lecture 1

MVC, Swift & Xcode

cs198-001 : fall 2017

today’s lecture

- logistics
- staff introduction
- more swift
- Xcode Overview
- MVC

logistic clarifications

enrollments

- everyone *should* be enrolled in cs198-001
  - let us know after class if you haven't

attendance policy

- NP if you have 4+ unexcused absences (lecture and lab)
- excused absences for midterm conflicts only
  - make a private piazza post *before* lecture with the class name and time of your midterm

lab sections

- sign up for lab sections after lecture
  - 310 Soda - Chris
  - 320 Soda - Nithi
  - 210 Wheeler - Paige (for students new to practical programming)
- 7-8:30pm on Wednesdays

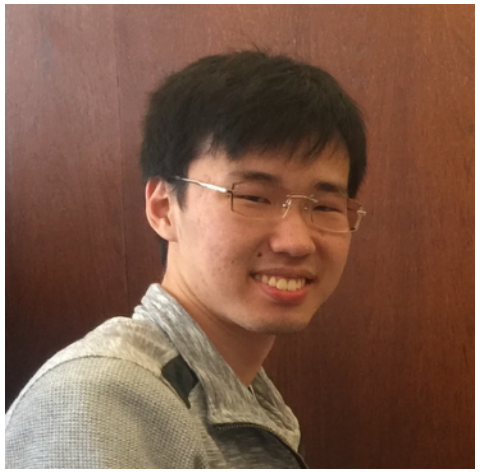
staff introductions

course staff

Paige Plander
instructor

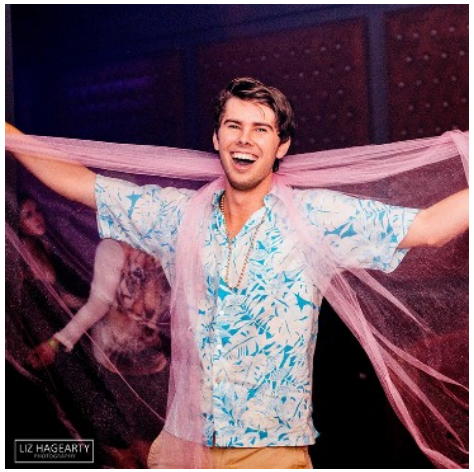
Chris Zielinski
instructor

Nithi Narayanan
head TA



Jim

Sarah



Jack

Aspen

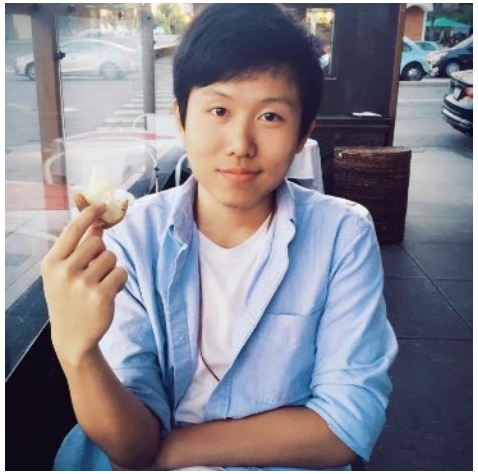
Victor

Daniel

Gokul

Marisa

Alex



Teddy

MVC

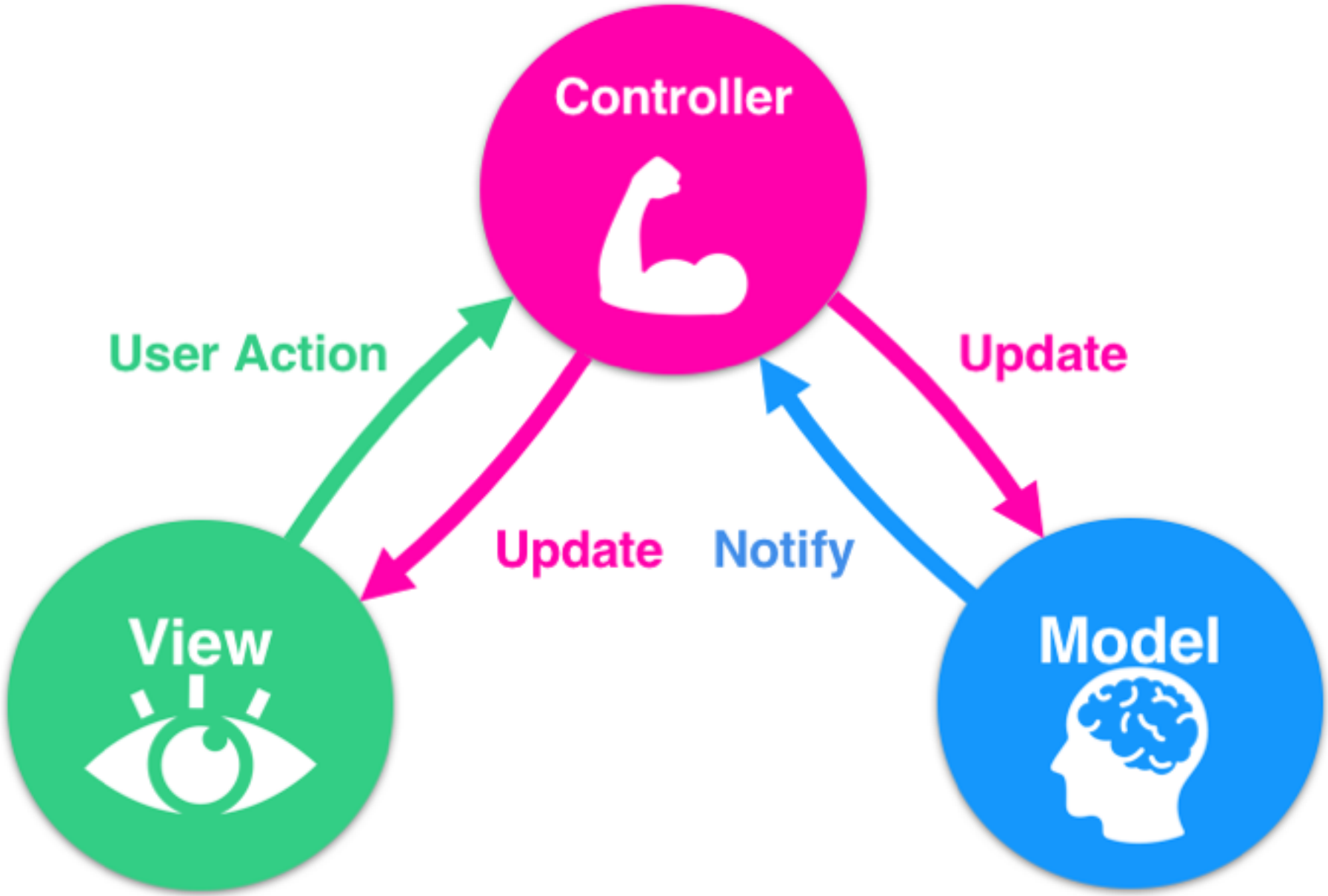
design patterns

- software design patterns are reusable solutions to common problems in software design
- **main idea:** assign objects in your application distinct roles using well-defined patterns and object relationships
  - There are many different types of patterns (not just basic MVC)!

model view controller (MVC)

assigns objects in your application one of the following distinct roles:

- **model** - encapsulates data and defines logic / computations
- **view** - what the users see and interact with
- **controller - intermediary between models and views**



Model View Controller

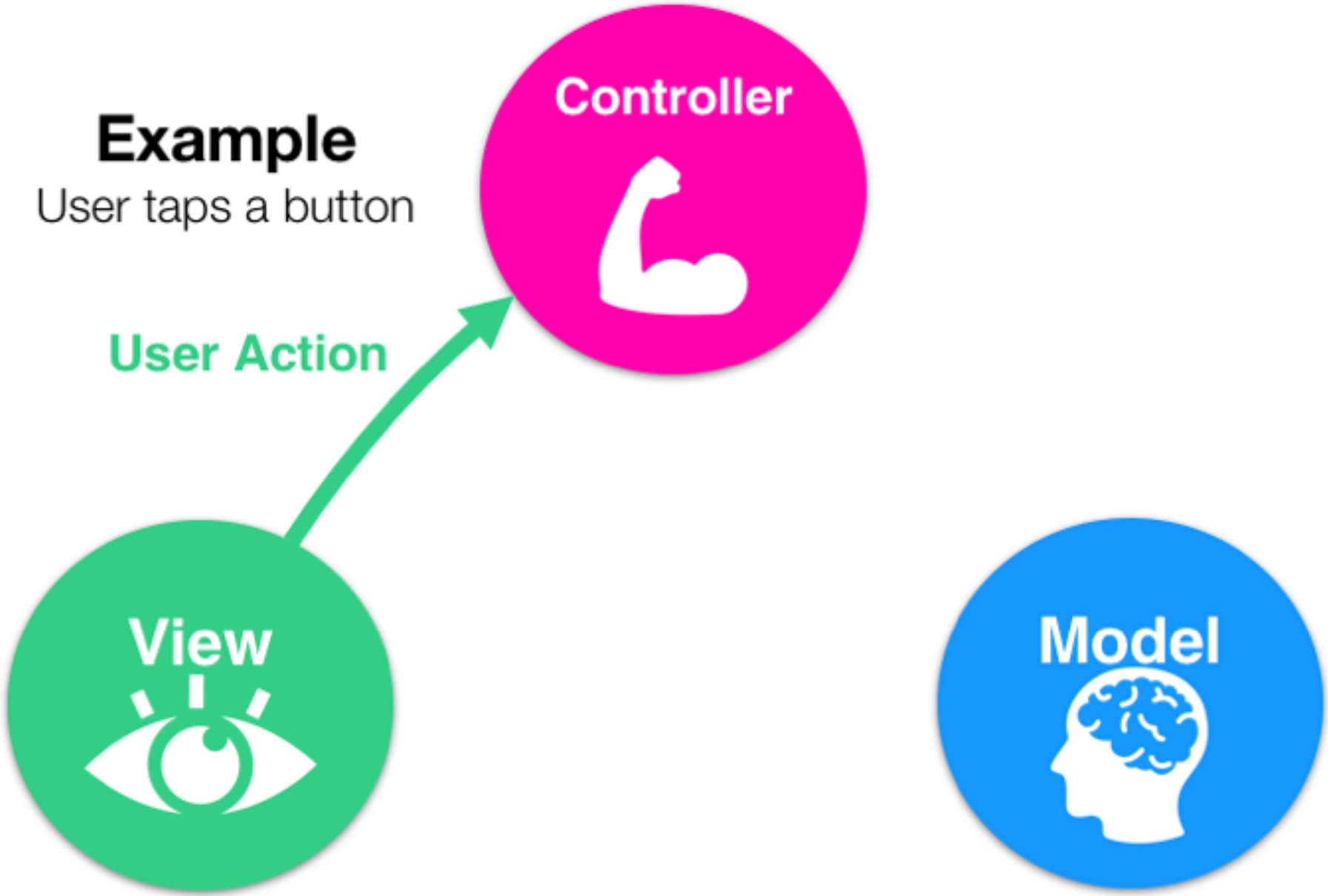
Example

User taps a button

User interacts with **View** (eg user taps "=" button in calculator)

