

Mobile Applications Development 2 (MAD2)

Diploma in IT

Teaching Team:
Dr Joel Yang
Mr Charles Keck

MAD2 Oct 2018

Chapter 6

Alert View and CocoaPod



Objectives

To be able to understand:

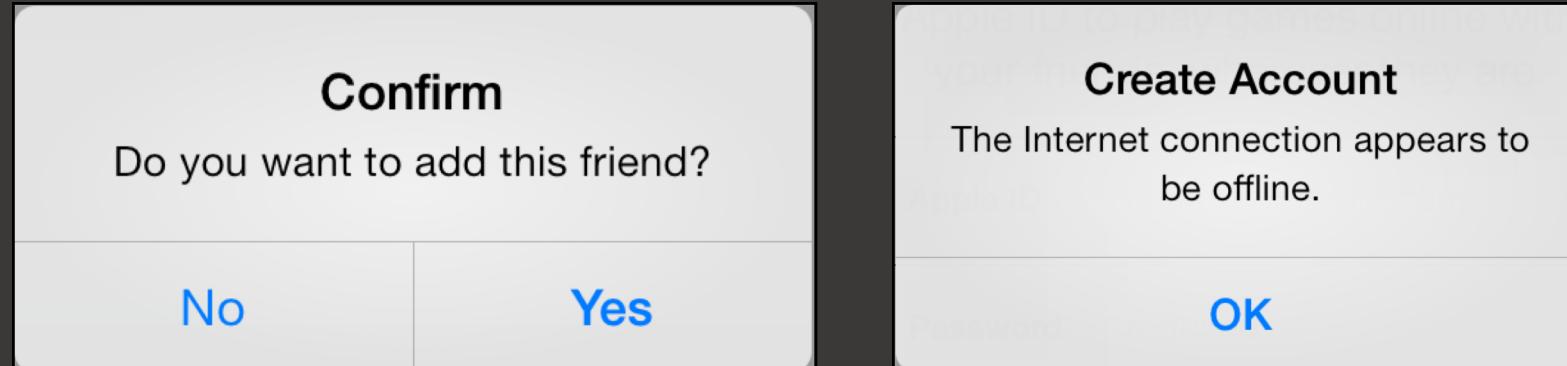
- Alert View with UIAlertController
- Dependancy Manager, Libraries & Frameworks : CocoaPod



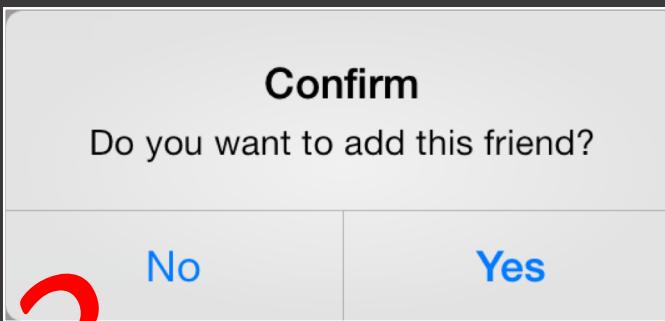
Alert View

Alert View with UIAlertController

- Used to **display an alert message** to the user
- Helps to provide good user experience
 - Visual feedback
 - Confirmation for user actions
 - Data validation message
- Configurable buttons



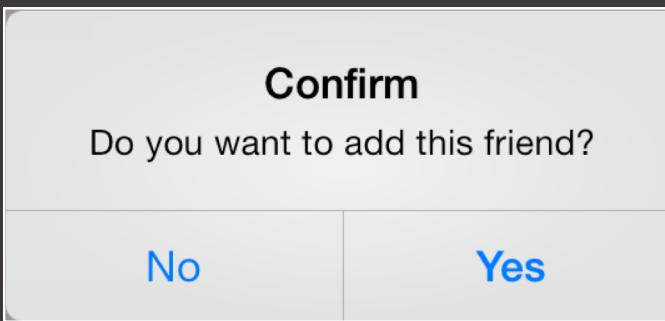
Creating Alert View



```
let alertView = UIAlertController(  
    title: "Confirm",  
    message: "Do you want to add this friend?",  
    preferredStyle: UIAlertControllerStyle.Alert)  
  
alertView.addAction(UIAlertAction(title: "Ok",  
    style: UIAlertActionStyle.Default,  
    handler: nil)) //to be discussed  
  
alertView.addAction(UIAlertAction(title: "Cancel",  
    style: UIAlertActionStyle.Default,  
    handler: nil)) //to be discussed  
  
dispatch_async(dispatch_get_main_queue(), {  
    self.presentViewController(alertView, animated:true, completion:nil)  
})
```

