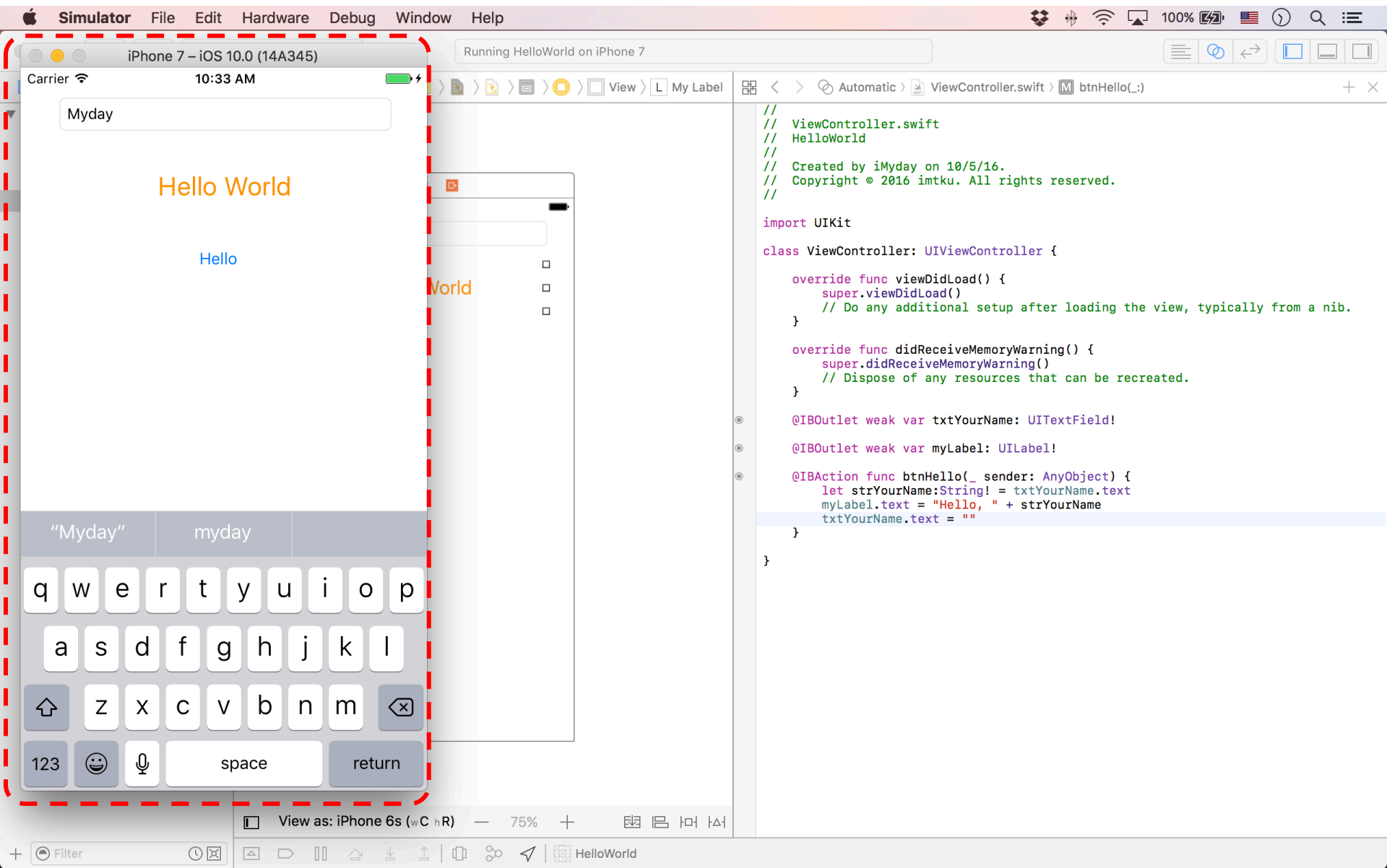
    @IBAction func btnHello(_ sender: AnyObject) {
        let strYourName:String! = txtYourName.text
