Mobile Application Development in iOS

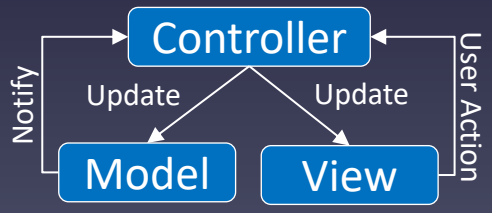
School of EECS

Washington State University

Instructor: Larry Holder

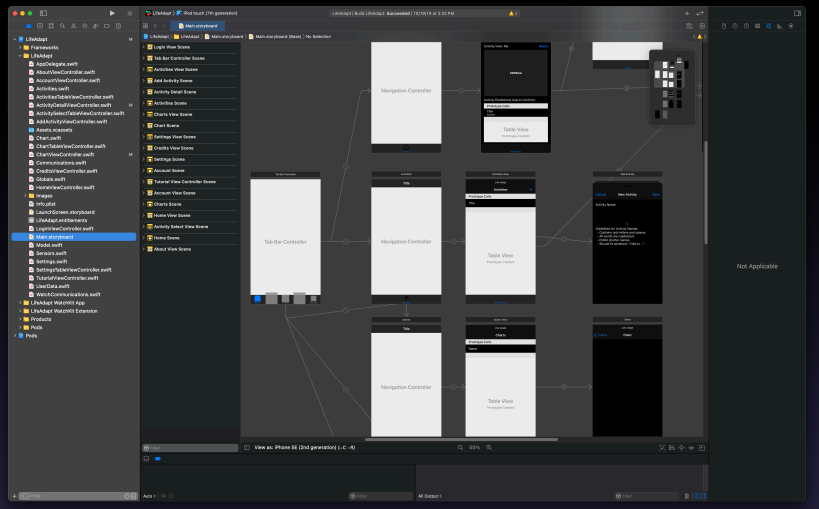
Course Overview

- Overview of iOS
- Language: Swift
- Development environment: Xcode
- Lifecycle: Design, implement, test, deploy
- Model-View-Controller (MVC) paradigm



Course Topics

- Swift
- UI design
- Navigation and segues
- Tables
- Settings
- Alerts and notifications
- Gestures

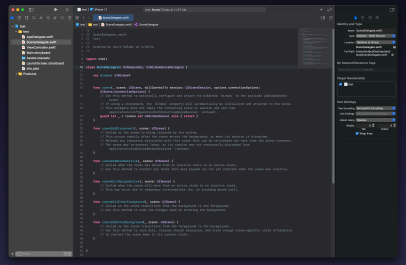


Course Topics (cont.)

- Sensors
- Communications
- Data storage
- Multimedia
- Graphics and animation
- Apple Watch

Course Outcomes

- Proficient with iOS development environment
- Design, implement, test and deploy iOS app
- Understand and use the main iOS frameworks



Course Details

- Syllabus on Blackboard Learn https://learn.wsu.edu
- Prerequisites
  - Advanced Data Structures
  - Object-oriented design
- You will need access to the latest MacOS and Xcode
  - Currently, MacOS 11.1 (Big Sur), Xcode 12.3 (iOS 14.3)

Course Details (cont.)

Instructor

- Larry Holder
- Email: holder@wsu.edu
- Office hours: By appointment

Teaching Assistants

- Amir Gholami
  - Email: amir.gholami@wsu.edu
  - Office hours: By appointment
  - Grading: Homeworks 1, 3, 5, 7, 9, 11
- Mariana Rodriguez Rojas
  - Email: m.rodriguezrojas@wsu.edu
  - Office hours: By appointment
  - Grading: Homeworks 2, 4, 6, 8, 10, 12

Course Details (cont.)

- Lectures
  - Live lectures on Mondays (recorded)
  - Other weekly lectures pre-recorded
- Attendance
  - Attendance is not required
  - Students are responsible for all lecture content
- Textbook and Materials
  - No textbook
  - Lecture notes and all other materials on Blackboard Learn

Course Details (cont.)

- 12 Homeworks (80%)
  - Programming assignments due every Wednesday
- Final Project (20%)
  - Approved proposal required
- Assignments submitted via Blackboard Learn
- Late Policy: Assignments submitted after the deadline will receive a zero

Course Details (cont.)

Grading (no curve)

Letter grade	Percentage
A	93–100%
A–	90–92%
B+	87–89%
B	83–86%
B–	80–82%
C+	77–79%

Letter grade	Percentage
C	73–76%
C-	70–72%
D+	67–69%
D	63–66%
D-	60–62%
F	0–59%

Course Details (cont.)

Coding Rules

- Students are expected to adhere to the following rules for code submitted in this course. Violation of these rules will be considered plagiarism and result in a zero on the assignment according to the university's Academic Integrity policy.
  - No portion of your code can be a direct copy of another student's code. Direct copies include making simple changes to the code.
  - You must not allow your code to be copied by another student.
  - You may use small segments of code from publicly-available online sources as long as you cite the source in your code.
- Questions? Contact instructor

Course Details (cont.)

- Mobile App Development Resources
  - Apple Developer
  - Android Developer
  - Windows Developer
  - Swift Programming Language
- Swift Style Guides
  - Swift API Design Guidelines
  - The Official raywenderlich.com Swift Style Guide
  - Google's Swift Style Guide
- Other resources
  - Stanford Developing Apps for iOS Course
  - Ray Wenderlich Tutorials

Course Details (cont.)

- See syllabus for more details on...
  - Academic integrity
  - Students with disabilities
  - COVID-19 policy
  - Accommodation for religious observances or activities
  - Safety and emergency information

Course Schedule

Week	Dates	Topic	Assignments Due	Notes
1	Jan 18-22	Introduction		No class Jan 18
2	Jan 25-29	Swift	Homework 1 (Jan 27)	
3	Feb 1-5	UI Design	Homework 2 (Feb 3)	
4	Feb 8-12	Navigation and Segues	Homework 3 (Feb 10)	
5	Feb 15-19	Tables	Homework 4 (Feb 17)	No class Feb 15
6	Feb 22-26	Settings	Homework 5 (Feb 24)	
7	Mar 1-5	Alerts and Notifications	Homework 6 (Mar 3)	
8	Mar 8-12	Gestures	Homework 7 (Mar 10)	
9	Mar 15-19	Sensors	Homework 8 (Mar 17)	No class Mar 17
10	Mar 22-26	Communications	Homework 9 (Mar 24)	
11	Mar 29 - Apr 2	Data Storage	Homework 10 (Mar 31)	
12	Apr 5-9	Multimedia	Project Proposal (Apr 7)	
13	Apr 12-16	Graphics and Animation	Homework 11 (Apr 14)	
14	Apr 19-23	Graphics and Animation	Homework 12 (Apr 21)	
15	Apr 26-30	Apple Watch		
16	May 3-7	Finals Week	Final Project (May 5)	No class

Mobile App Development: Platforms

- iOS (Swift)

- Android (Java)

- Windows (C#)

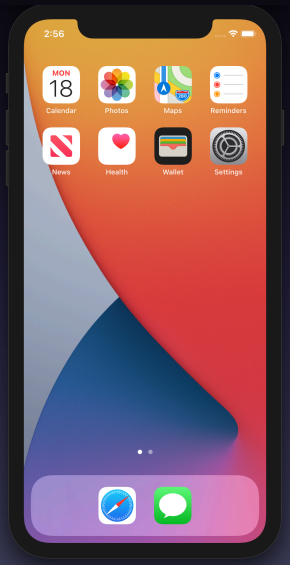
  - Xamarin (C# → iOS/Android/Windows)