Example

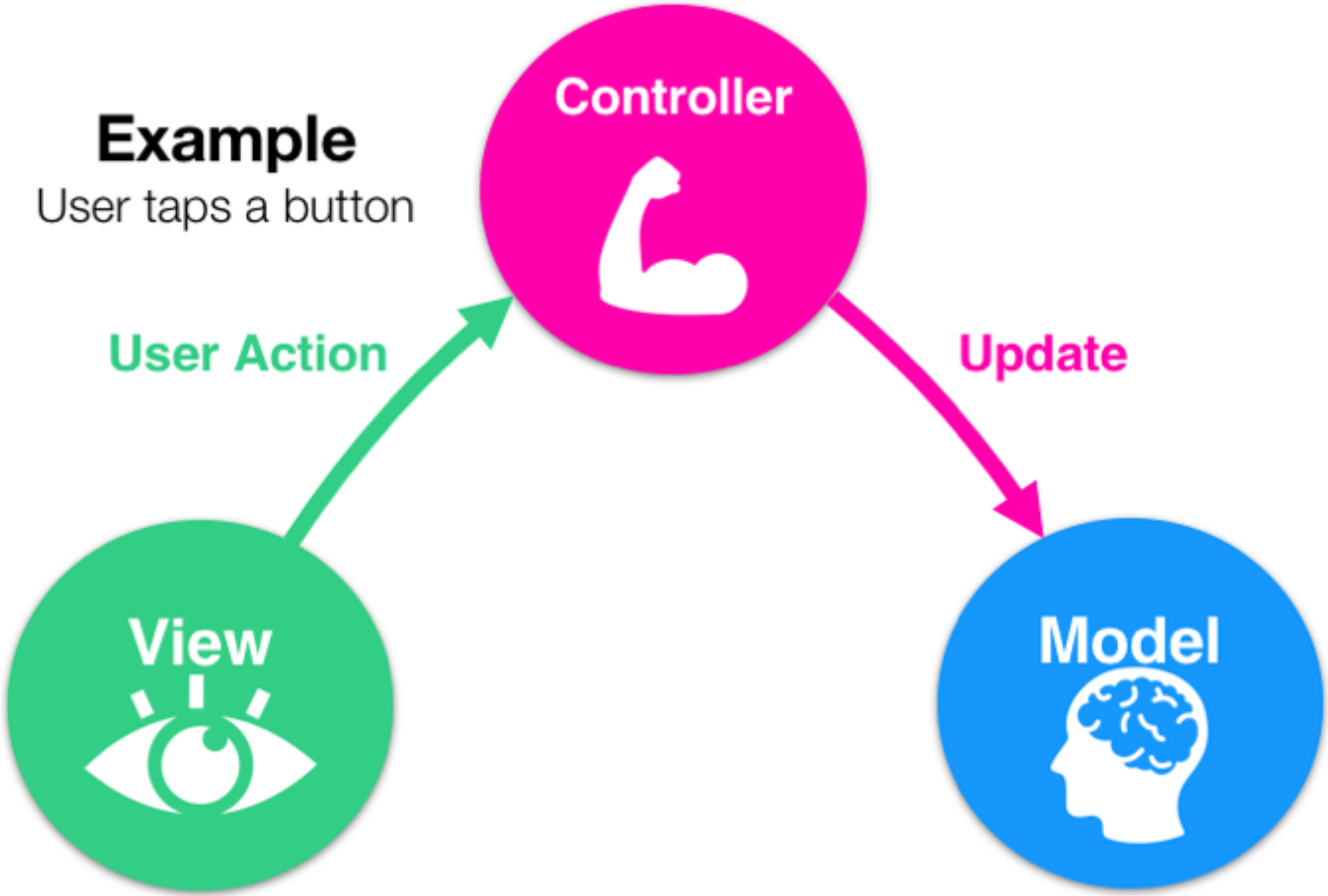
User taps a button



Controller is notified that the user has made an action

Example

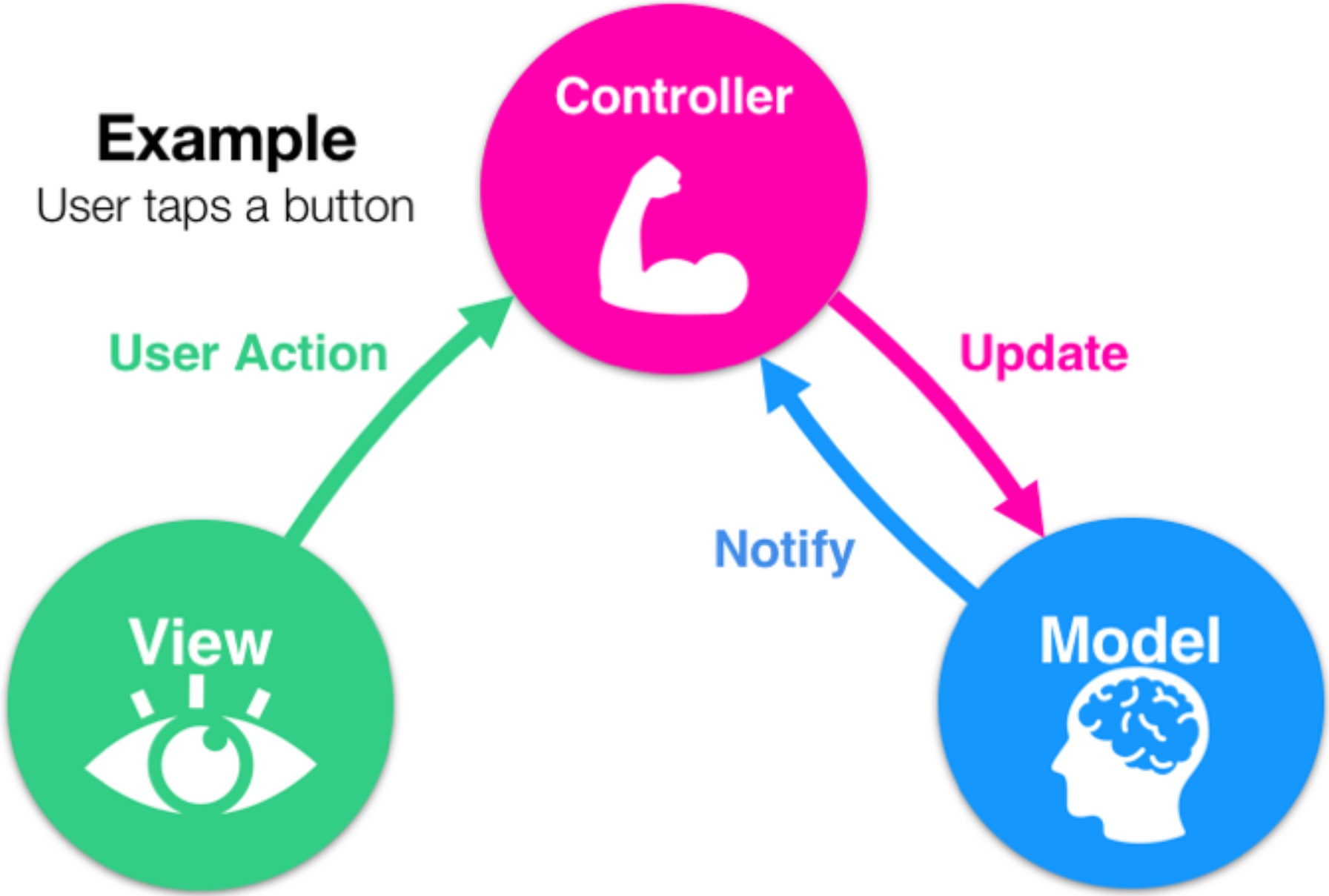
User taps a button



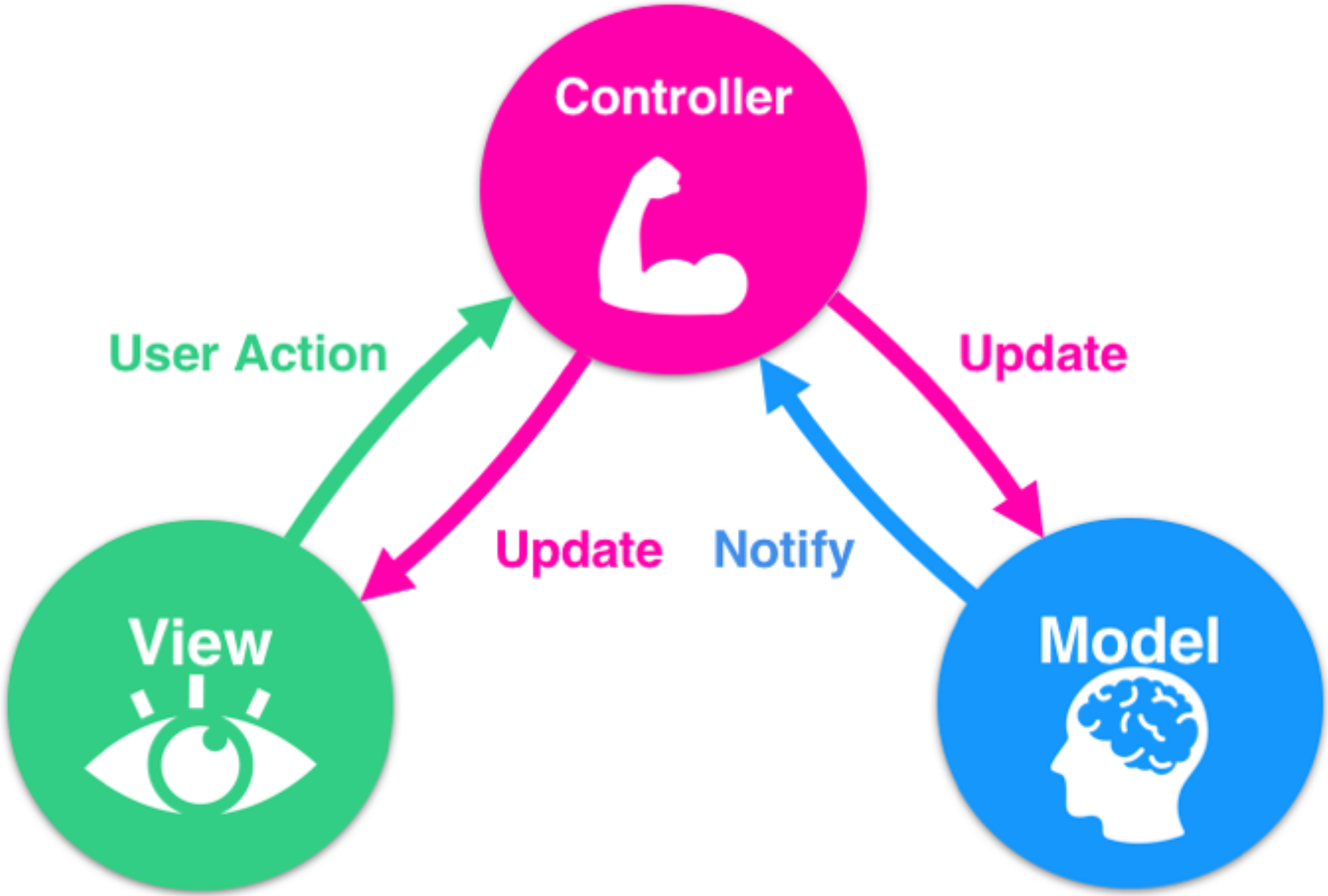
Controller updates the **Model** to reflect the users change