        myLabel.text = "Hello, " + strYourName
        txtYourName.text = ""
    }
}
```

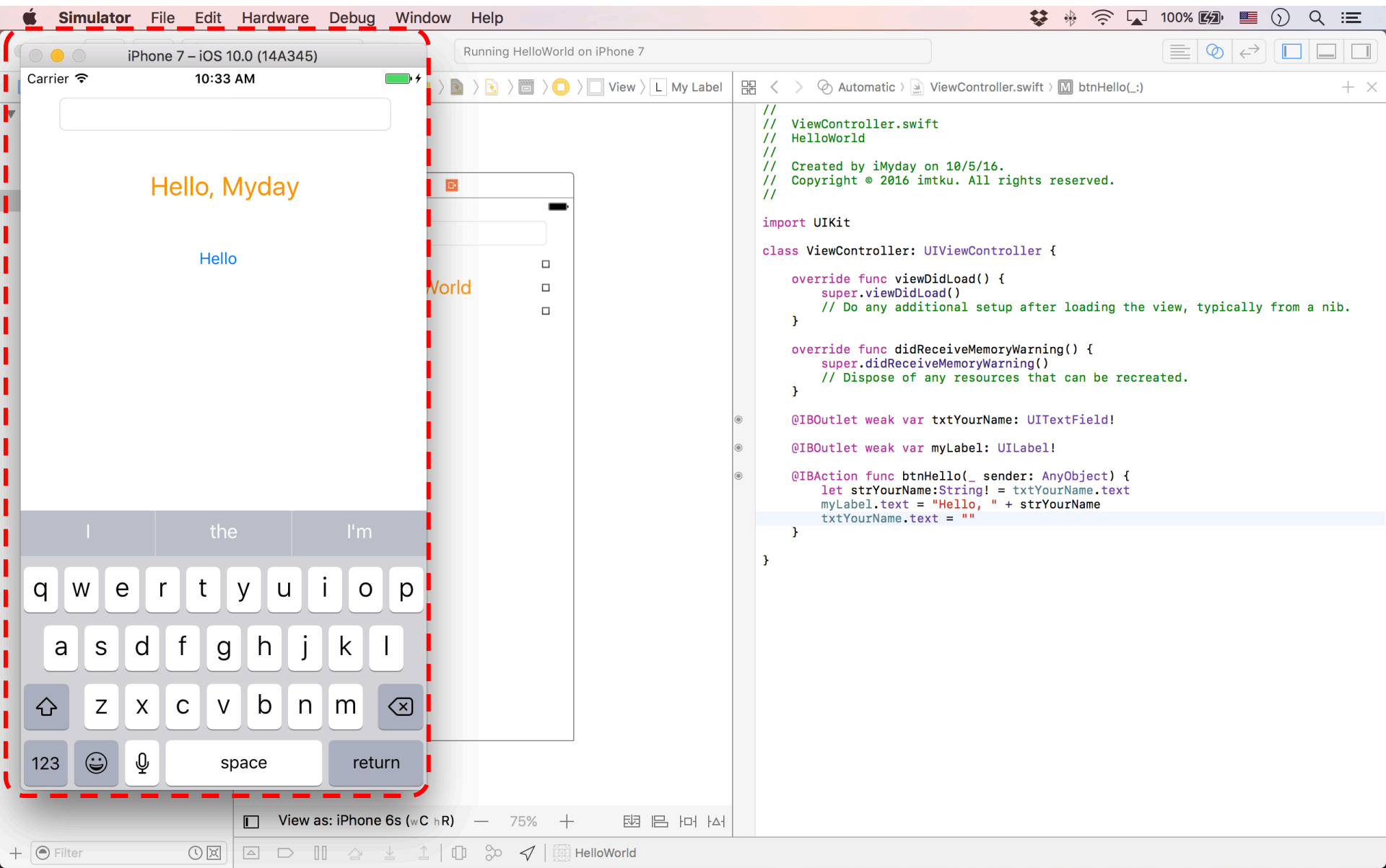