- Tizen (C#)

iOS Devices

iPad

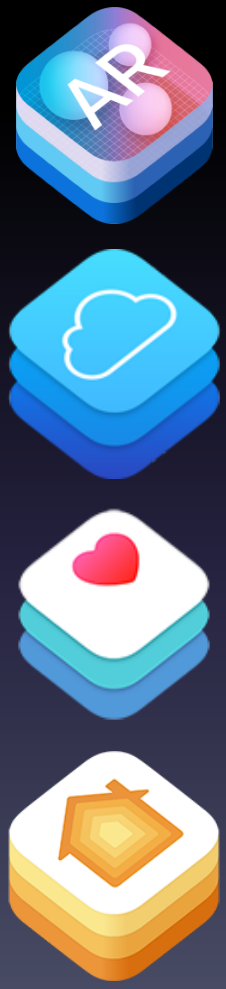
iPhone



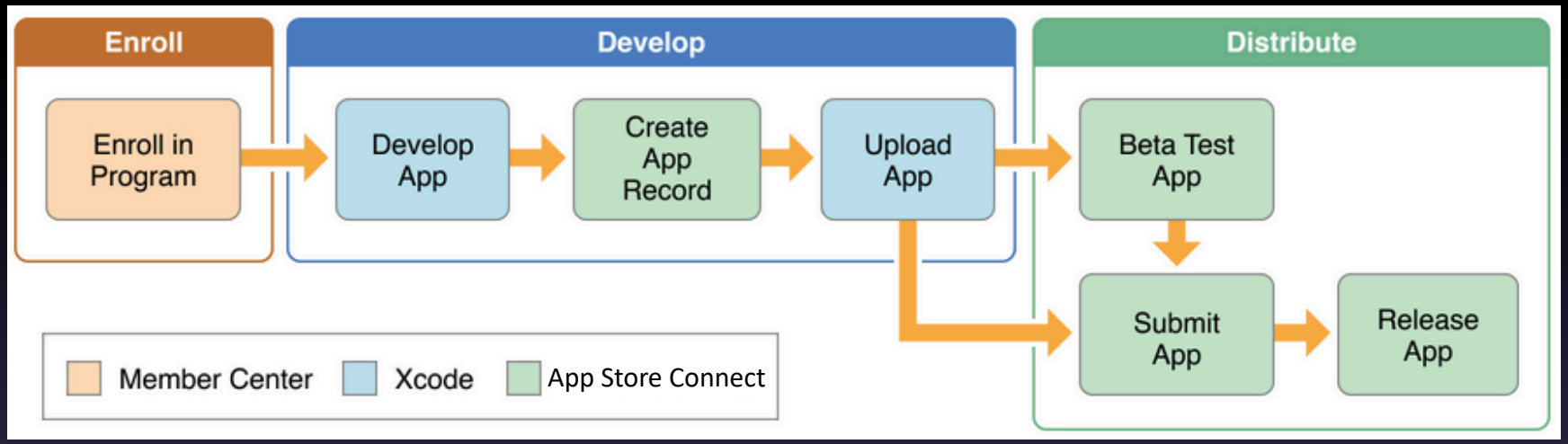
Apple Watch

iOS Frameworks

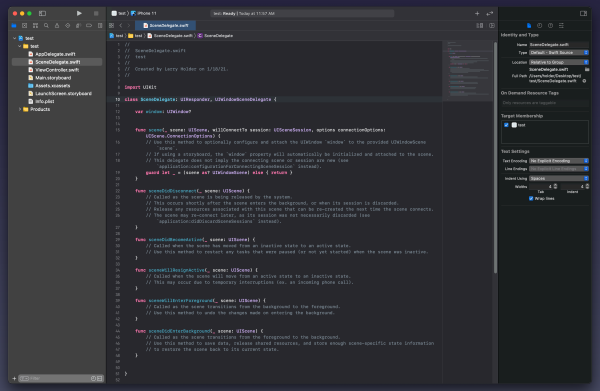
- ARKit
- CloudKit
- HealthKit
- HomeKit
- GameKit
- MapKit
- SceneKit
- MusicKit
- WebKit
- UIKit
- CoreData
- CoreLocation
- CoreMotion
- CoreML
- WatchOS/WatchKit
- Over 50 more...



Development Environment: Xcode



Xcode



UI Design in Xcode

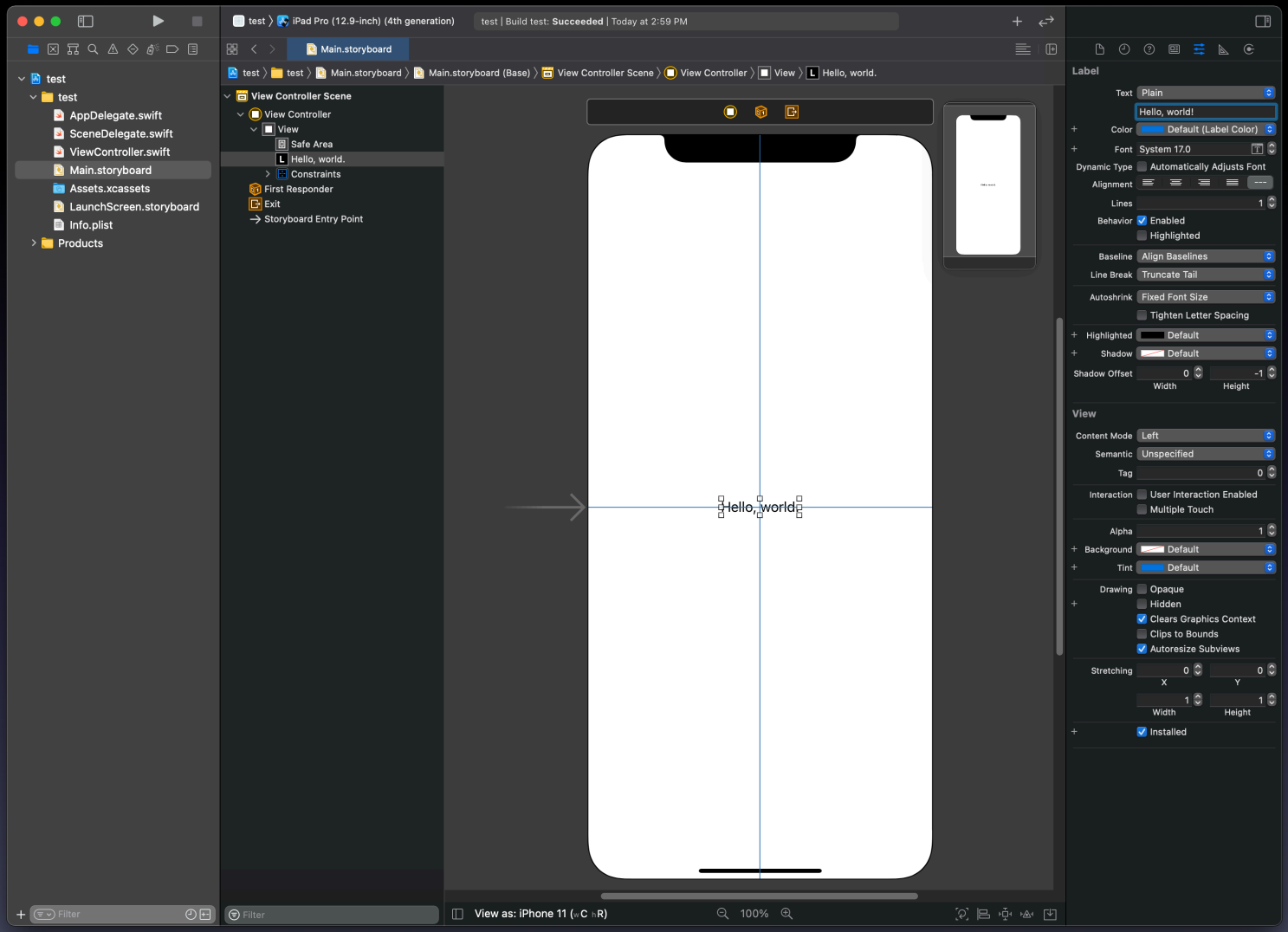
Storyboard

- For all iOS versions
- Mostly visual: drag-and-drop with some coding
- Some disconnect between Storyboard XML and Swift code
- Stable
- Well-documented