SWIFT 2 and 3

Creating Alert View



```
let alertView = UIAlertController(  
    title: "Confirm",  
    message: "Do you want to add this friend?",  
    preferredStyle: UIAlertController.Style.alert)  
  
alertView.addAction(UIAlertAction(title: "Ok",  
    style: UIAlertAction.Style.default,  
    handler: nil)) //to be discussed  
  
alertView.addAction(UIAlertAction(title: "Cancel",  
    style: UIAlertAction.Style.default,  
    handler: nil)) //to be discussed  
  
DispatchQueue.main.async {  
    self.present(alertView, animated:true, completion:nil)  
}
```

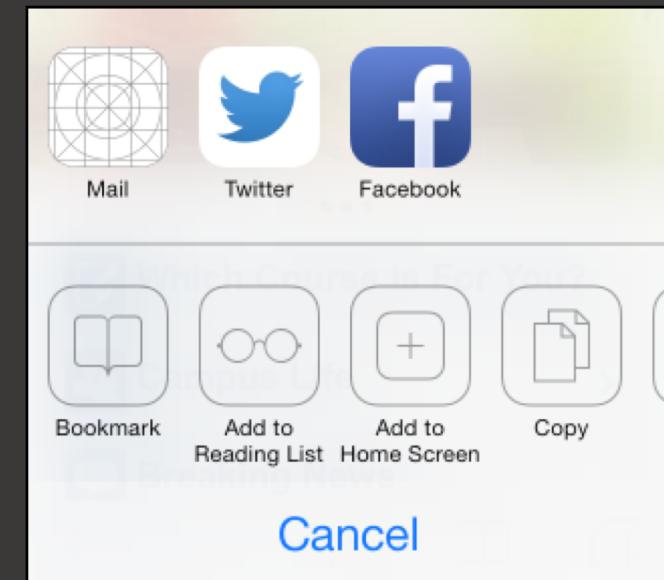
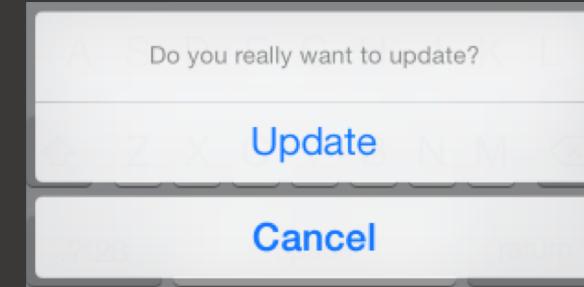
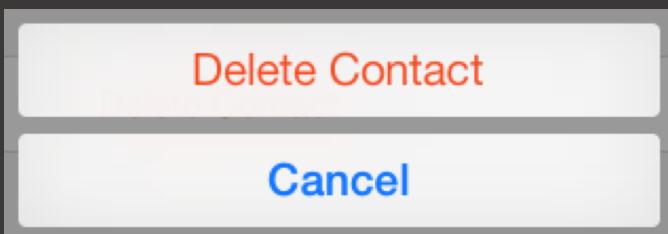
UIAlertAction with Handler

```
func someHandler(alert: UIAlertAction!) {
    print("OK is Clicked!")
}

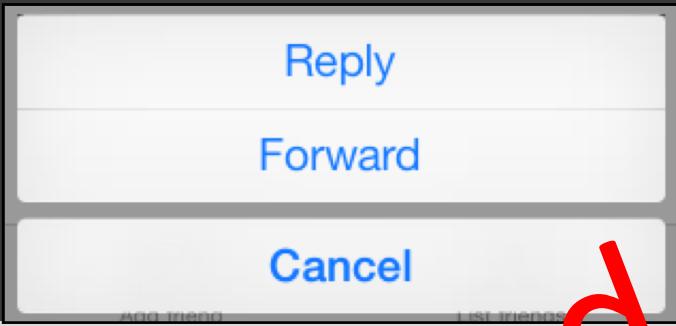
alertView.addAction(UIAlertAction(title: "Ok",
                                  style:
UIAlertAction.Style.default, handler: someHandler))
```