Example

User taps a button

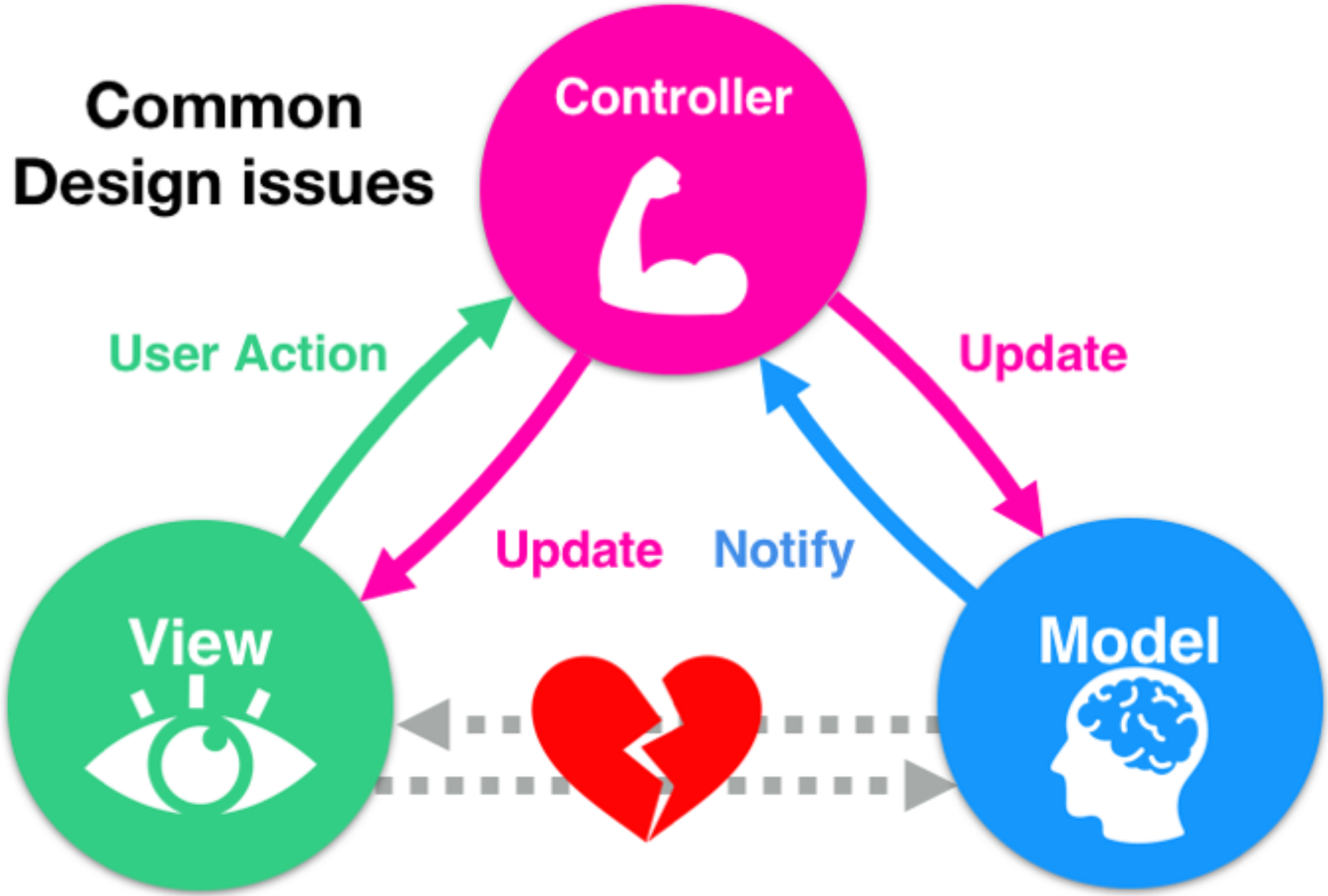


Model stores info / calculates then notifies controller of result



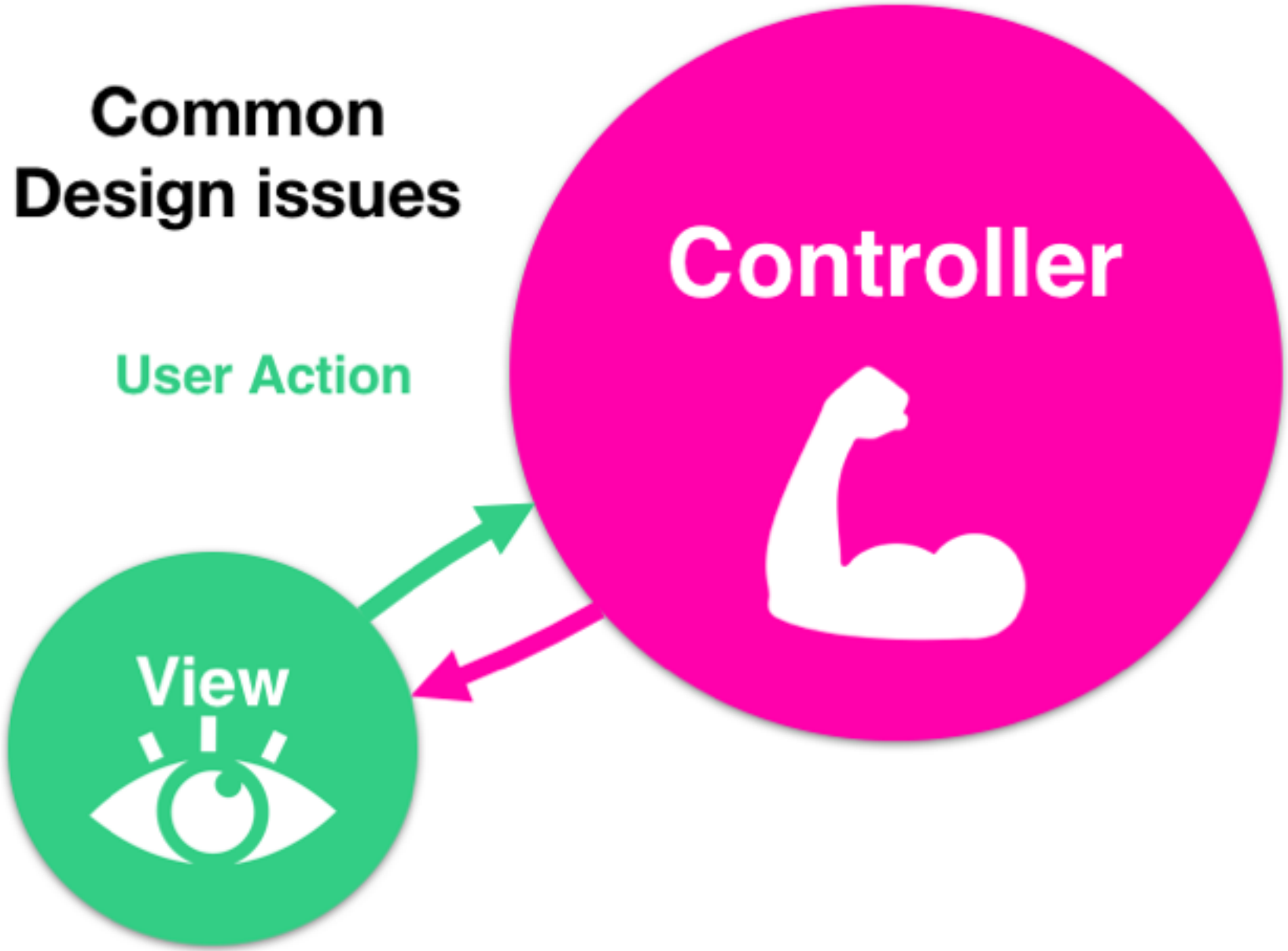
Notice how there is no direct connection from **Model** to **View**!

Common Design issues



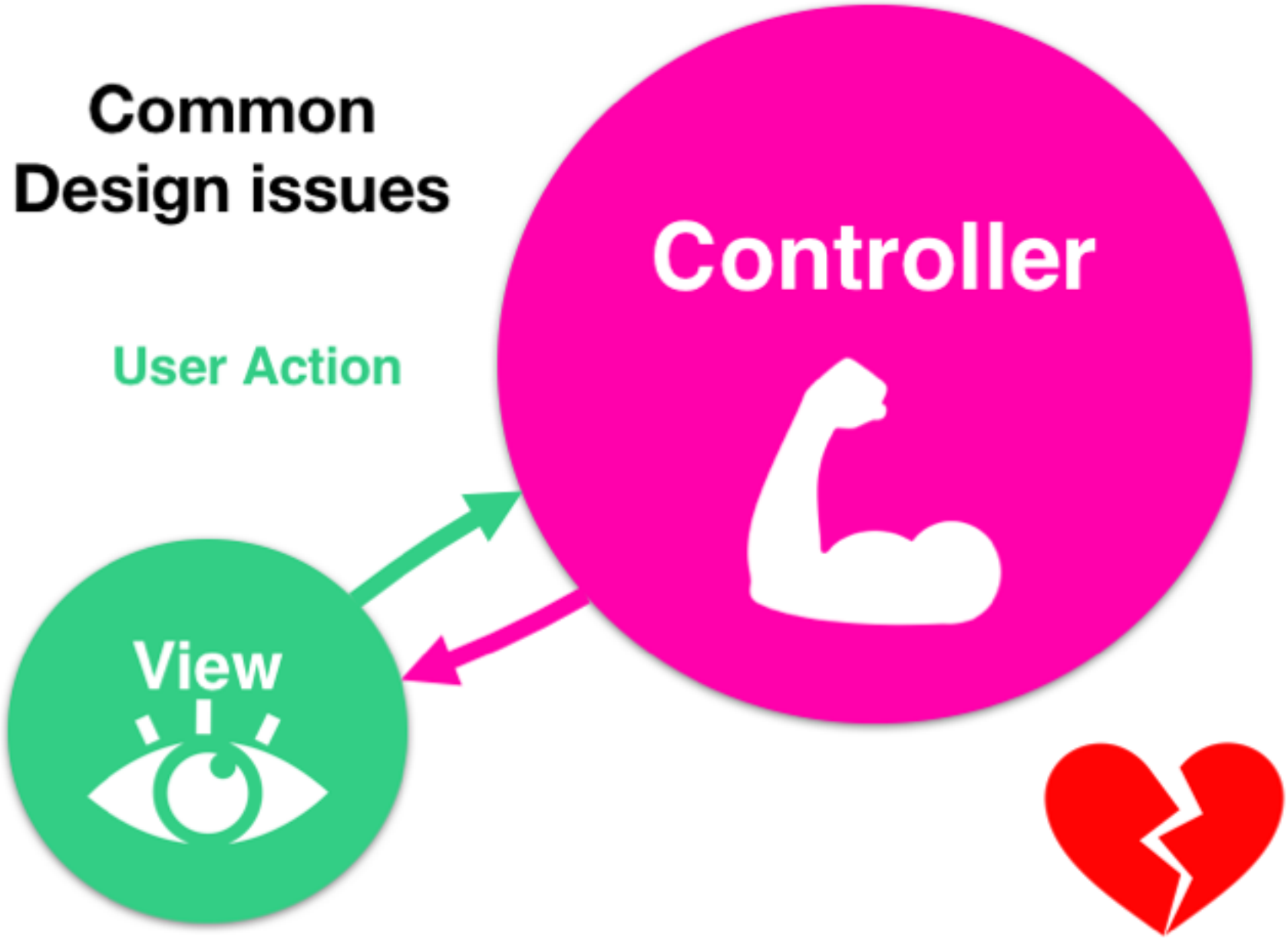
Issue #1 : Breaking Model / View Abstraction