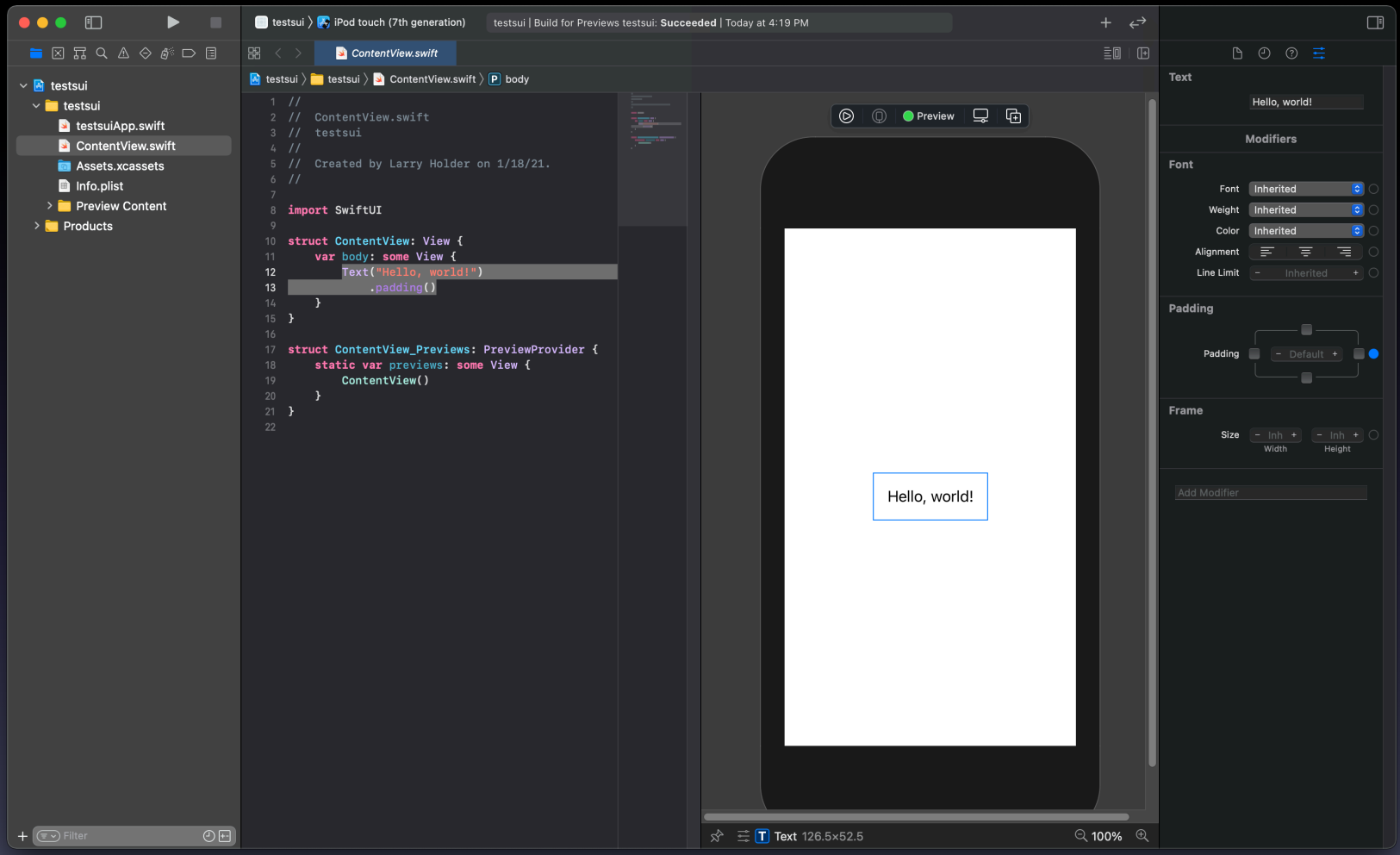
SwiftUI

- For iOS 13 or greater
- Mostly code: but some drag-and-drop and real-time visual
- UI design and code all in Swift
- New (and better?)
- Less documentation

Storyboard


test
iPad Pro (12.9-inch) (4th generation)
test | Build test: Succeeded | Today at 2:59 PM
Main.storyboard
test
test
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
View
Safe Area
Hello, world.
Constraints
First Responder
Exit
Storyboard Entry Point
Hello, world
View as: iPhone 11
100%
Label
Text
Plain
Hello, world!
Color
Default (Label Color)
Font
System 17.0
Dynamic Type
Automatically Adjusts Font
Alignment
Lines
Behavior
Enabled
Highlighted
Baseline
Align Baselines
Line Break
Truncate Tail
Autoshrink
Fixed Font Size
Tighten Letter Spacing
Highlighted
Default
Shadow
Default
Shadow Offset
Width
Height
View
Content Mode
Left
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Default
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clips to Bounds
Autoresize Subviews
Stretching
X
Y
Width
Height
Installed

SwiftUI


testsui
iPod touch (7th generation)
testsui | Build for Previews testsui: Succeeded | Today at 4:19 PM
ContentView.swift
testsui
testsui
testsuiApp.swift
ContentView.swift
Assets.xcassets
Info.plist
Preview Content
Products
testsui
testsui
ContentView.swift
body
// ContentView.swift // testsui // // Created by Larry Holder on 1/18/21. //
import SwiftUI
struct ContentView: View {
var body: some View {
Text("Hello, world!")
.padding()
}
}
struct ContentView_Previews: PreviewProvider {
static var previews: some View {
ContentView()
}
}
Preview
Hello, world!
Text 126.5×52.5
100%
Text
Hello, world!
Modifiers
Font
Font
Inherited
Weight
Inherited
Color
Inherited
Alignment
Line Limit
Inherited
Padding
Padding
Default
Frame
Size
Width
Height
Add Modifier