ActionSheet with UIAlertController

- Like the Alert view, this is a dialog to present the user with a set of alternatives for how to proceed with a given task
- Contains an optional title and one or more buttons



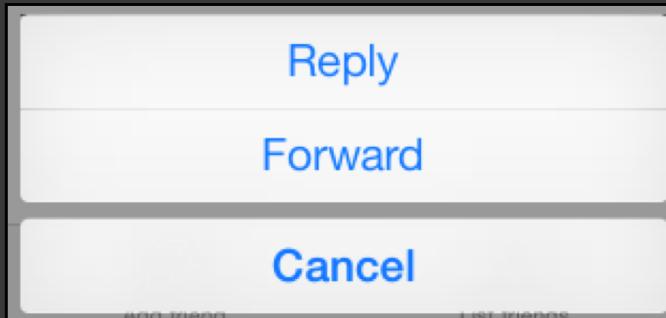
Creating ActionSheet



```
let actionSheet = UIAlertController(  
    title: "Choose option",  
    message: "",  
    preferredStyle: UIAlertController.Style.ActionSheet)  
  
    actionSheet.addAction(UIAlertAction(title: "Reply",  
        style: UIAlertActionStyle.Default, handler: nil))  
    actionSheet.addAction(UIAlertAction(title: "Forward",  
        style: UIAlertActionStyle.Default, handler: nil))  
  
    actionSheet.addAction(UIAlertAction(title: "Cancel",  
        style: UIAlertActionStyle.Cancel, handler: nil))  
  
    dispatch_async(dispatch_get_main_queue(), {  
        self.presentViewController(actionSheet, animated:true, completion:nil)  
    })
```