Common Design issues



Issue #2 : Bloated Controllers

Common Design issues



Issue #2 : Bloated Controllers

MVC in Xcode

model

data, logic, computation

abstract classes

To Create

New > File > Swift File

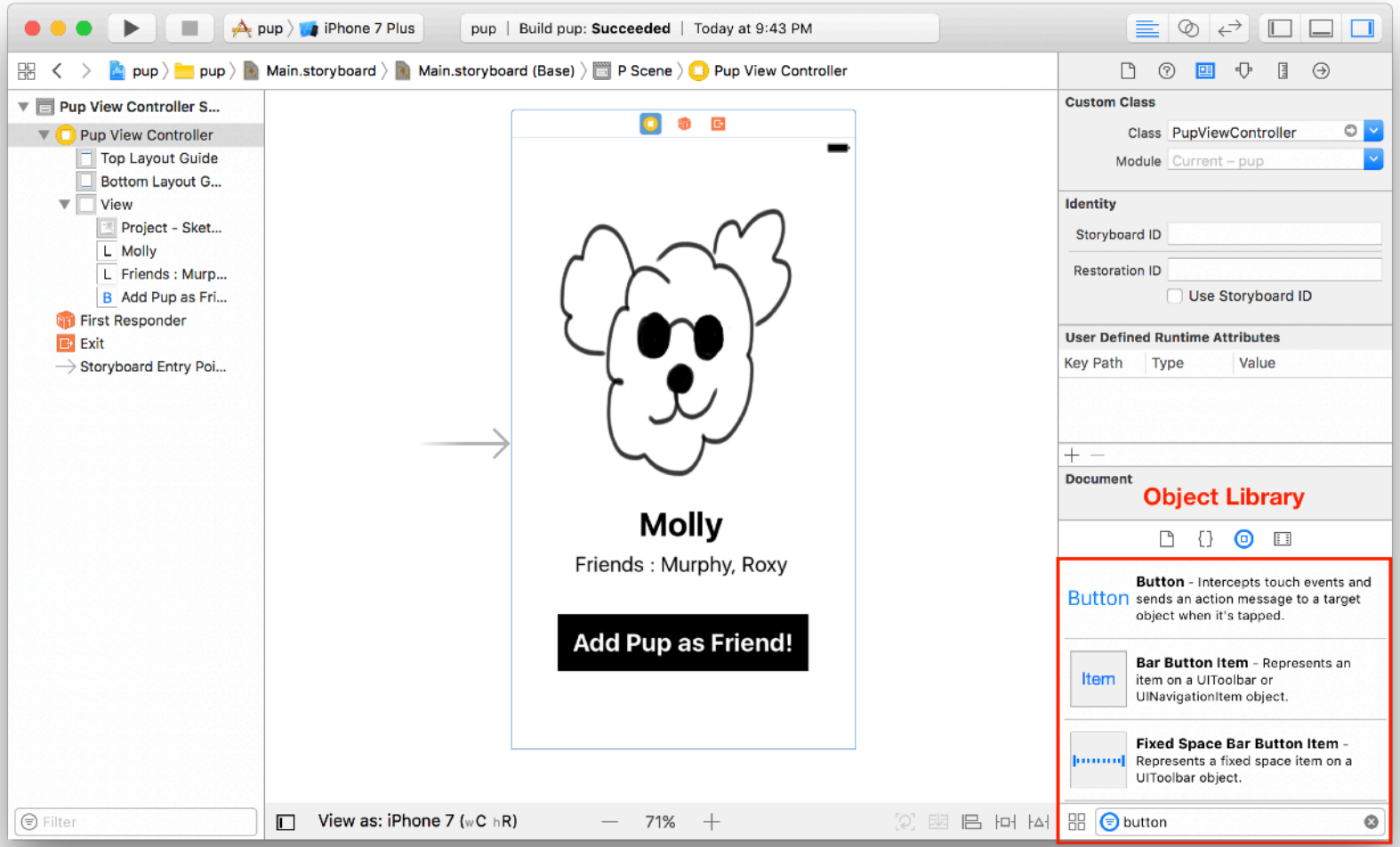
Puppy.swift — Edited

Puppy.swift 〉 No Selection

```swift
// Puppy.swift

import Foundation

/// Model Class for Pups
class Puppy {

    var name: String
    var friendList: [Puppy]?
    let location: Location?
    var superPower: SuperPower?

    init(pupName: String) {
        name = pupName
    }

    func bark(direction: Direction) {
        // ...
    }

    func makeNewFriend(withOtherPup pup: Puppy) {
        // ...
    }

    func activateSuperPower(superPower: SuperPower?) {
        // ...
    }
}
```

Swift File

views: interface builder



views: programmatic

Cocoa Touch Class

creating views programmatically

New > File > Cocoa Touch Class.

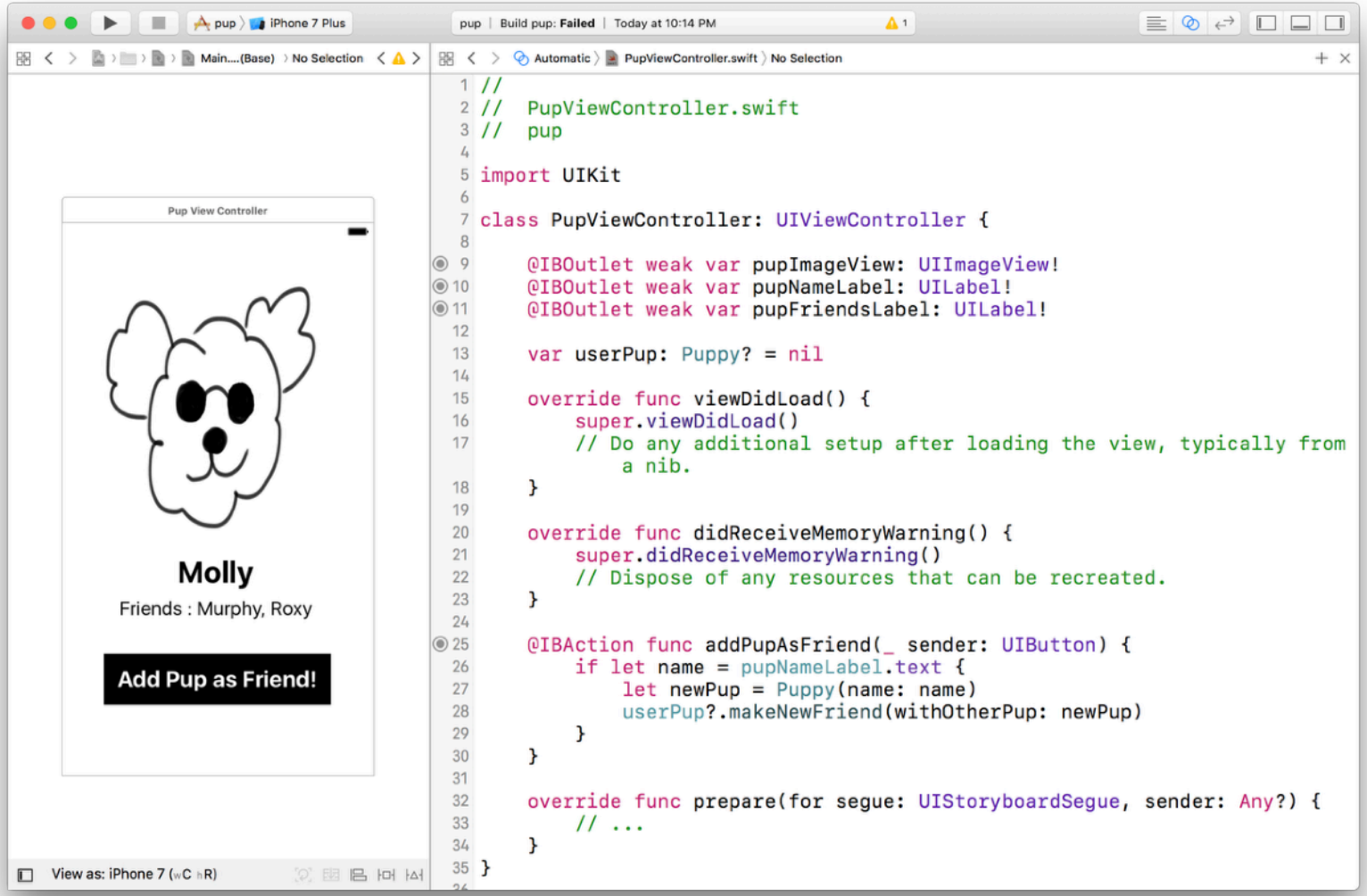
subclass an existing type of View

PupView.swift — Edited

PupView.swift 〉 No Selection

```swift
//  PupView.swift

import UIKit

class PupView: UIView {

    struct Constants {
        static let marginSize: CGFloat = 8
        static let buttonText: String = "Add Pup as Friend!"
    }

    let pupView = UIImageView(image: UIImage(named: "pup"))
    let friendsListLabel = UILabel()

    var addPupFriendButton: UIButton = {
        var button = UIButton()
        button.setTitle(Constants.buttonText, for: .normal)
        return button
    }()

    override init(frame: CGRect) {
        super.init(frame: CGRect.zero)
        addSubview(pupView)
        addSubview(friendsListLabel)
        addSubview(addPupFriendButton)
        setupConstraints()
    }

    func setupConstraints() {
        pupView.topAnchor.constraint(equalTo: topAnchor,
            constant: Constants.marginSize)
        // ...
    }
}
```