First App

Mobile Application Development in iOS

School of EECS

Washington State University

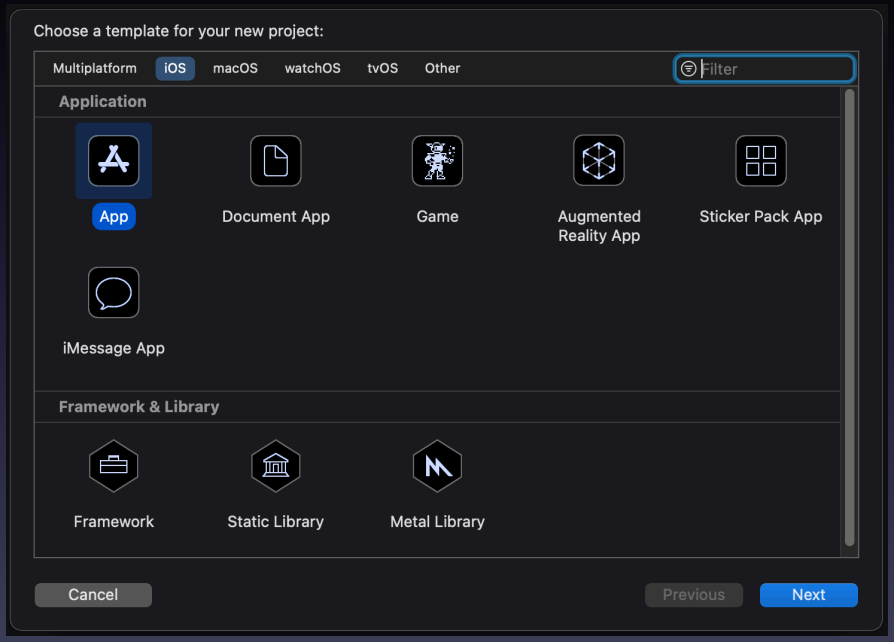
Instructor: Larry Holder

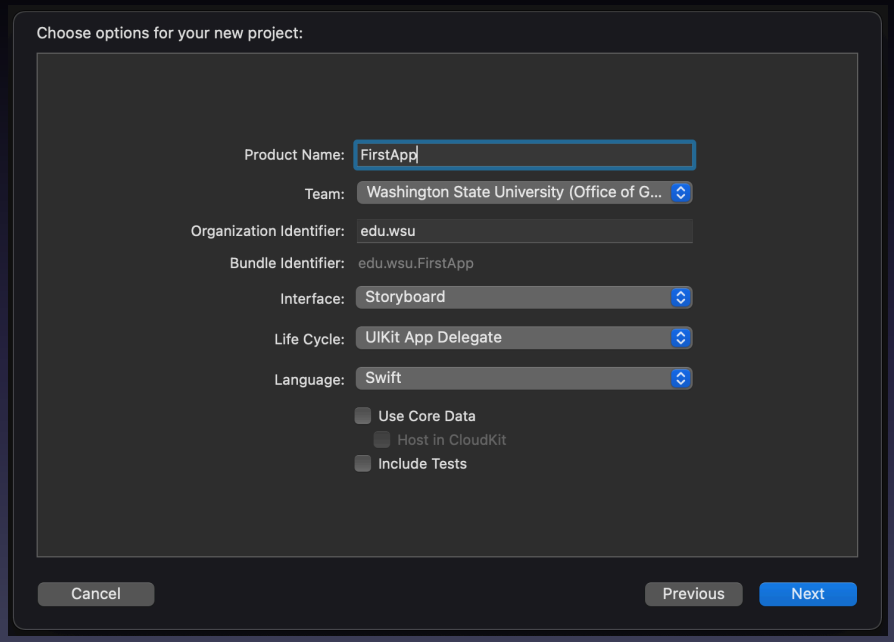
Outline

- Xcode 12 projects
- Anatomy of an iOS app
  - iOS 14 (iPhone 6s or higher)
- Storyboarding
- Object attributes
- Auto-layout constraints
- Outlets and Actions