```
Running HelloWorld on iPhone 7  
View > L My Label  
Automatic > ViewController.swift > M btnHello(_:)  
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
import UIKit  
class ViewController: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
    @IBOutlet weak var txtYourName: UITextField!  
    @IBOutlet weak var myLabel: UILabel!  
    @IBAction func btnHello(_ sender: AnyObject) {  
        let strYourName:String! = txtYourName.text  
        myLabel.text = "Hello, " + strYourName  
        txtYourName.text = ""  
    }  
}
```







```
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.

import UIKit

class ViewController: UIViewController {

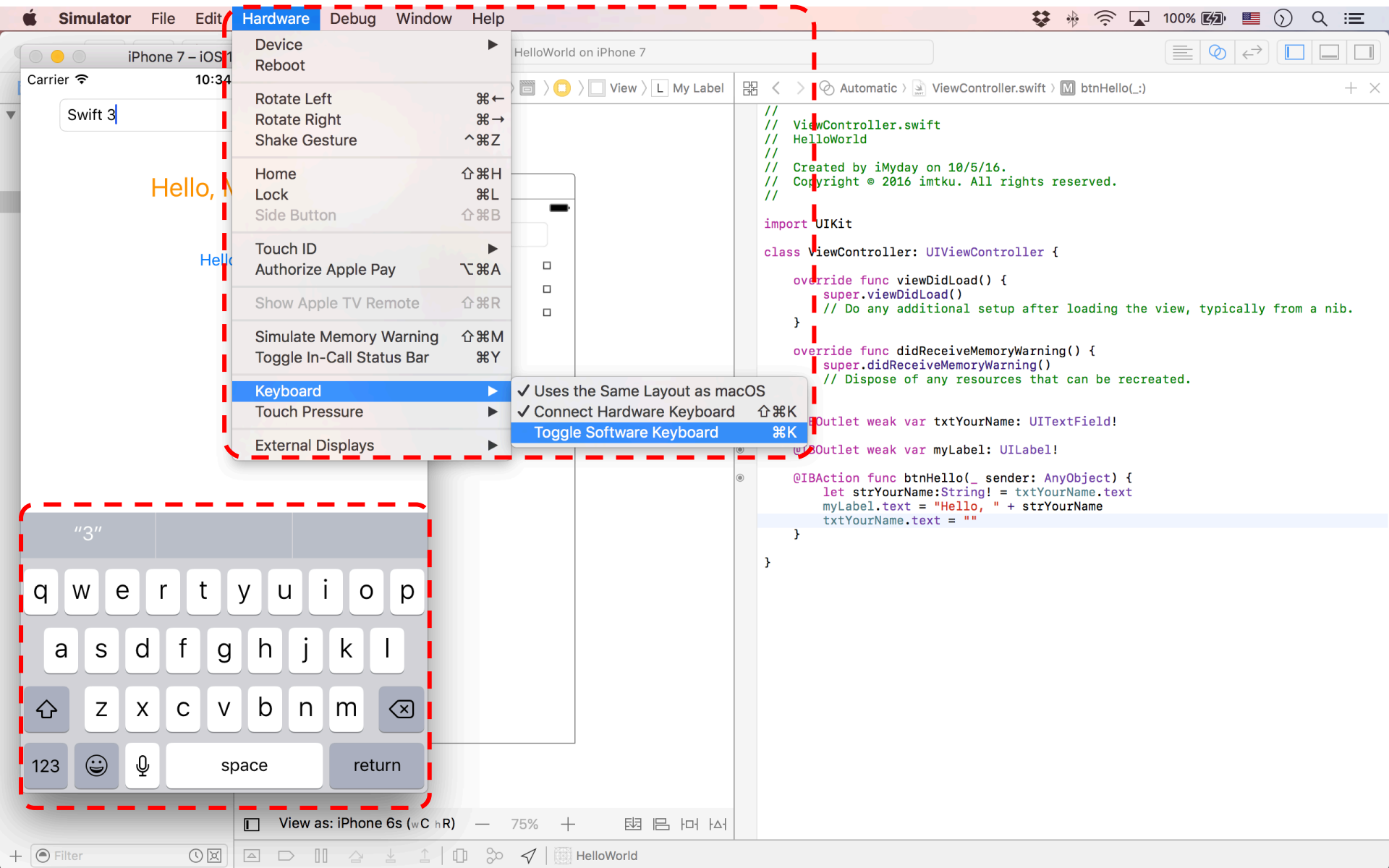
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