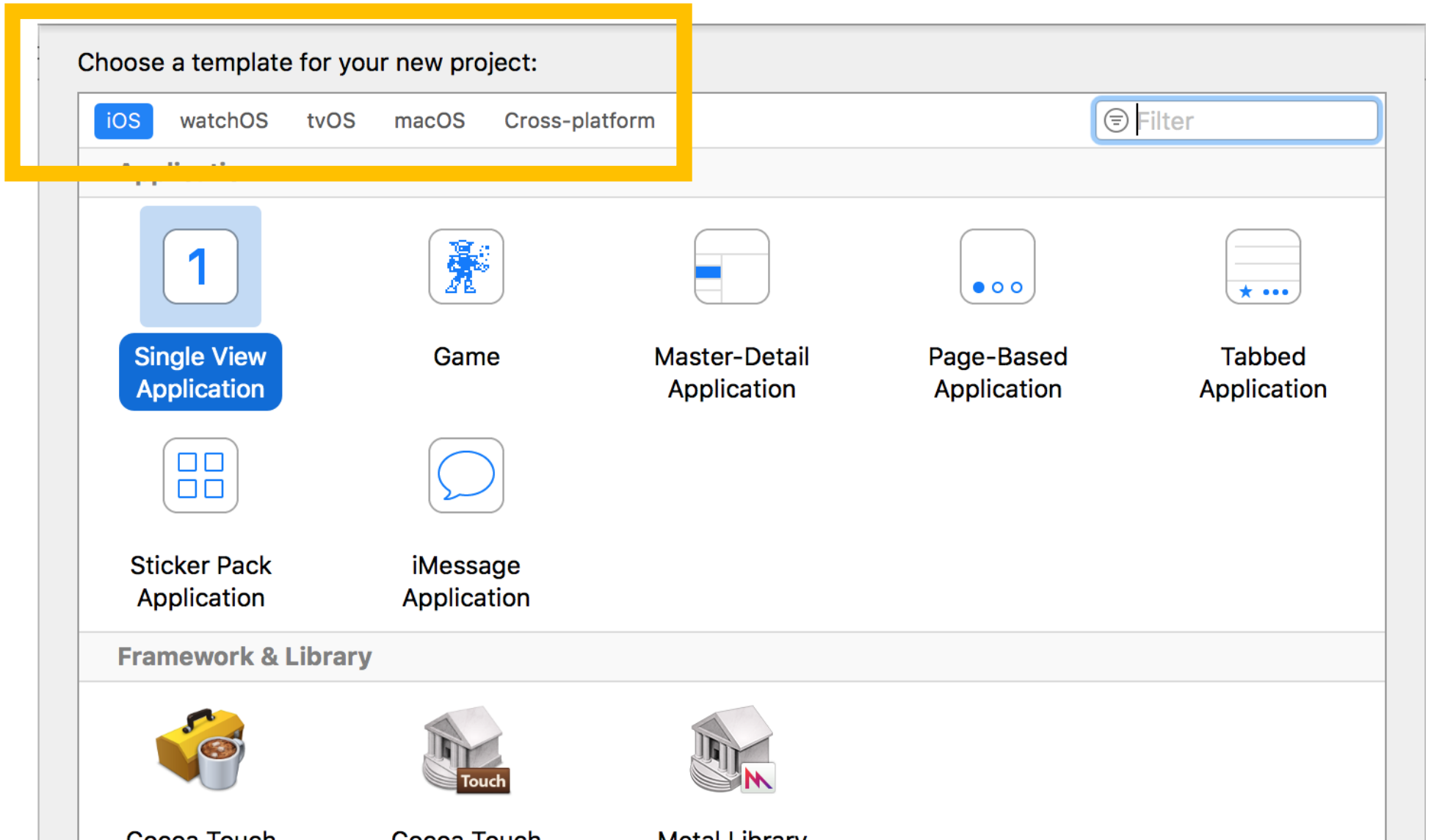
views controllers

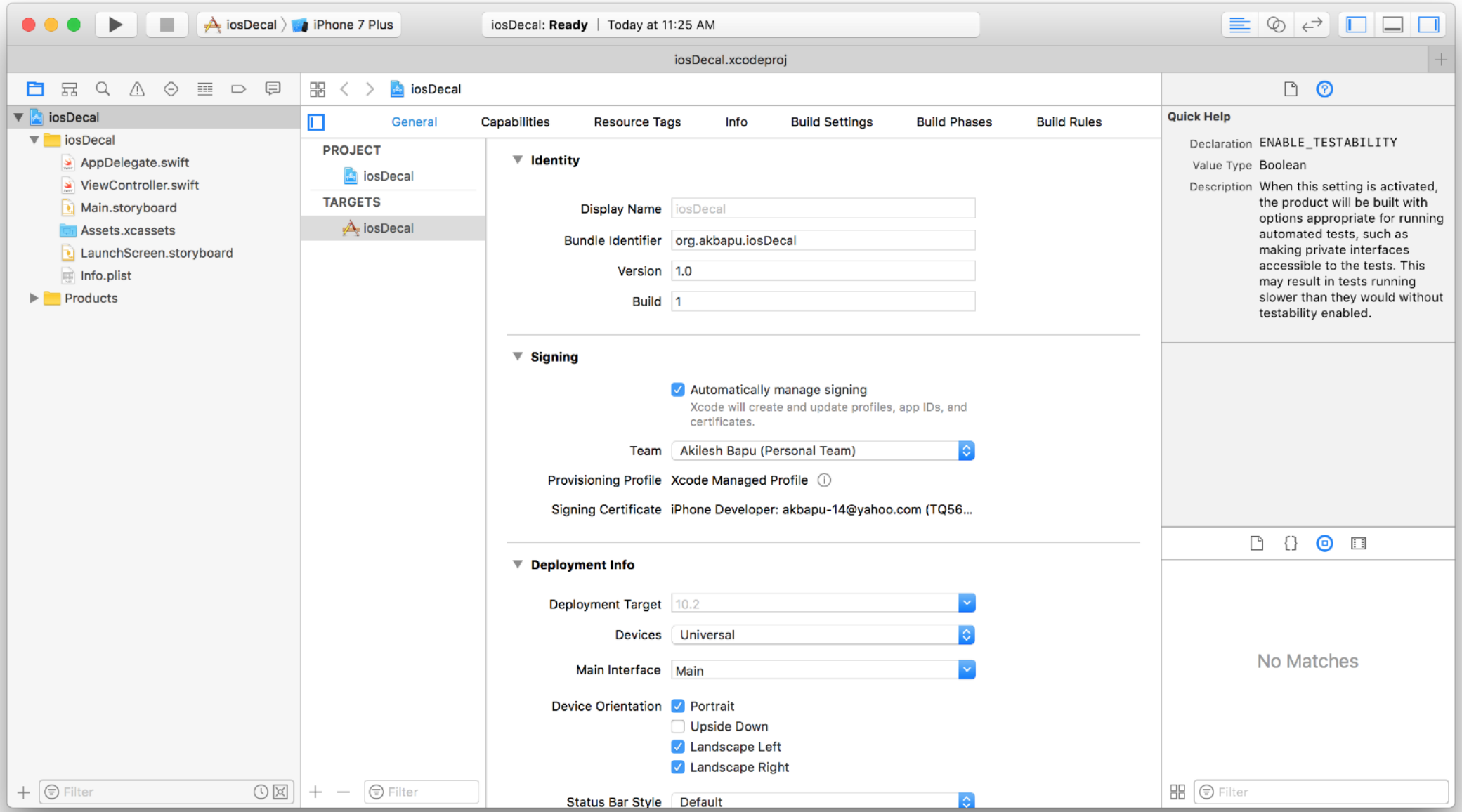


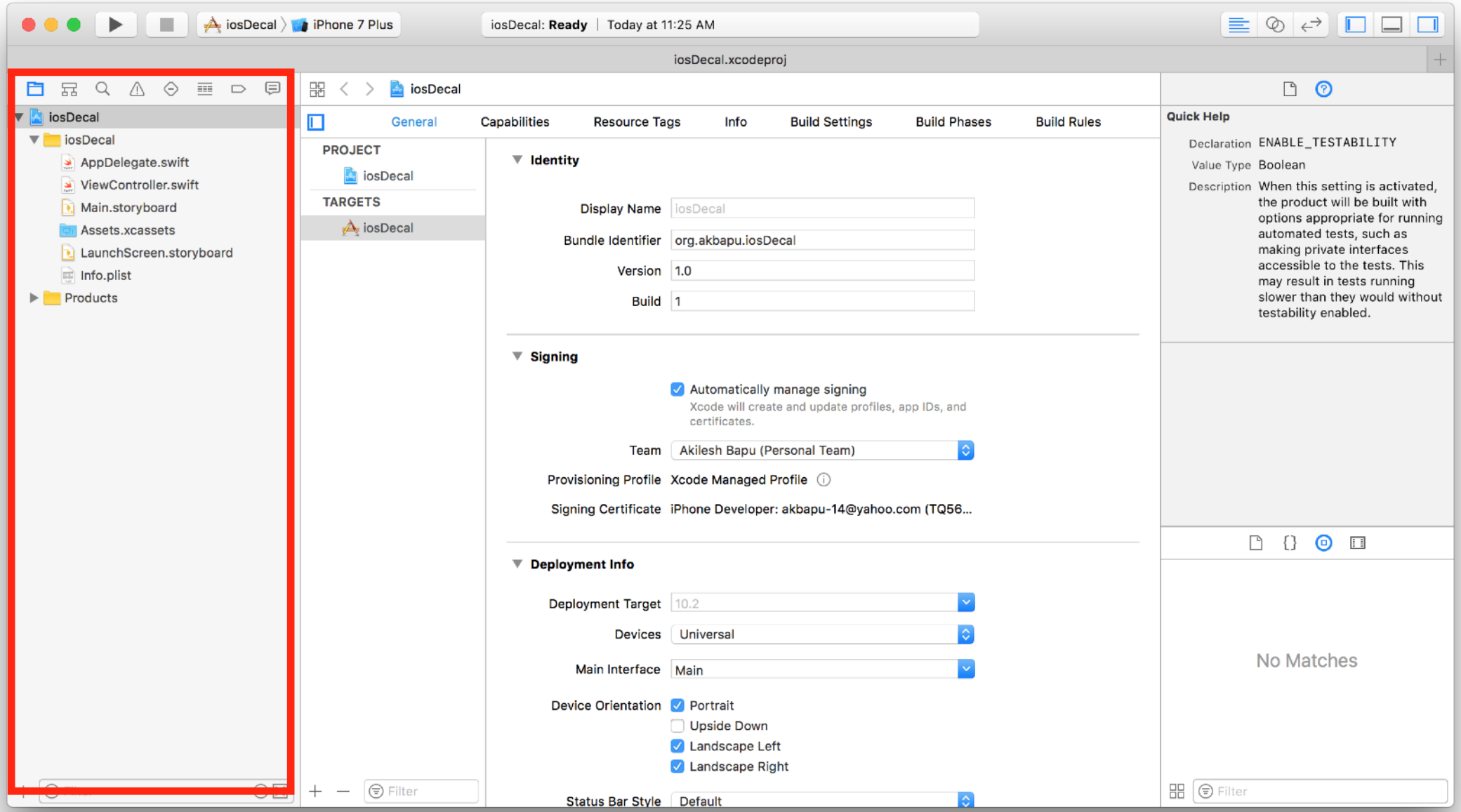
check in

Xcode

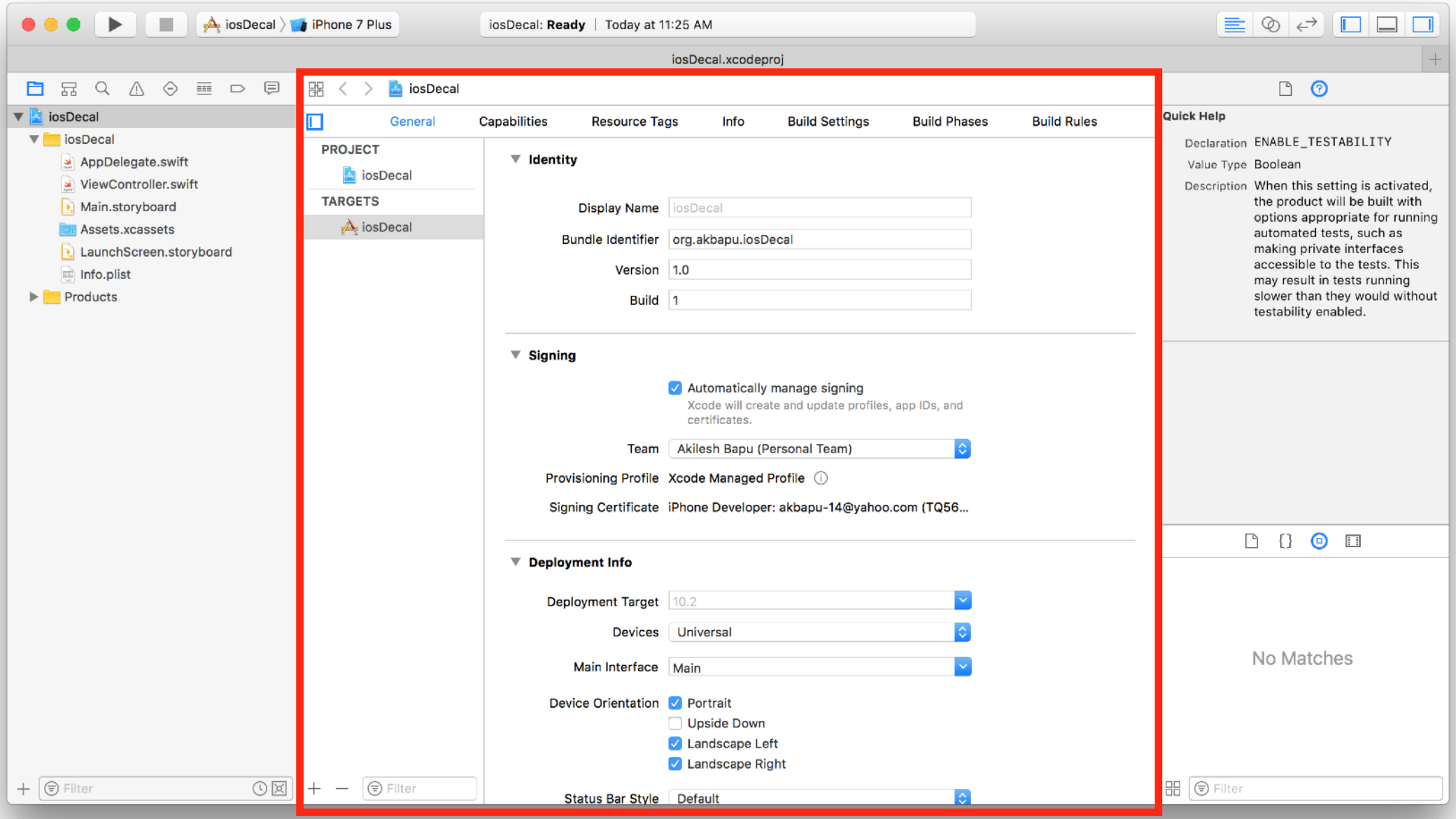
The IDE to Rule Them All



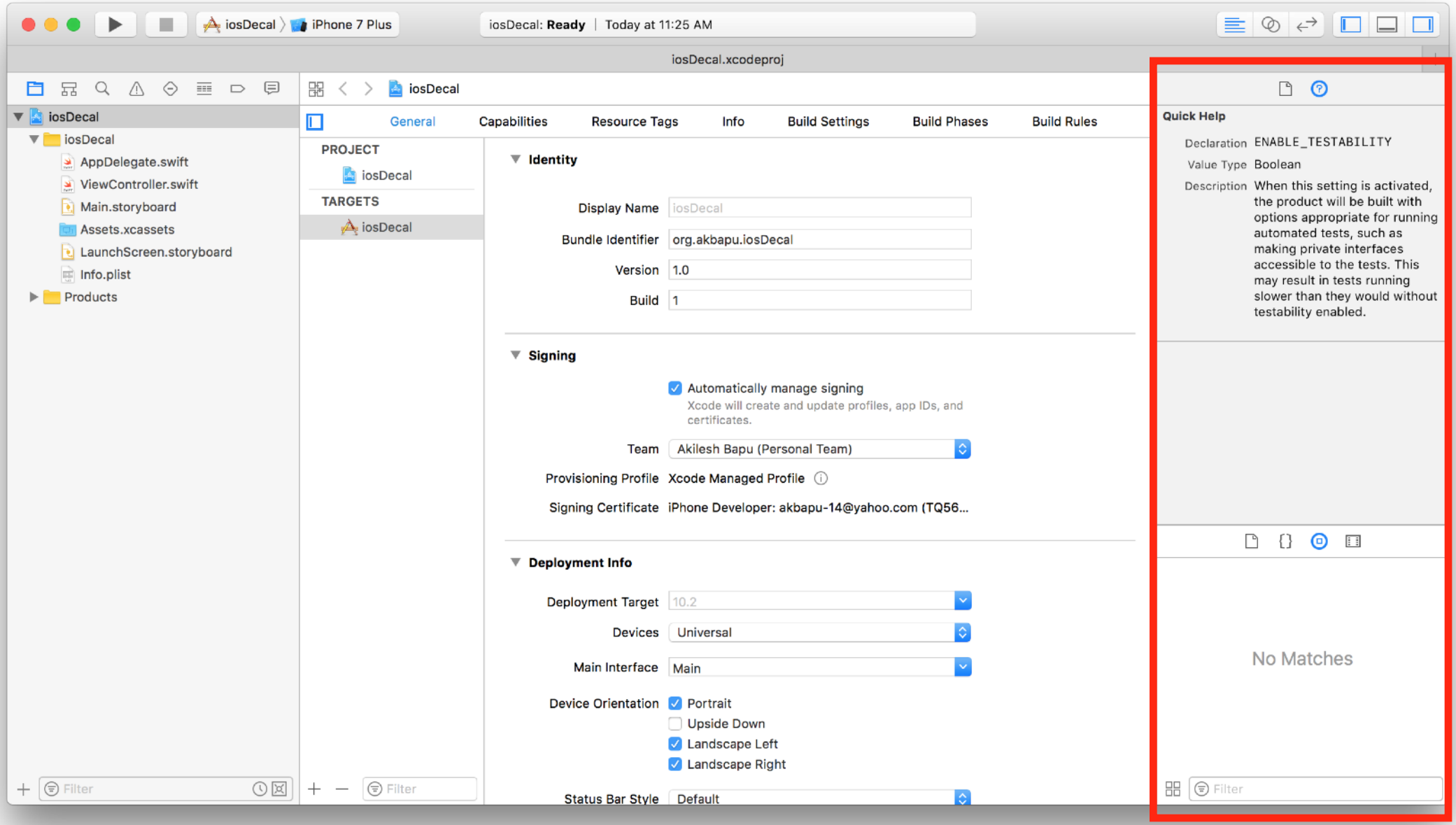

iosDecal 〉 iPhone 7 Plus
iosDecal: Ready | Today at 11:25 AM
iosDecal.xcodeproj
iosDecal
iosDecal
iosDecal
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Filter
iosDecal
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
iosDecal
TARGETS
iosDecal
Filter
Identity
Display Name
iosDecal
Bundle Identifier
org.akbapu.iosDecal
Version
1.0
Build
1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
Akilesh Bapu (Personal Team)
Provisioning Profile
Xcode Managed Profile
Signing Certificate
iPhone Developer: akbapu-14@yahoo.com (TQ56...
Deployment Info
Deployment Target
10.2
Devices
Universal
Main Interface
Main
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Quick Help
Declaration
ENABLE_TESTABILITY
Value Type
Boolean
Description
When this setting is activated, the product will be built with options appropriate for running automated tests, such as making private interfaces accessible to the tests. This may result in tests running slower than they would without testability enabled.
No Matches
Filter