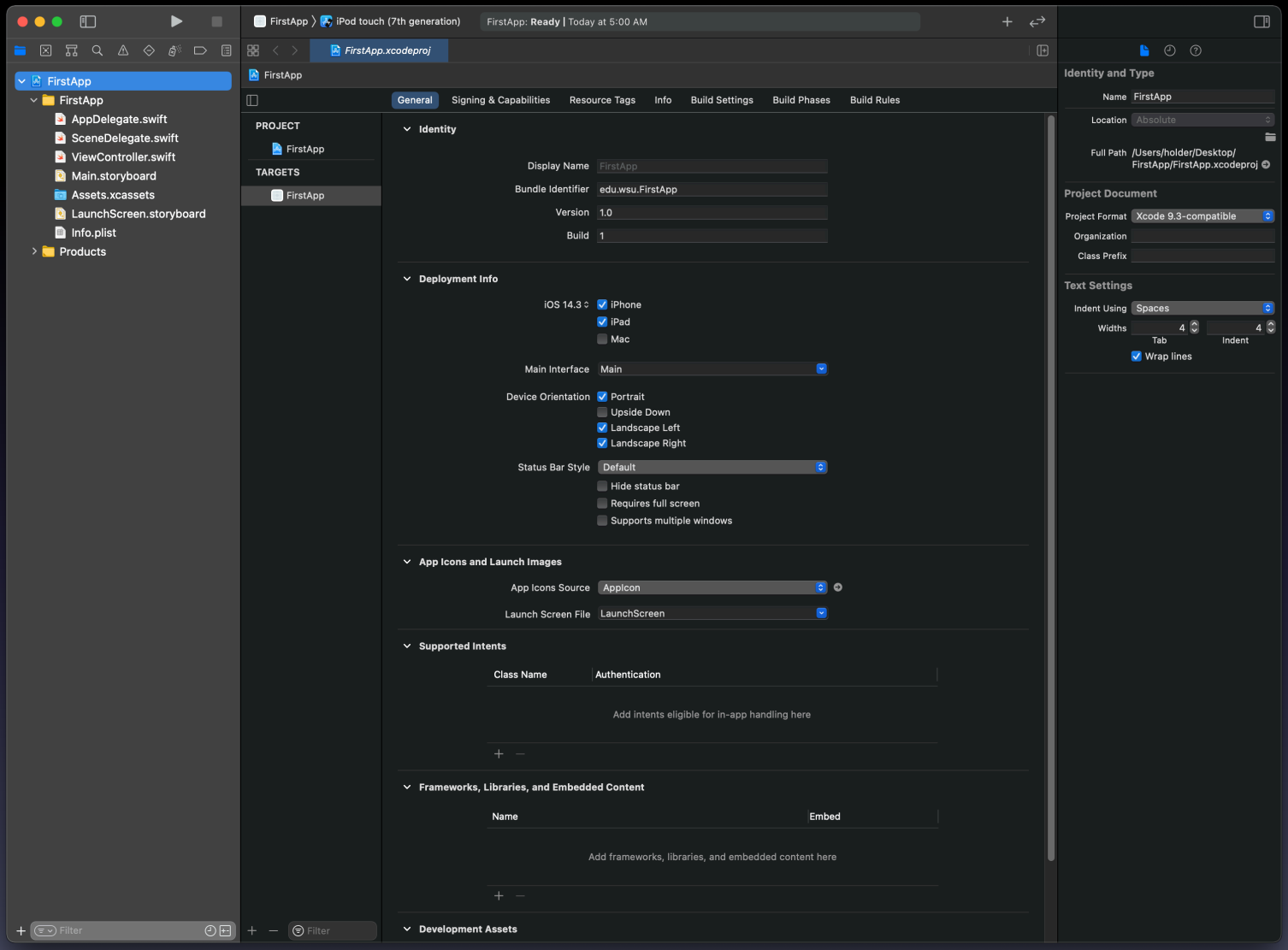
Xcode Projects

- File → New → Project

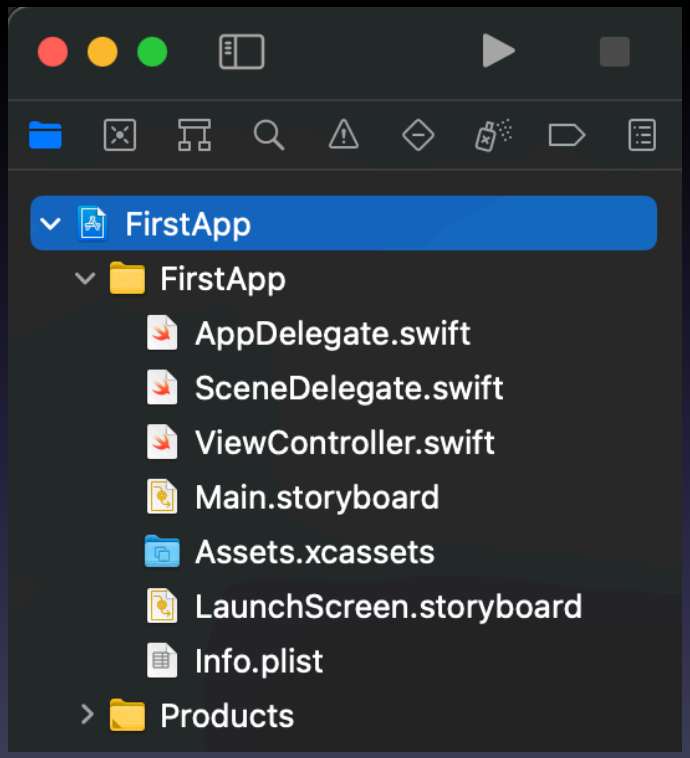




Xcode

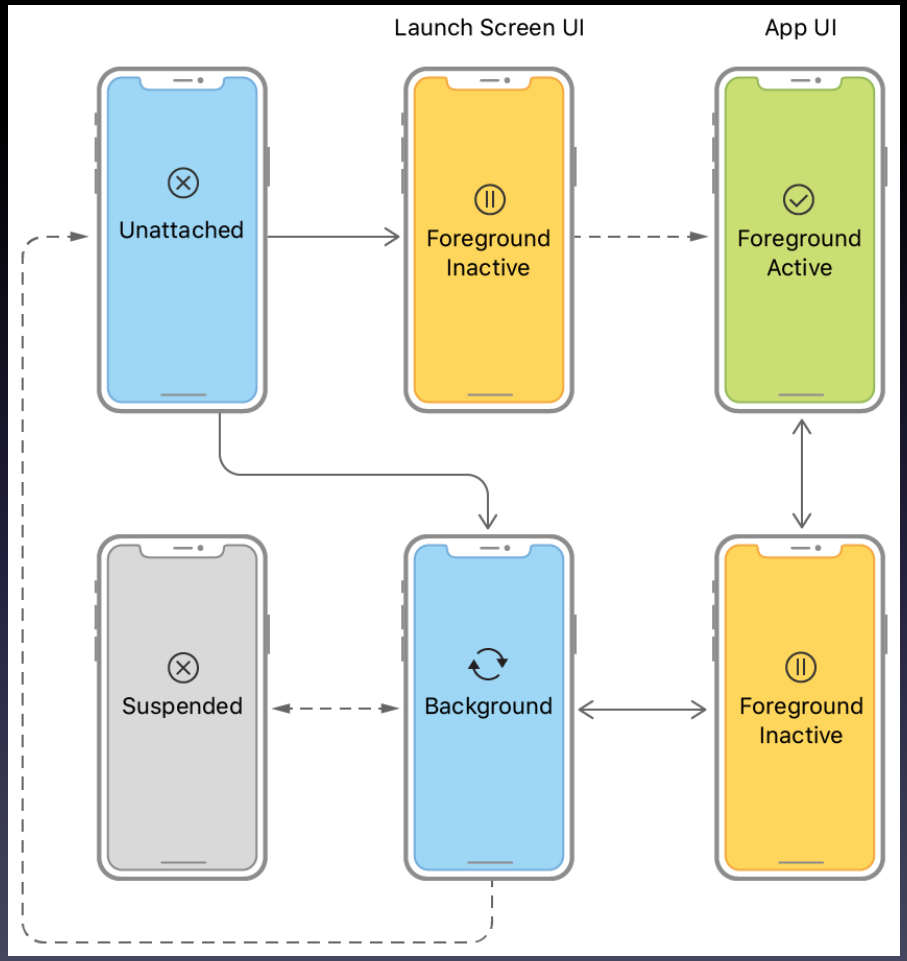

FirstApp
FirstApp
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
FirstApp 〉 iPod touch (7th generation)
FirstApp: Ready | Today at 5:00 AM
FirstApp.xcodeproj
FirstApp
General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
FirstApp
TARGETS
FirstApp
Identity
Display Name FirstApp
Bundle Identifier edu.wsu.FirstApp
Version 1.0
Build 1
Deployment Info
iOS 14.3
iPhone
iPad
Mac
Main Interface Main
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style Default
Hide status bar
Requires full screen
Supports multiple windows
App Icons and Launch Images
App Icons Source AppIcon
Launch Screen File LaunchScreen
Supported Intents
Class Name Authentication
Add intents eligible for in-app handling here
Frameworks, Libraries, and Embedded Content
Name Embed
Add frameworks, libraries, and embedded content here
Development Assets
Identity and Type
Name FirstApp
Location Absolute
Full Path /Users/holder/Desktop/FirstApp/FirstApp.xcodeproj
Project Document
Project Format Xcode 9.3-compatible
Organization
Class Prefix
Text Settings
Indent Using Spaces
Widths 4 4
Tab Indent
Wrap lines