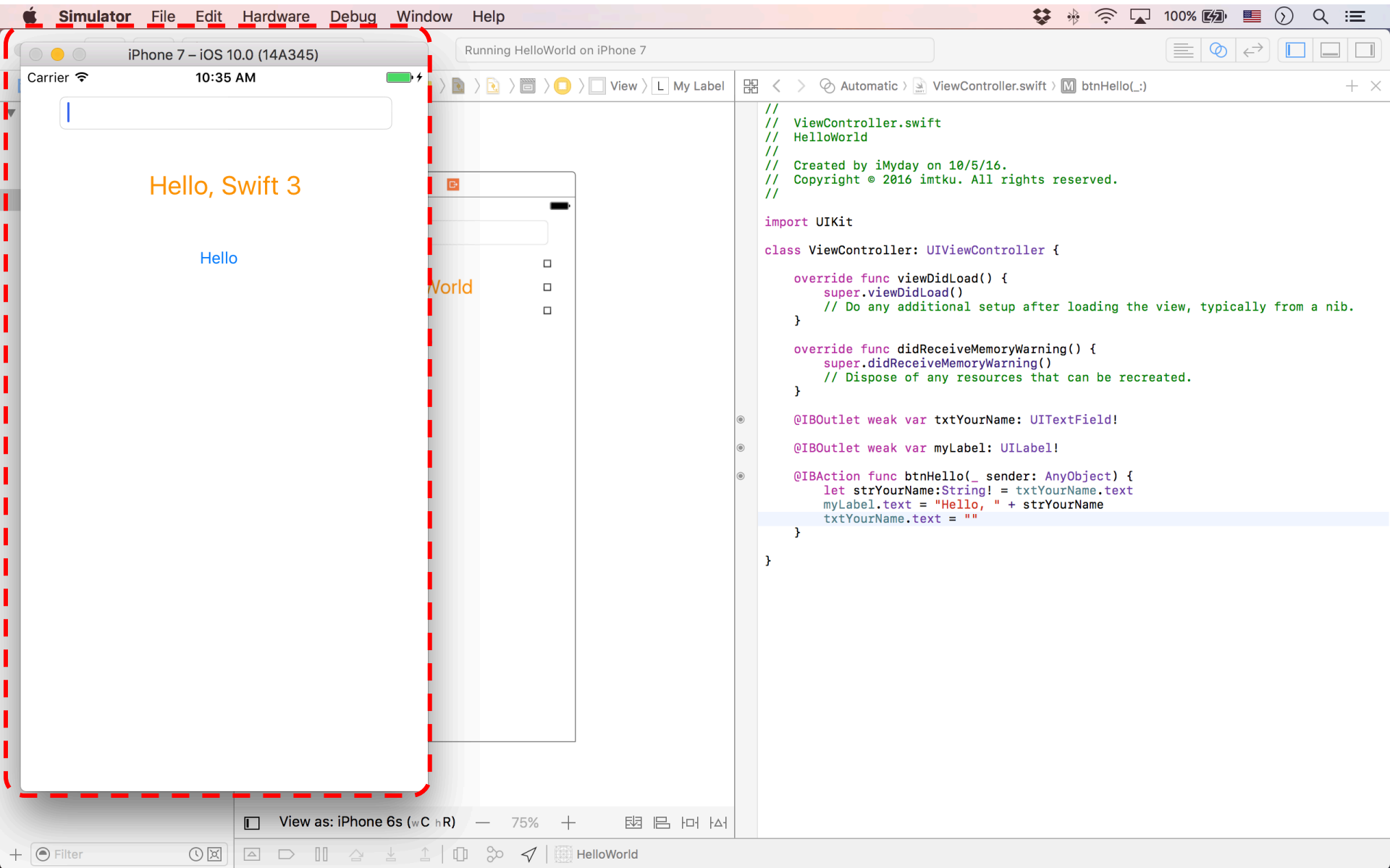
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