SWIFT 2 and 3

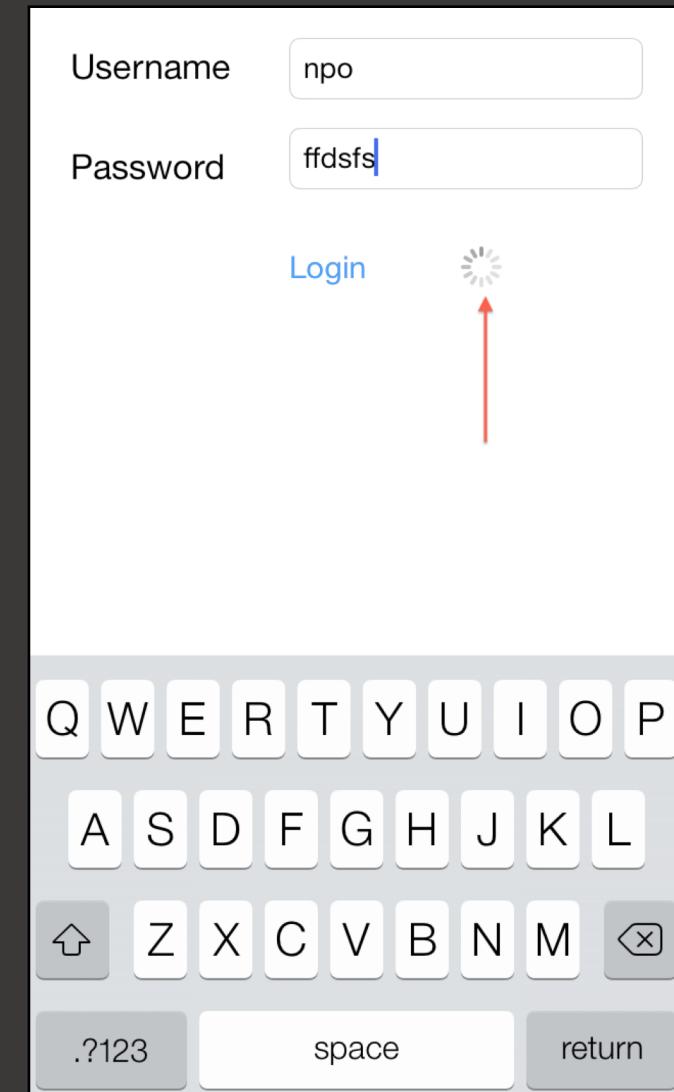
Creating ActionSheet



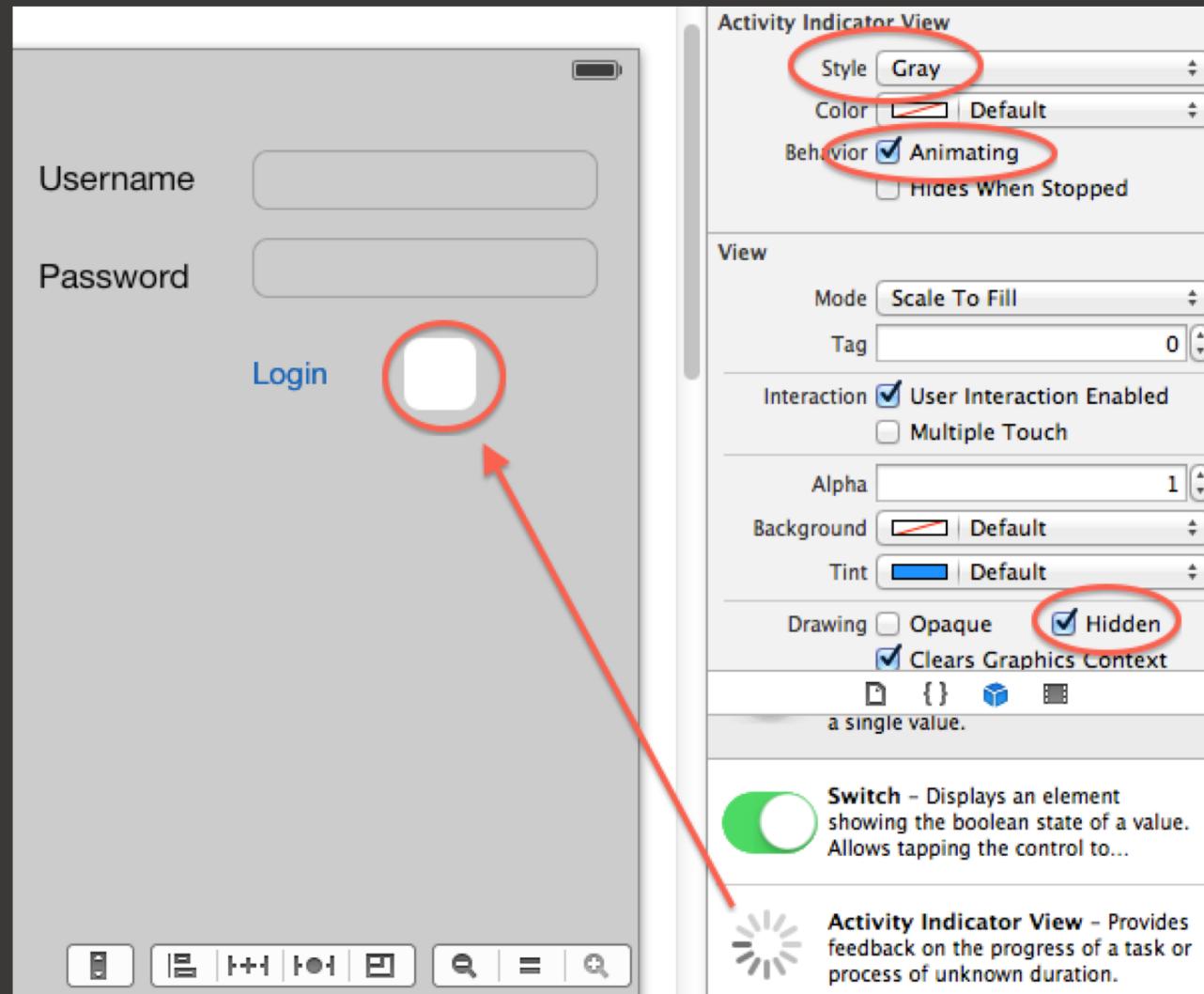
```
let actionSheet = UIAlertController(  
    title: "Choose option",  
    message: "",  
    preferredStyle: UIAlertController.Style.actionSheet)  
  
actionSheet.addAction(UIAlertAction(title: "Reply",  
    style: UIAlertAction.Style.default, handler: nil))  
actionSheet.addAction(UIAlertAction(title: "Forward",  
    style: UIAlertAction.Style.default, handler: nil))  
actionSheet.addAction(UIAlertAction(title: "Cancel",  
    style: UIAlertAction.Style.cancel, handler: nil))  
  
DispatchQueue.main.async {  
    self.present(actionSheet, animated:true, completion:nil)  
}
```