Project Navigator



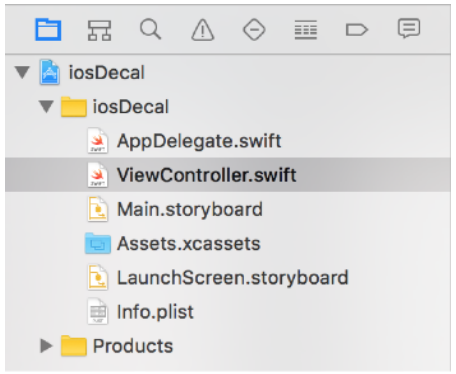
Project/Target Settings



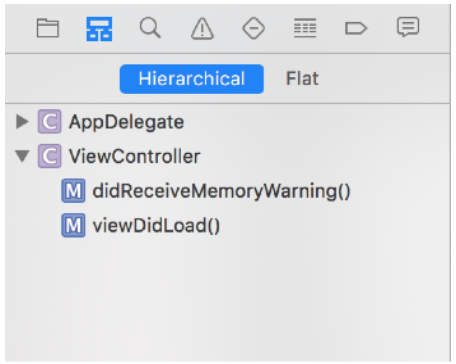
Utilities Area

Xcode structure: left panel

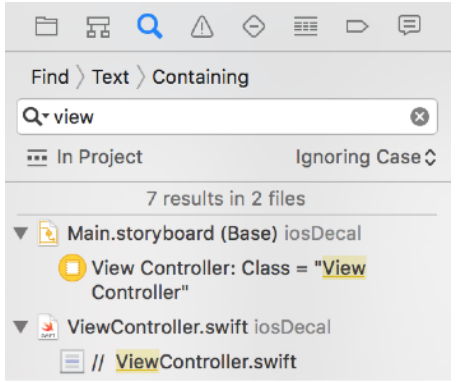
Project Navigator



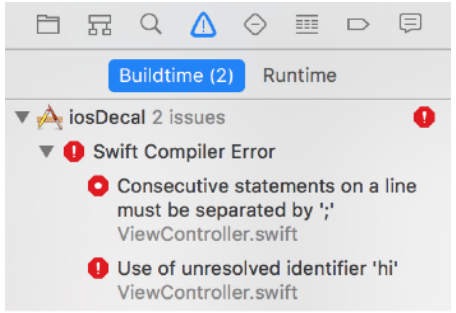
Symbol Navigator



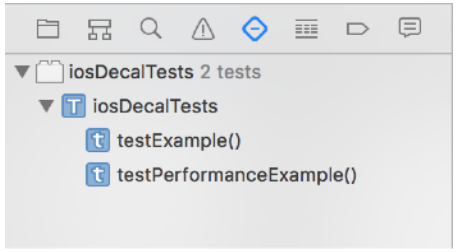
Find Navigator



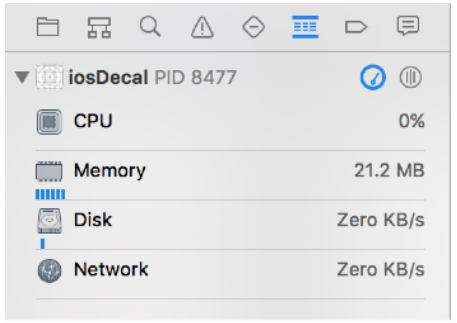
Issue Navigator



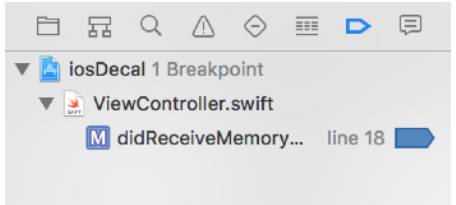
Test Navigator



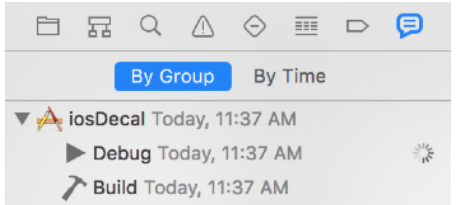
Debug Navigator



Breakpoint Navigator

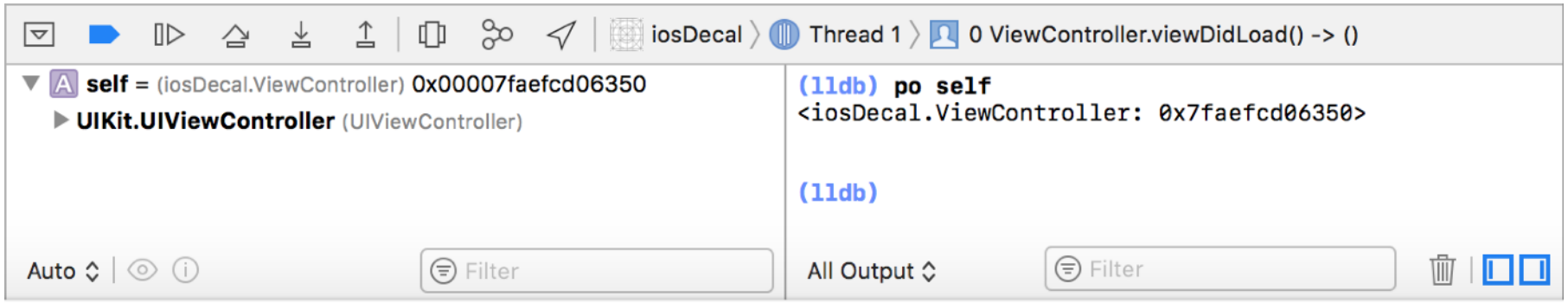


Report Navigator

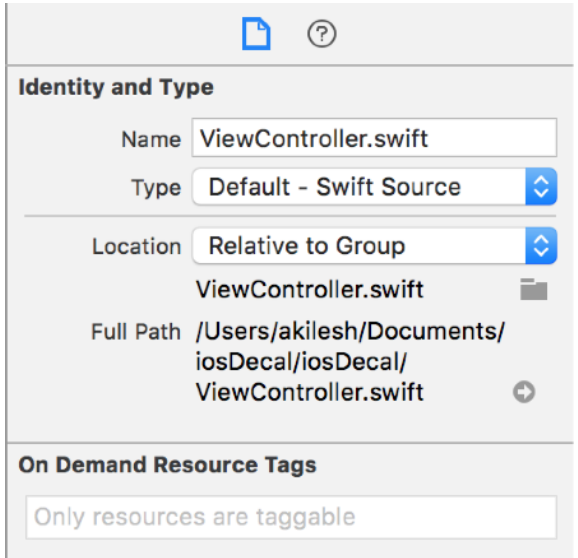


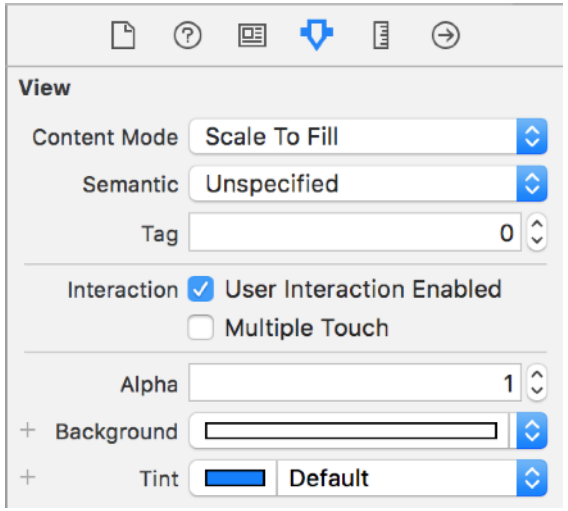
Xcode navigator

Debug Area

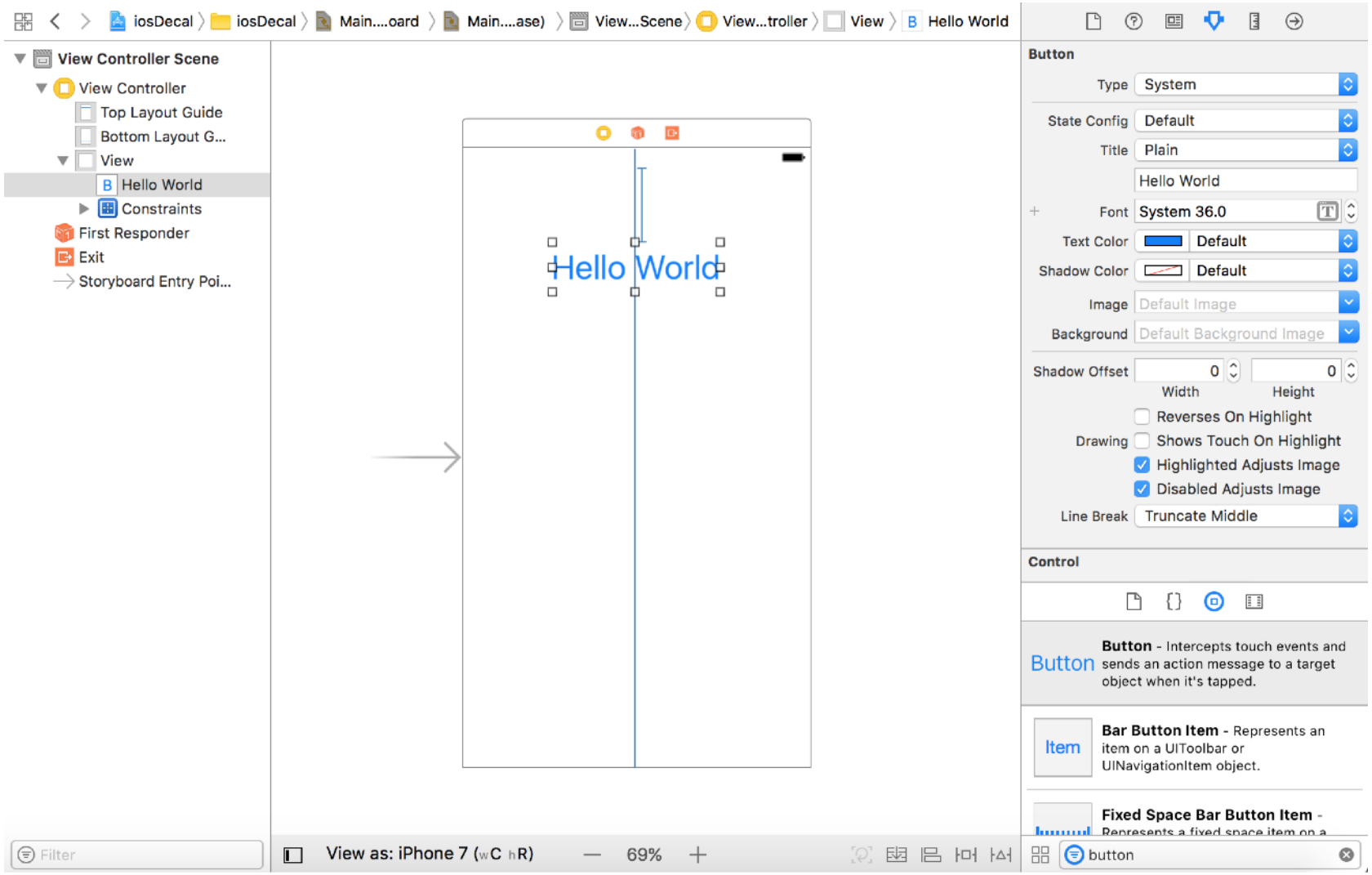


Utilities Area





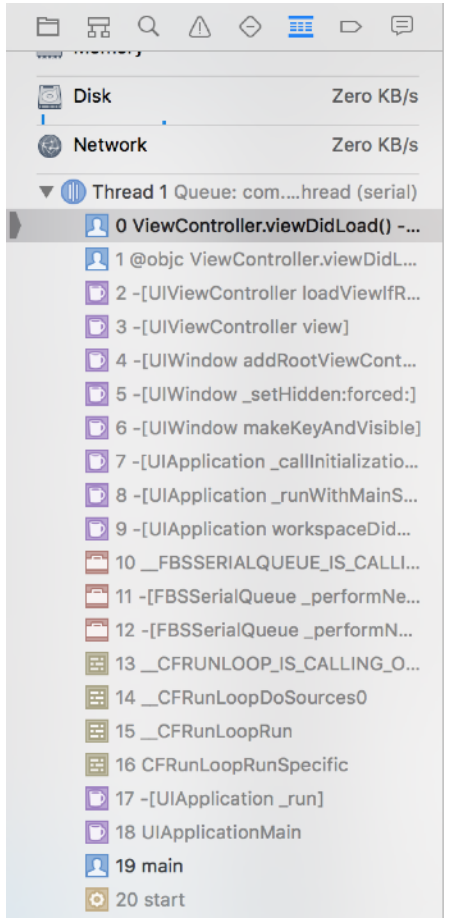
storyboards



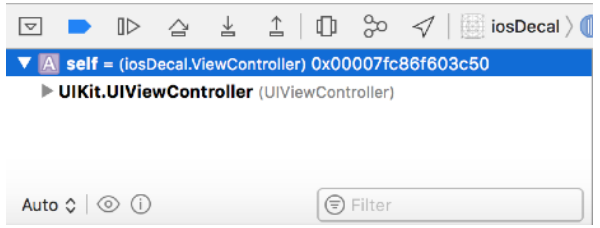
Xcode Tips and Tricks

debugging

Call Hierarchy - Great if you step through code and end up at some random file and want to see how it's relevant