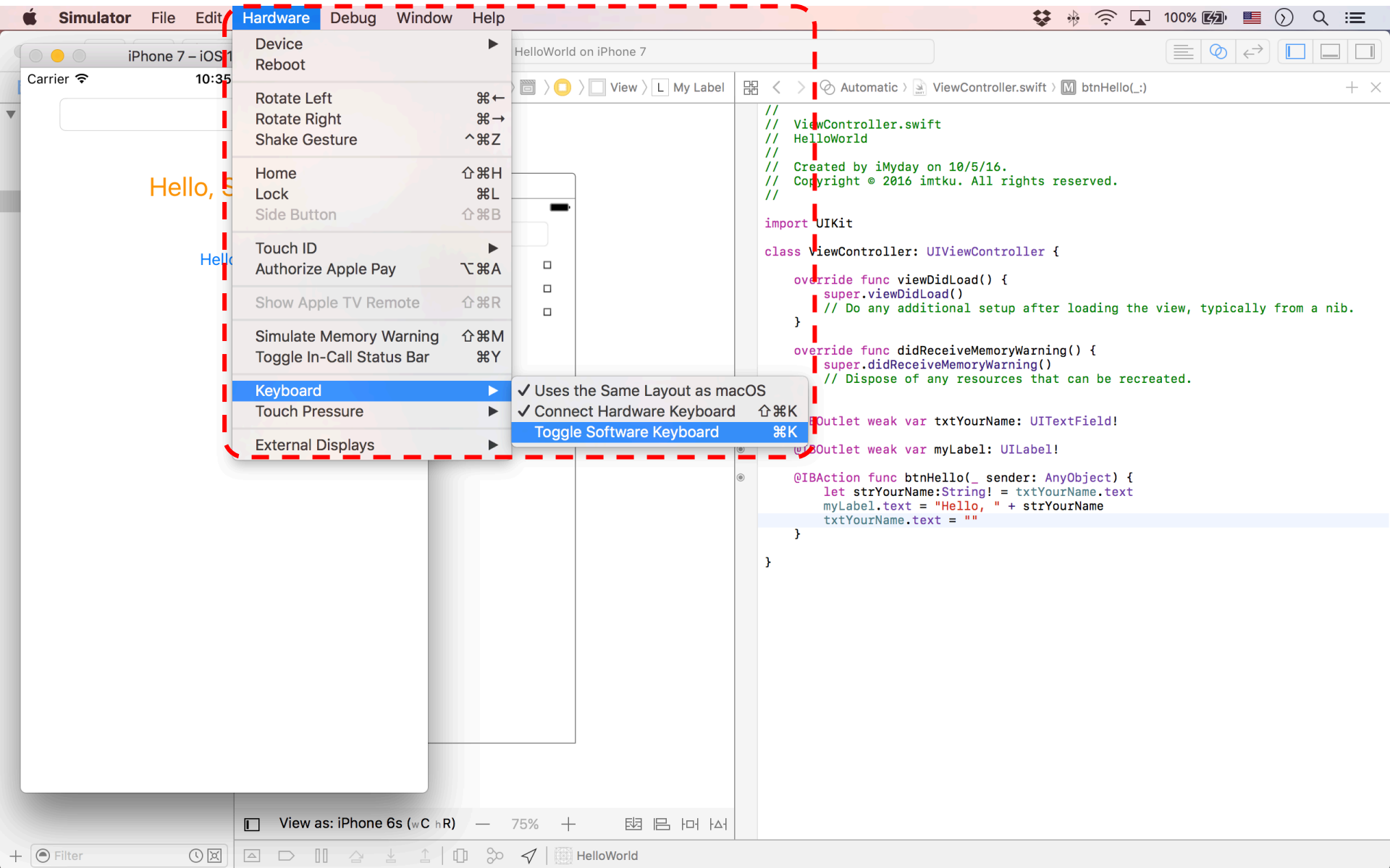
    @IBOutlet weak var txtYourName: UITextField!