Anatomy of an iOS App

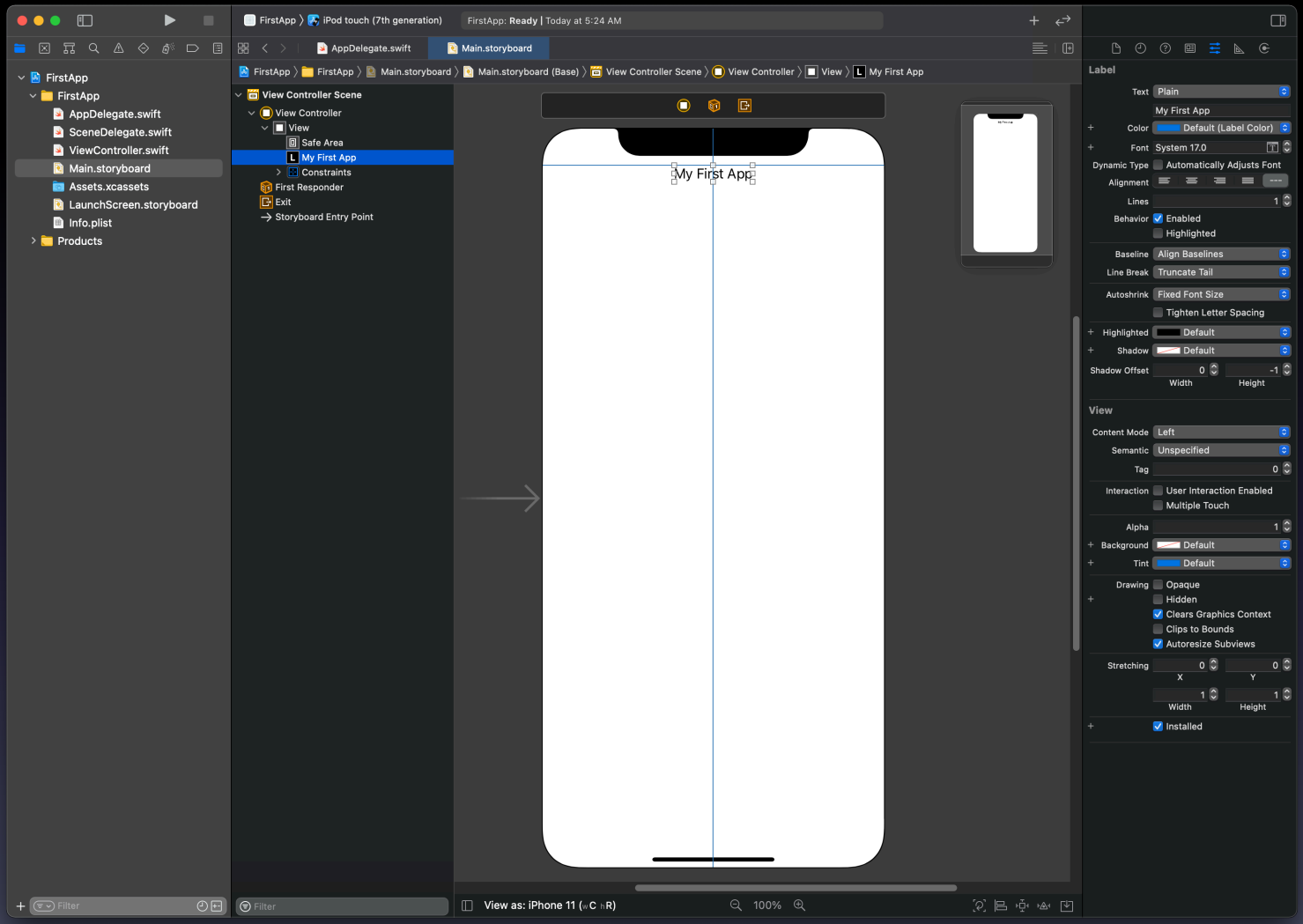


- AppDelegate
  - Entry point
- SceneDelegate
  - Lifecycle management
- ViewController
  - Controls a view
- Launch & Main storyboards
- Assets.xcassets
  - Icons, images, videos
- Info.plist
  - App properties

App Scene Lifecycle

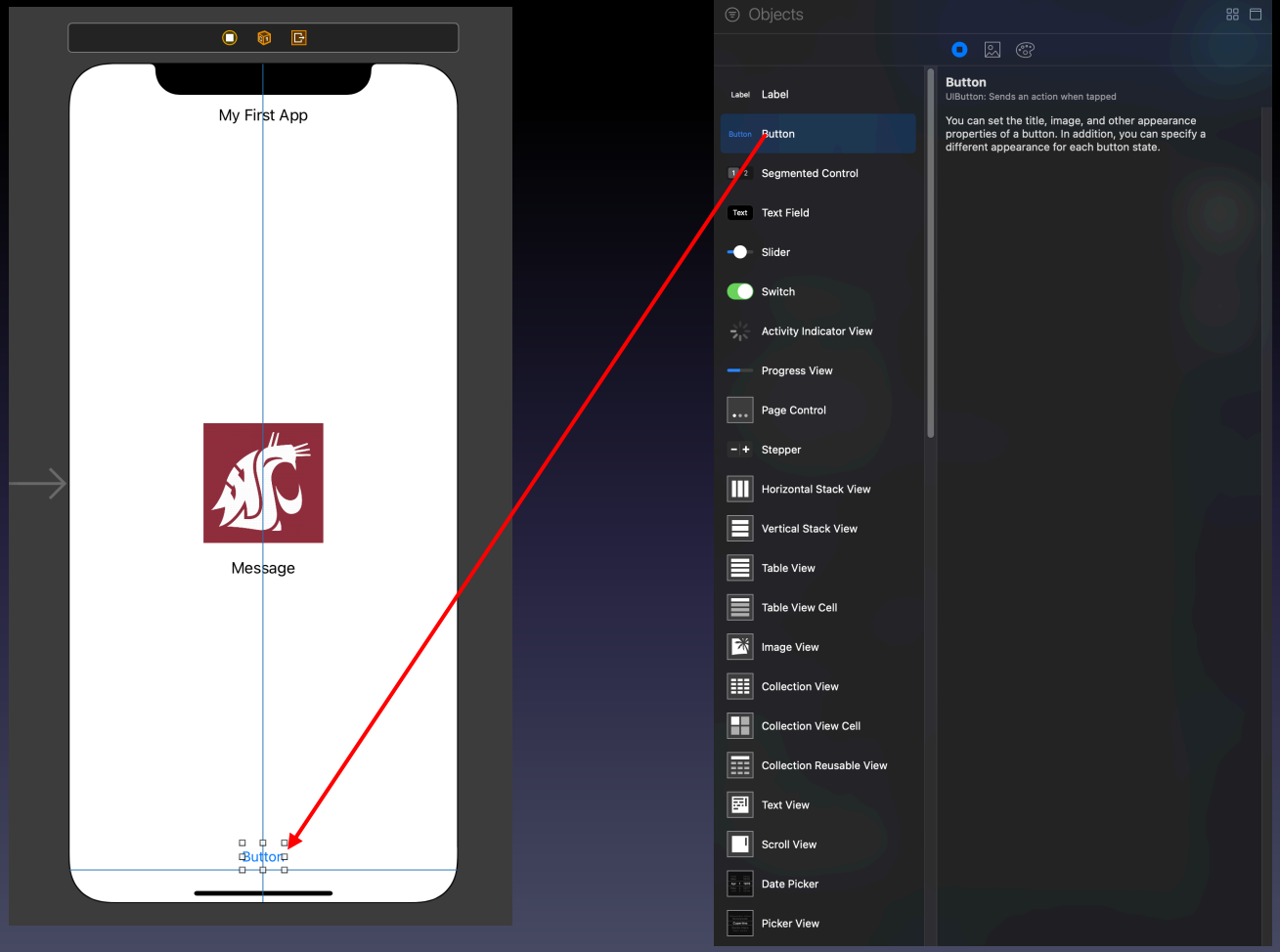


Storyboard


FirstApp
iPod touch (7th generation)
FirstApp: Ready | Today at 5:24 AM
AppDelegate.swift
Main.storyboard
FirstApp
SceneDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
View
Safe Area
My First App
Constraints
First Responder
Exit
Storyboard Entry Point
My First App
Label
Text
Plain
My First App
Color
Default (Label Color)
Font
System 17.0
Dynamic Type
Automatically Adjusts Font
Alignment
Lines
Behavior
Enabled
Highlighted
Baseline
Align Baselines
Line Break
Truncate Tail
Autoshrink
Fixed Font Size
Tighten Letter Spacing
Highlighted
Default
Shadow
Default
Shadow Offset
Width
Height
View
Content Mode
Left
Semantic
Unspecified
Tag
Interaction
User Interaction Enabled
Multiple Touch
Alpha
Background
Default
Tint
Default
Drawing
Opaque
Hidden
Clears Graphics Context
Clips to Bounds
Autoresize Subviews
Stretching
X
Y
Width
Height
Installed
View as: iPhone 11
100%