UIActivityIndicator

- Visual feedback to inform user to “Please wait”
- Tells user something is happening and that the app is still running
- Creates a positive user experience



UIActivityIndicator



MAD2 Oct 2018

UIActivityIndicatorView

```
@IBOutlet weak var activityIndicator: UIActivityIndicatorView!

@IBAction func onClicked(sender: AnyObject) {
    activityIndicator.hidden = false
    NSThread.detachNewThreadSelector(Selector("mythread"),
                                    toTarget: self, withObject: nil)
}

func mythread() {
    NSThread.sleepForTimeInterval(3.0) //Sleep for 3 seconds
    for var i = 1 ; i <= 3 ; i++ {
        print("\(i)")
    }
    self.performSelector("onCompletion", withObject: nil)
}

func onCompletion(){
    activityIndicator.stopAnimating()
}
```

SWIFT 2 and 3

UIActivityIndicatorView

```
@IBOutlet weak var activityIndicator: UIActivityIndicatorView!

@IBAction func onClicked(_ sender: Any) {
    activityIndicator.isHidden = false
    activityIndicator.startAnimating()

    DispatchQueue.global(qos: .userInitiated).async {
        self.mythread()
    }
}

func mythread() {
    for i in 1...3 {
        Thread.sleep(forTimeInterval: 3.0) //Sleep for 3 seconds
        print("\(i)")
    }
    DispatchQueue.main.async {
        // Update the UI
        self.onCompletion()
    }
}

func onCompletion(){
    activityIndicator.stopAnimating()
    activityIndicator.isHidden = true
}
```

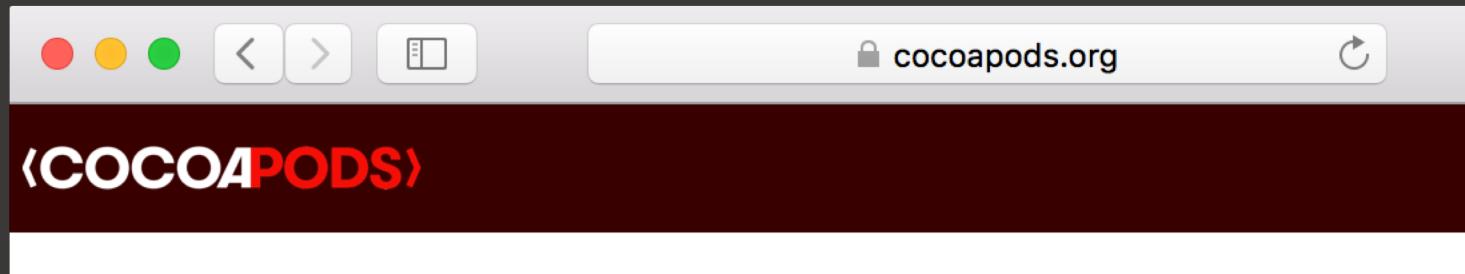


Libraries & Frameworks

CocoaPods

What is CocoaPods?

- CocoaPods is a **dependency manager** for Swift and Objective-C Cocoa projects.
- It has over **53 thousand libraries** and is used in over **3 million apps**.



Setting up

- <https://www.youtube.com/watch?v=iEAjvNRdZa0>



MAD2 Oct 2018

Search

The screenshot shows a web browser window on a Mac OS X desktop. The URL in the address bar is `cocoapods.org`. The search query is `QRCode`, which has resulted in 20 Swift-only pods. A red box highlights the first result, `QRCodeReader.swift`, with its version `7.2.0`. A callout bubble above the pod's card contains the text `For your Podfile` and the command `pod 'QRCodeReader.swift', '~> 7.2'`.