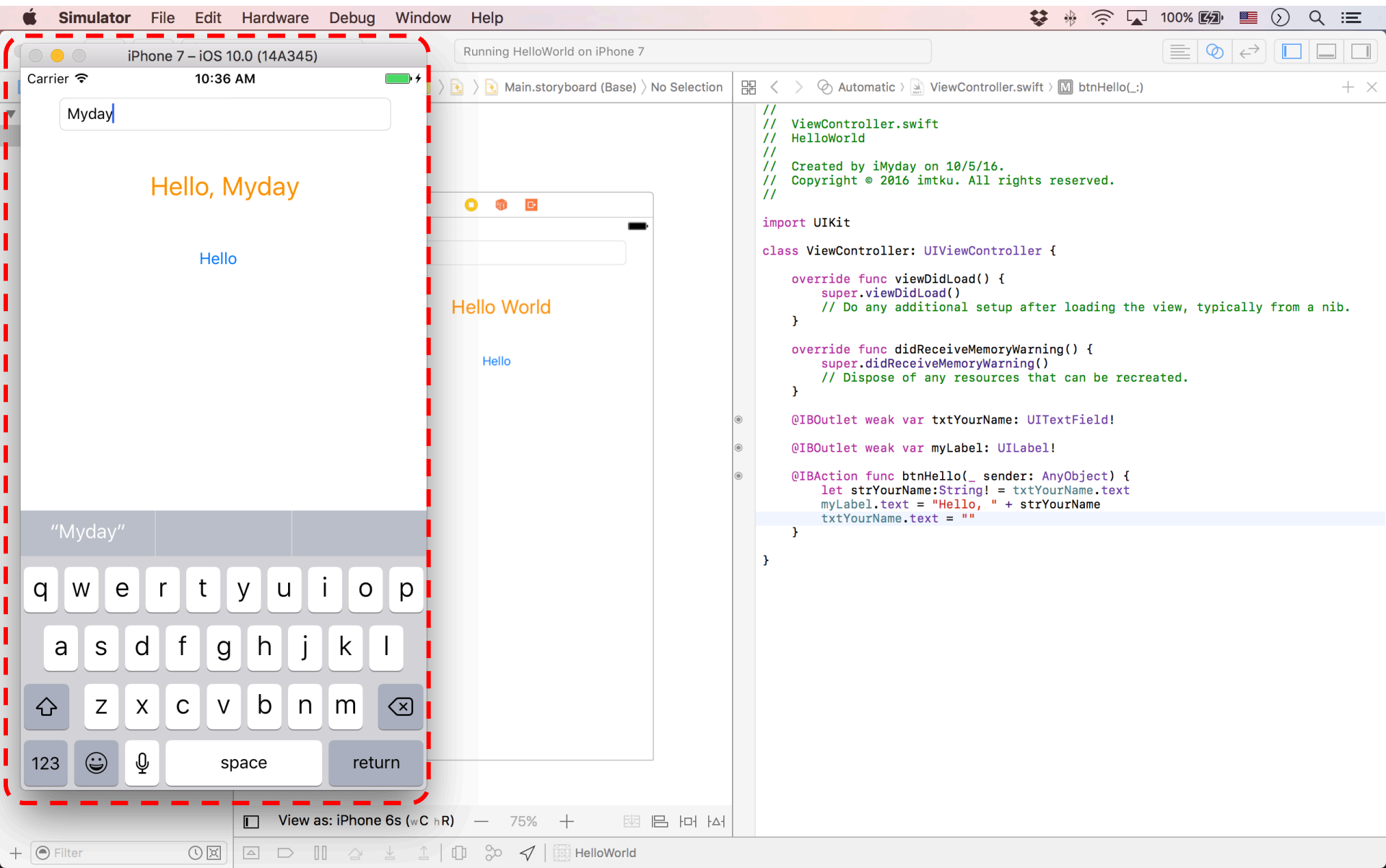
    @IBOutlet weak var myLabel: UILabel!