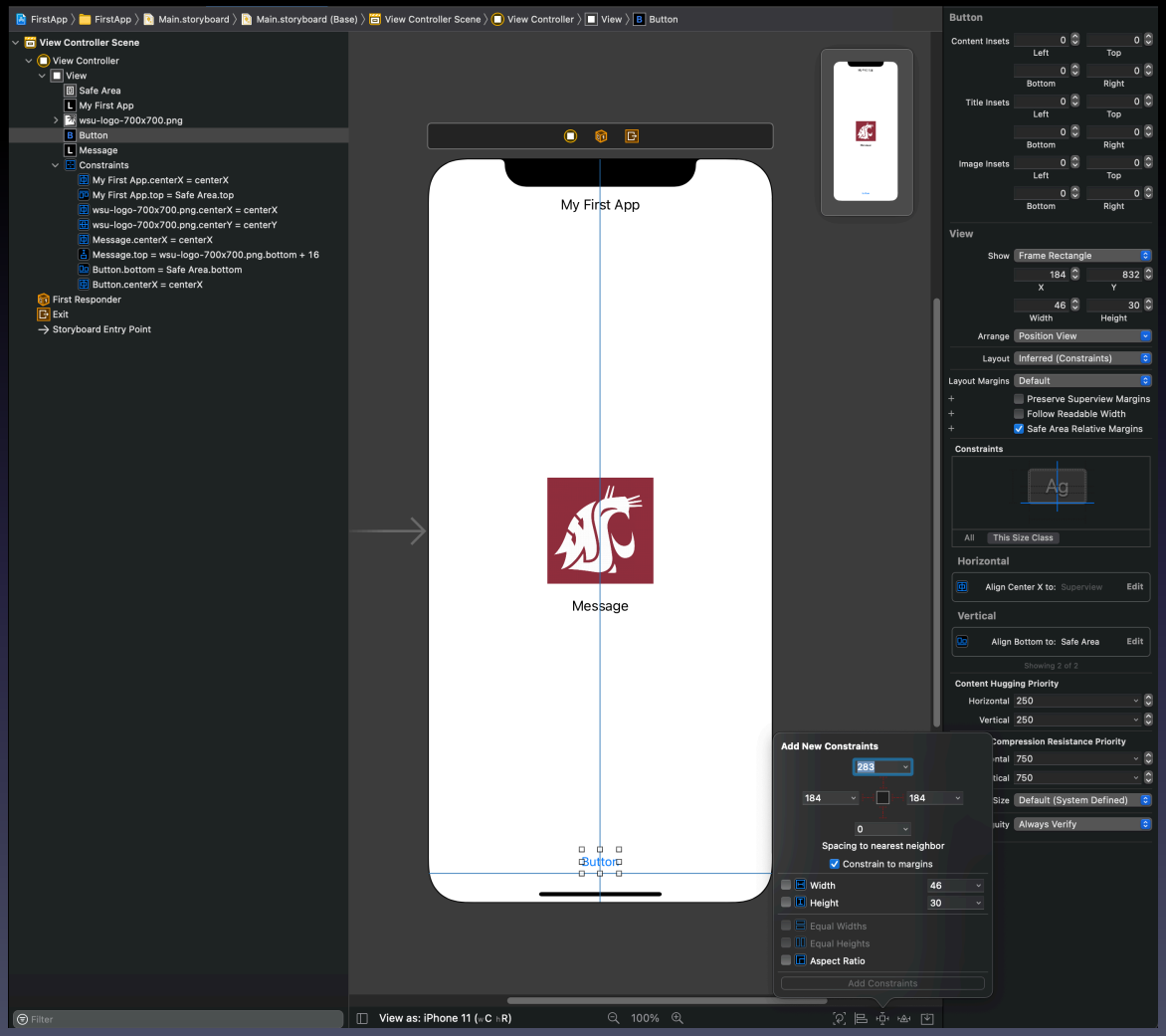
Storyboard: Adding Objects to View

1. Drag object into view.



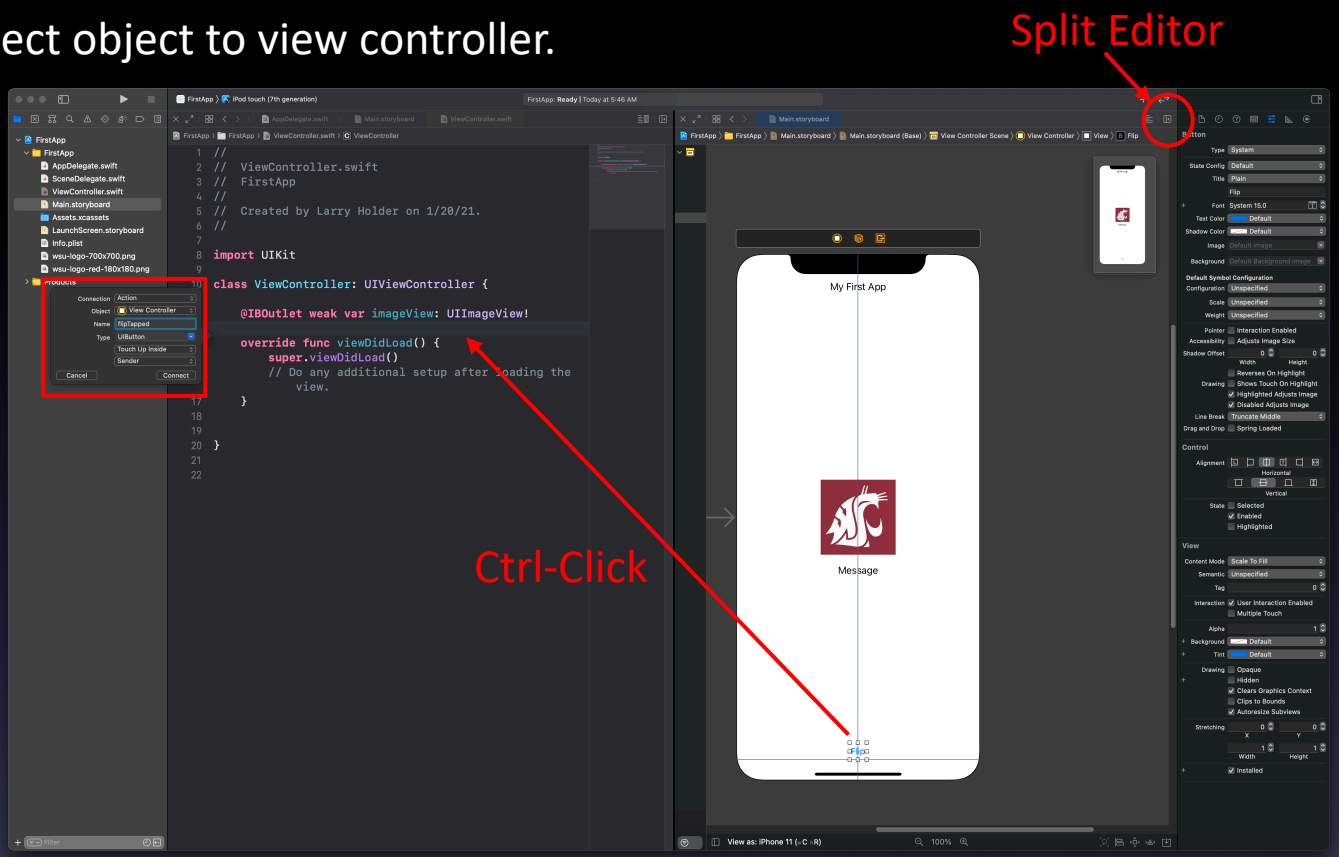
Storyboard: Adding Objects to View

2. Add layout contraints.



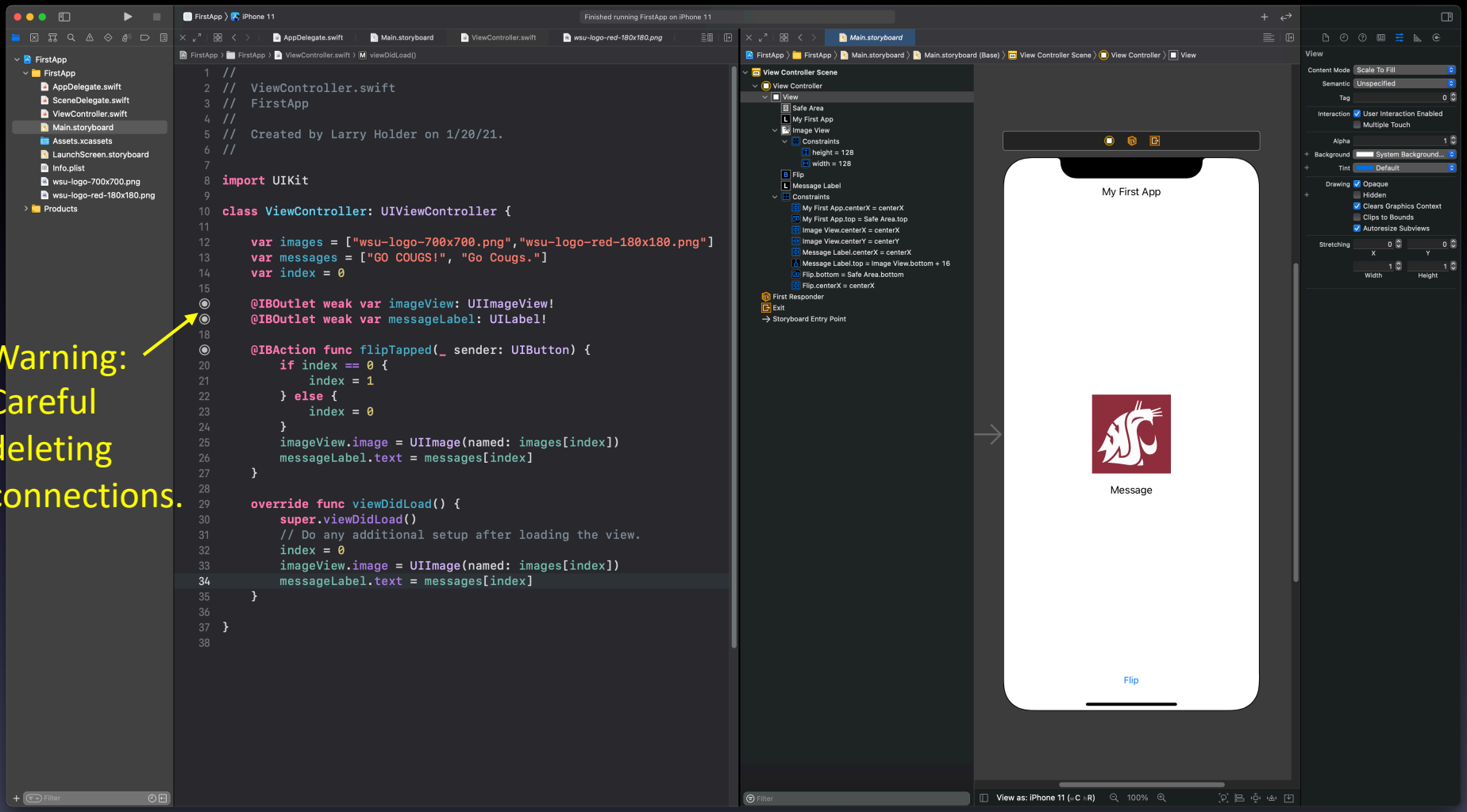
Storyboard: Adding Objects to View

3. Connect object to view controller.