QRCode

Any iOS macOS watchOS tvOS All Swift Obj-C Sort by: Quality

20 results. Show only: Name (10)

PODS – SWIFT ONLY, AN

QRCodeReader.swift 7.2.0

By Yannick Loriot [yannickloriot](#)

[yannickl/QRCodeReader.swift](#)

[OVERVIEW](#) [CHANGELOG](#)

For your Podfile

```
pod 'QRCodeReader.swift', '~> 7.2'
```

Documented ✓
Tested ✗
Language Swift
License MIT
Last Release Oct 2016
Supports SPM ✓

Maintained by [Yannick Loriot](#).

Downloads

Steps

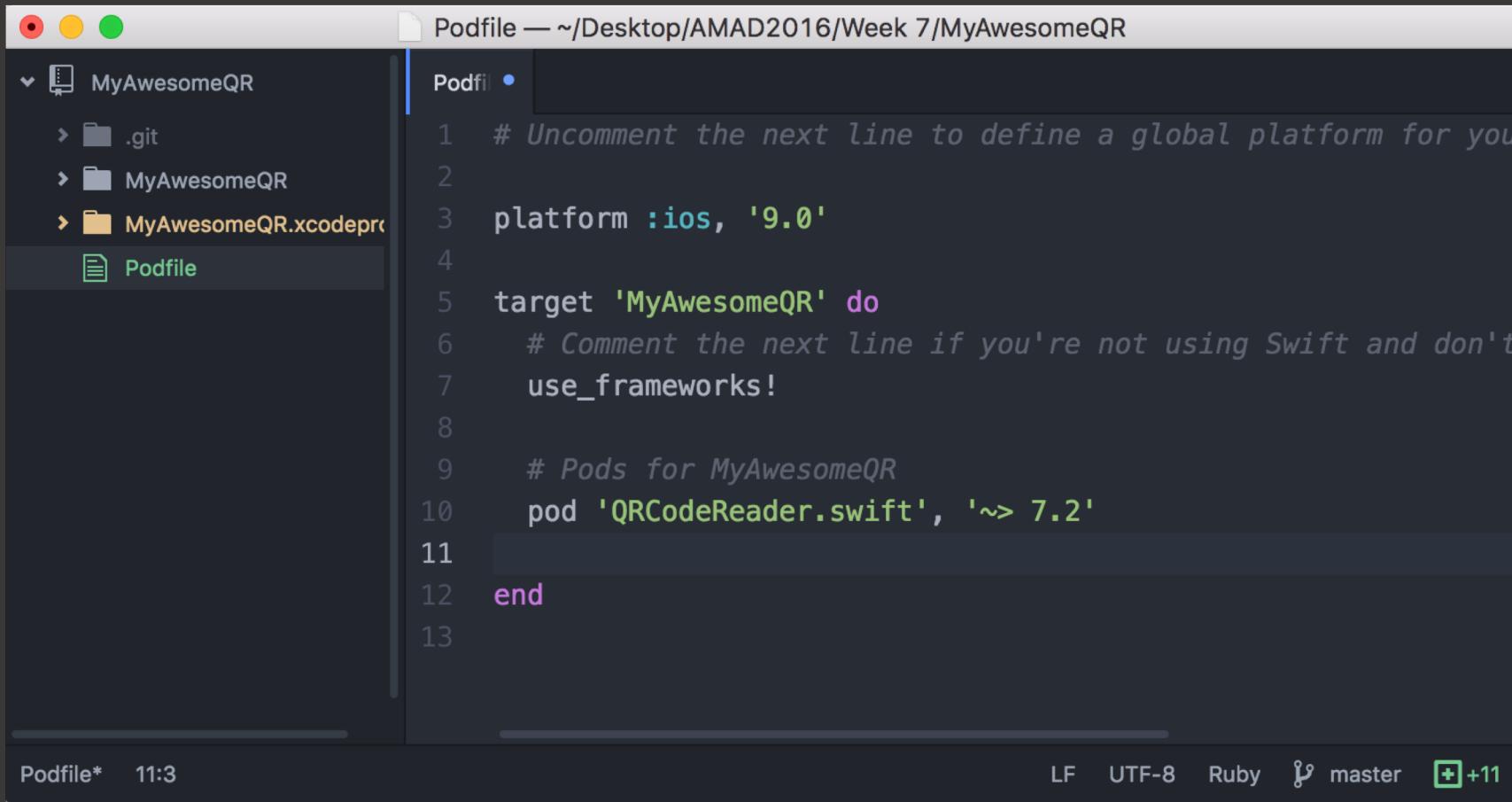
- Use of “Ruby” GEM ← A dependency manager

```
~> sudo gem update --system  
~> sudo gem install cocoapods  
~> pod setup
```

```
sudo gem install cocoapods -n /usr/local/bin
```

```
MyProject$ pod init
```