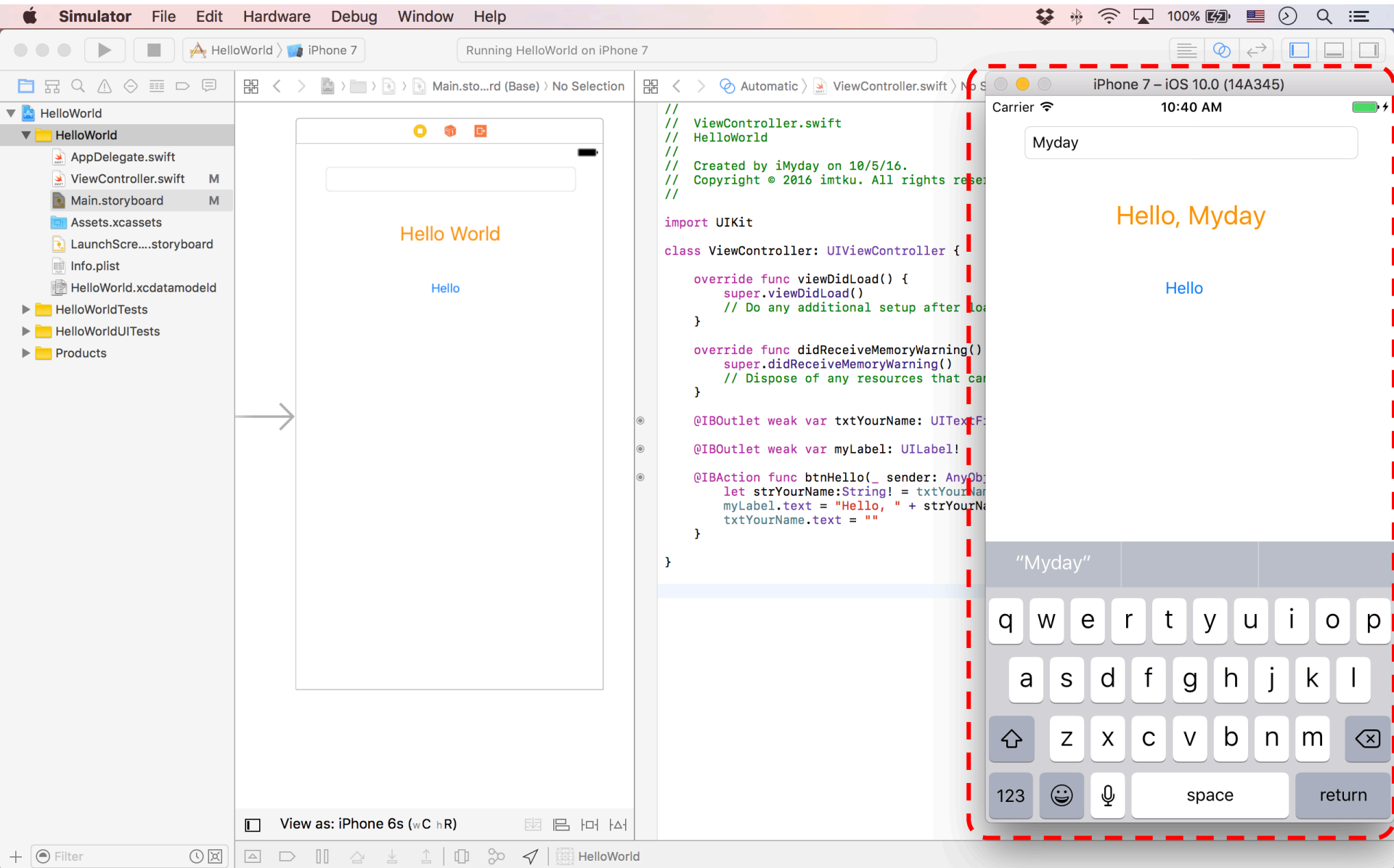
    @IBAction func btnHello(_ sender: AnyObject) {
        let strYourName:String! = txtYourName.text
        myLabel.text = "Hello, " + strYourName
        txtYourName.text = ""
    }
}
```











ViewController.swift

The screenshot shows the Xcode IDE with two panes open. The left pane shows the project structure for 'HelloWorld' on an iPhone 7. The right pane shows the Swift source code for 'ViewController.swift' and its Interface Builder counterpart.