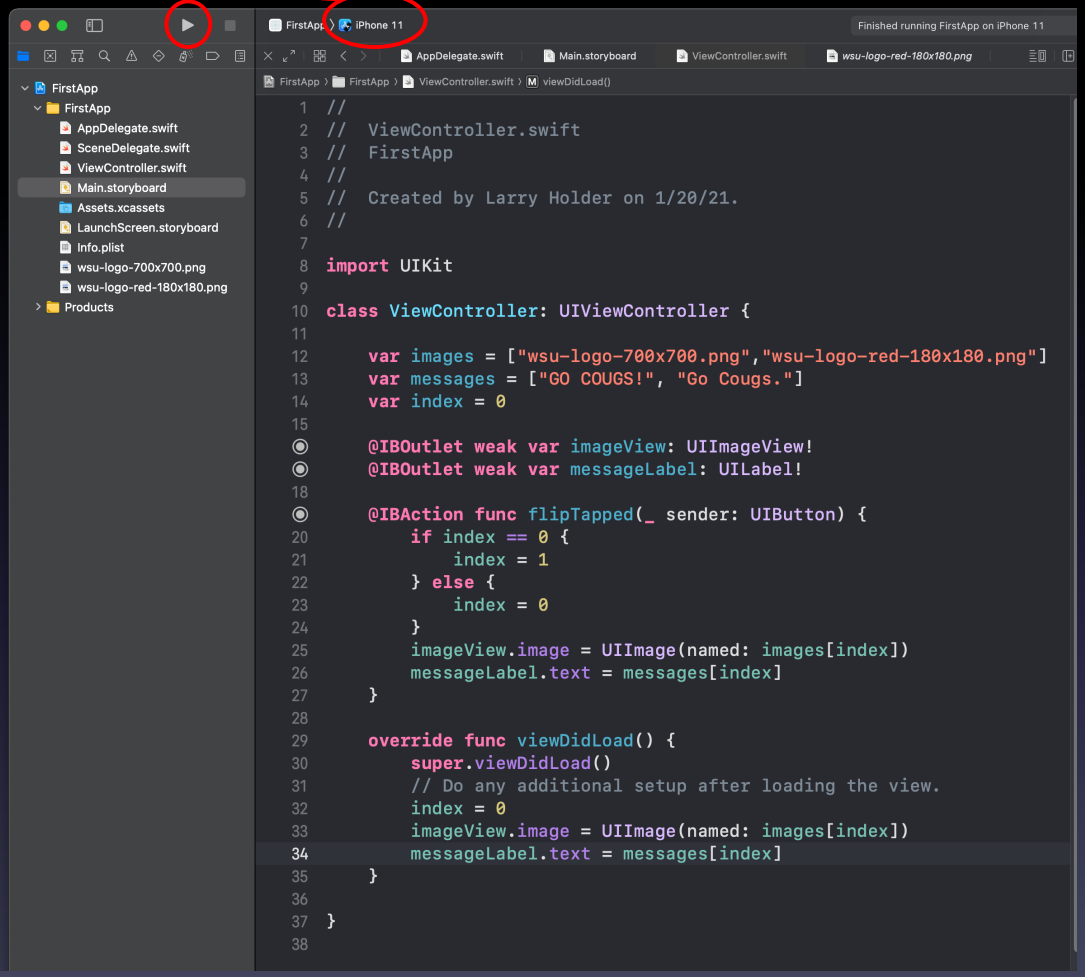
- Create IBOutlet to get/set properties of object
- Create IBAction to detect interaction with object

Storyboard: Adding Objects to View



Running App

Choose simulator



Resources

- developer.apple.com
- swift.org
- www.raywenderlich.com
- stackoverflow.com