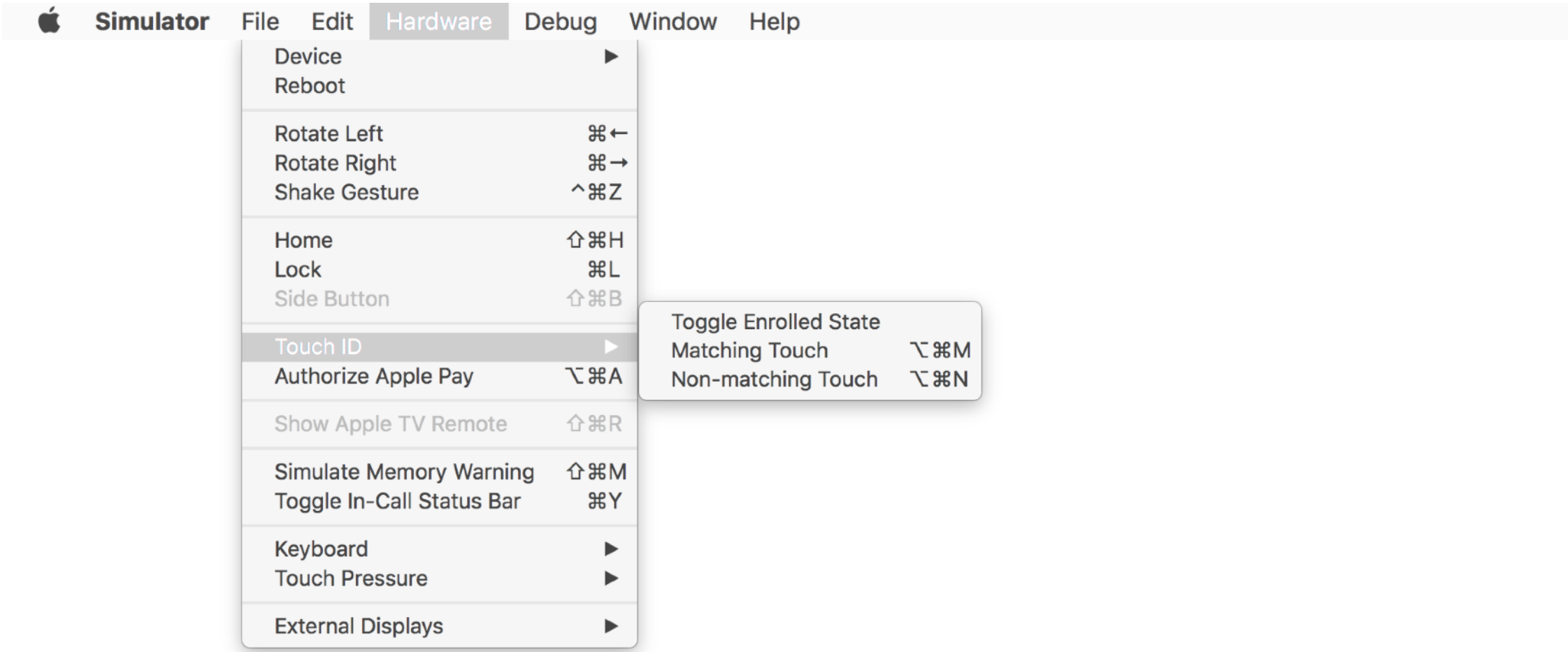
Currently Variables in Use - You can actually preview views



LLDB - Console Output + "po" (RUN CODE AS THE APP IS GOING)

simulator settings

Hardware and Debug Menus - Easily ignorable but amazingly helpful



Xcode IDE Demo

Let's Create an App

lab 1 – Xcode tutorial

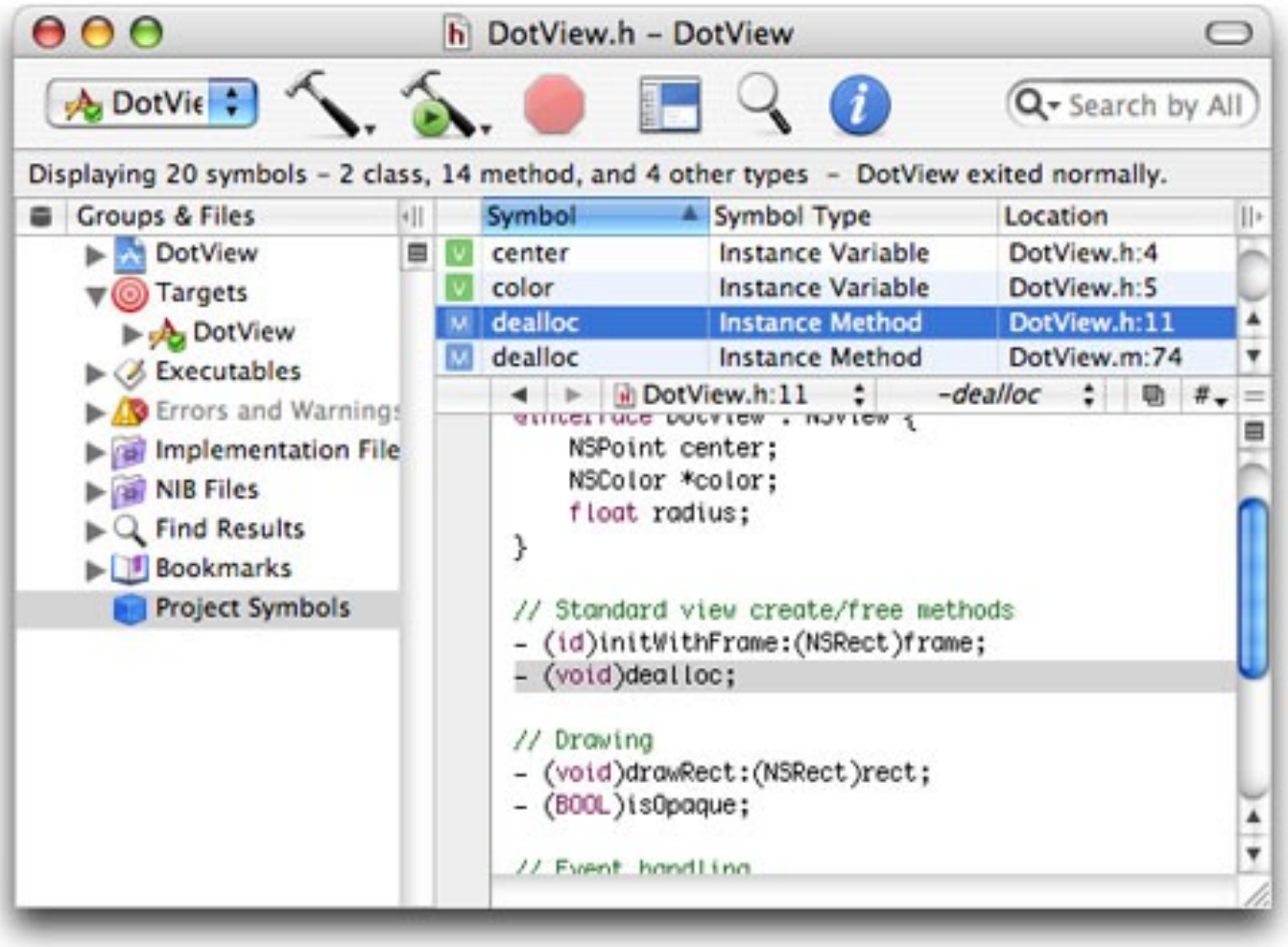
Wednesday at 7pm

remember to sign up for a lab section on piazza!

extra slides

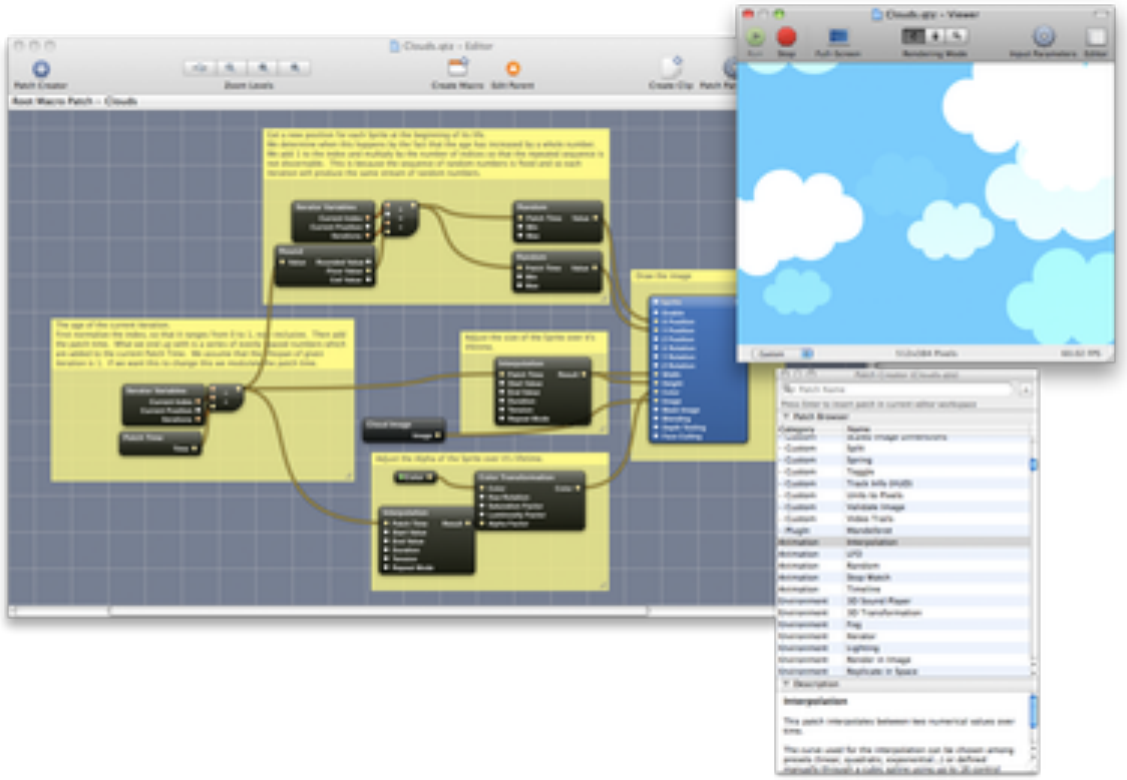
Project Builder

- IDE for the NeXTSTEP OS (source of macOS, iOS, tvOS, etc.)
- rewritten for OS X, and rebranded as **Xcode**



Xcode versions

- 2003: Xcode 1.0
- 2005: Xcode 2.x
  - included a visual programming language (Quartz Composer)
  - breakpoints and watchpoints
- 2007: Xcode 3.x
  - 2008: iOS SDK released to third party developers



the Quartz Composer