```
//
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically
        // from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var txtYourName: UITextField!

    @IBOutlet weak var myLabel: UILabel!

    @IBAction func btnHello(_ sender: AnyObject) {
        let strYourName:String! = txtYourName.text
        myLabel.text = "Hello, " + strYourName
        txtYourName.text = ""
    }

}
```

```
//
// ViewController.swift
// HelloWorld
//
// Created by iMyday on 10/5/16.
// Copyright © 2016 imtku. All rights reserved.
//

import UIKit

internal class ViewController : UIViewController {

    override internal func viewDidLoad()

    override internal func didReceiveMemoryWarning()

    @IBOutlet weak internal var txtYourName: UITextField!

    @IBOutlet weak internal var myLabel: UILabel!

    @IBAction internal func btnHello(_ sender: AnyObject)

}
```

ViewController.swift

```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
    @IBOutlet weak var txtYourName: UITextField!  
  
    @IBOutlet weak var myLabel: UILabel!  
  
    @IBAction func btnHello(_ sender: AnyObject) {  
        let strYourName:String! = txtYourName.text  
        myLabel.text = "Hello, " + strYourName  
        txtYourName.text = ""  
    }  
  
}
```

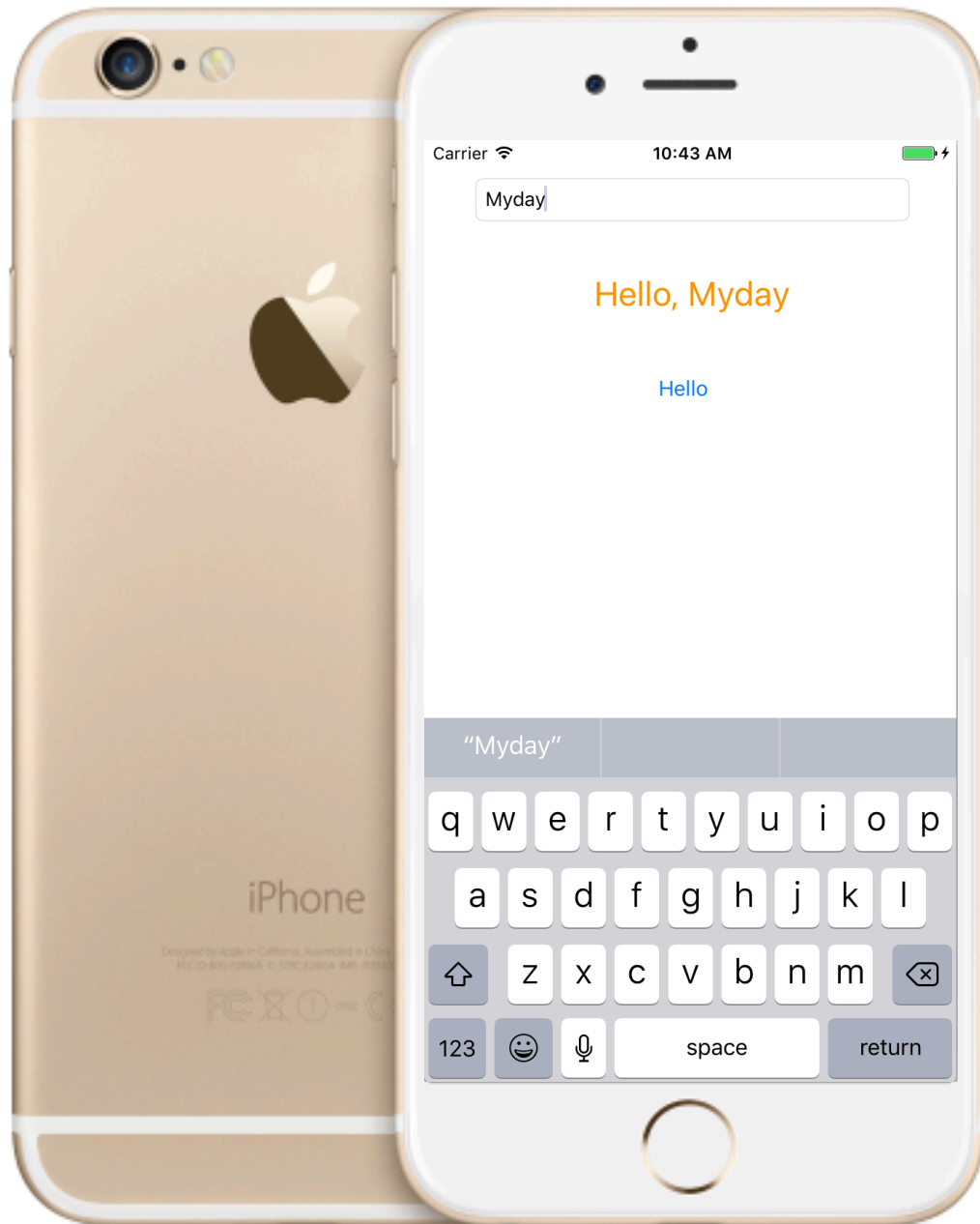
ViewController.swift

```
@IBOutlet weak var txtYourName: UITextField!  
  
@IBOutlet weak var myLabel: UILabel!  
  
@IBAction func btnHello(_ sender: AnyObject) {  
    let strYourName:String! = txtYourName.text  
    myLabel.text = "Hello, " + strYourName  
    txtYourName.text = ""  
}
```

ViewController.swift (Interface)

```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by iMyday on 10/5/16.  
// Copyright © 2016 imtku. All rights reserved.  
//  
  
import UIKit  
  
internal class ViewController : UIViewController {  
  
    override internal func viewDidLoad()  
  
    override internal func didReceiveMemoryWarning()  
  
    @IBOutlet weak internal var txtYourName: UITextField!  
  
    @IBOutlet weak internal var myLabel: UILabel!  
  
    @IBAction internal func btnHello(_ sender: AnyObject)  
}
```



Summary

- **Developing iPhone / iPad Native Apps with Swift 4 (Xcode 9)**
 - Mac OS X 10.8, 10.9, 10.10, 10.11, 10.12, 10.13
 - Xcode 6, Xcode 7, Xcode 8, Xcode 9
 - iOS 8, iOS 9, iOS 10, iOS 11
- **Building Your First iOS App with Xcode 9**



Xcode 8



Swift 3



Xcode 9



Swift 4

References

- Apple Developers, <https://developer.apple.com/>
- Start Developing iOS Apps Today,
 - <https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/>
 - <https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/RoadMapiOS.pdf>
- Rory Lewis and Laurence Moroney, **iPhone and iPad Apps for Absolute Beginners**, 4th Edition, Apress, 2013
- The Swift Programming Language,
 - <https://itunes.apple.com/us/book/the-swift-programming-language/id881256329>
 - https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/
- Apple - WWDC 2014
 - <https://www.youtube.com/watch?v=w87fOAG8fjk>
- Apple - WWDC 2015
 - https://www.youtube.com/watch?v=_p8AsQhaVKI
- Apple - WWDC 2016 Keynote
 - https://www.youtube.com/watch?v=n5jXg_NNiCA
- What's New in Swift 4 - Apple WWDC 2017
 - <https://www.youtube.com/watch?v=3y42Qg6MTvk>