Atom or any text editor



The screenshot shows a dark-themed text editor window titled "Podfile — ~/Desktop/AMAD2016/Week 7/MyAwesomeQR". The left sidebar shows a file tree for a project named "MyAwesomeQR" containing ".git", "MyAwesomeQR", "MyAwesomeQR.xcodeproj", and "Podfile". The main editor area contains the following code:

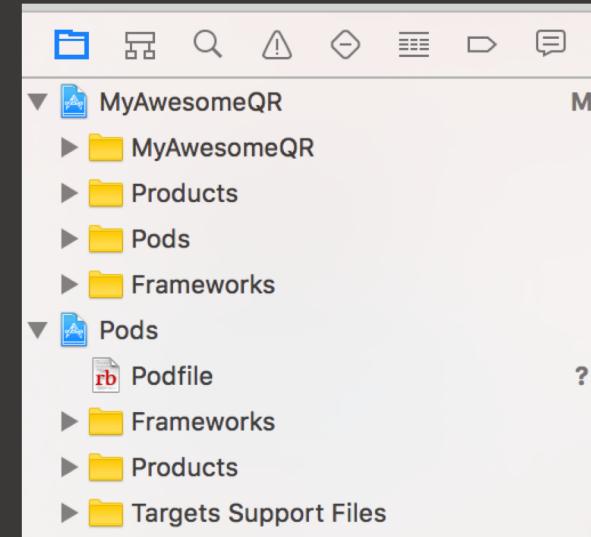
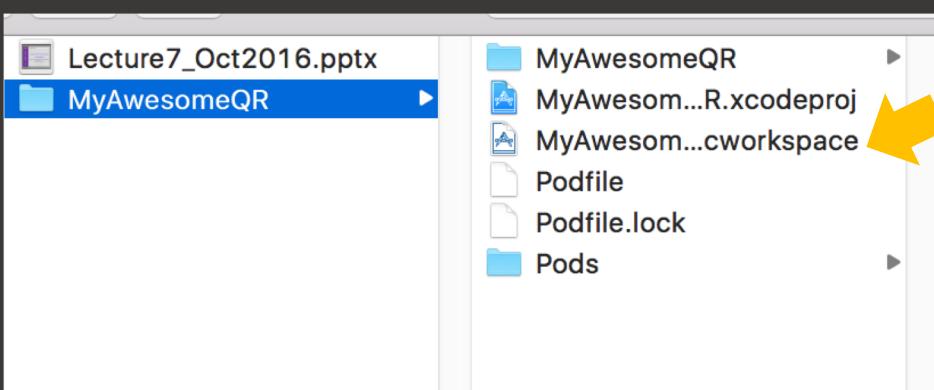
```
Podfile •
1 # Uncomment the next line to define a global platform for your
2 # entire repository
3 platform :ios, '9.0'
4
5 target 'MyAwesomeQR' do
6   # Comment the next line if you're not using Swift and don't
7   # use_frameworks!
8
9   # Pods for MyAwesomeQR
10 pod 'QRCodeReader.swift', '~> 7.2'
11
12 end
13
```

The status bar at the bottom indicates the file is "Podfile*" and has 11:3 lines. It also shows encoding as "UTF-8", language as "Ruby", and a branch "master" with 11 changes.

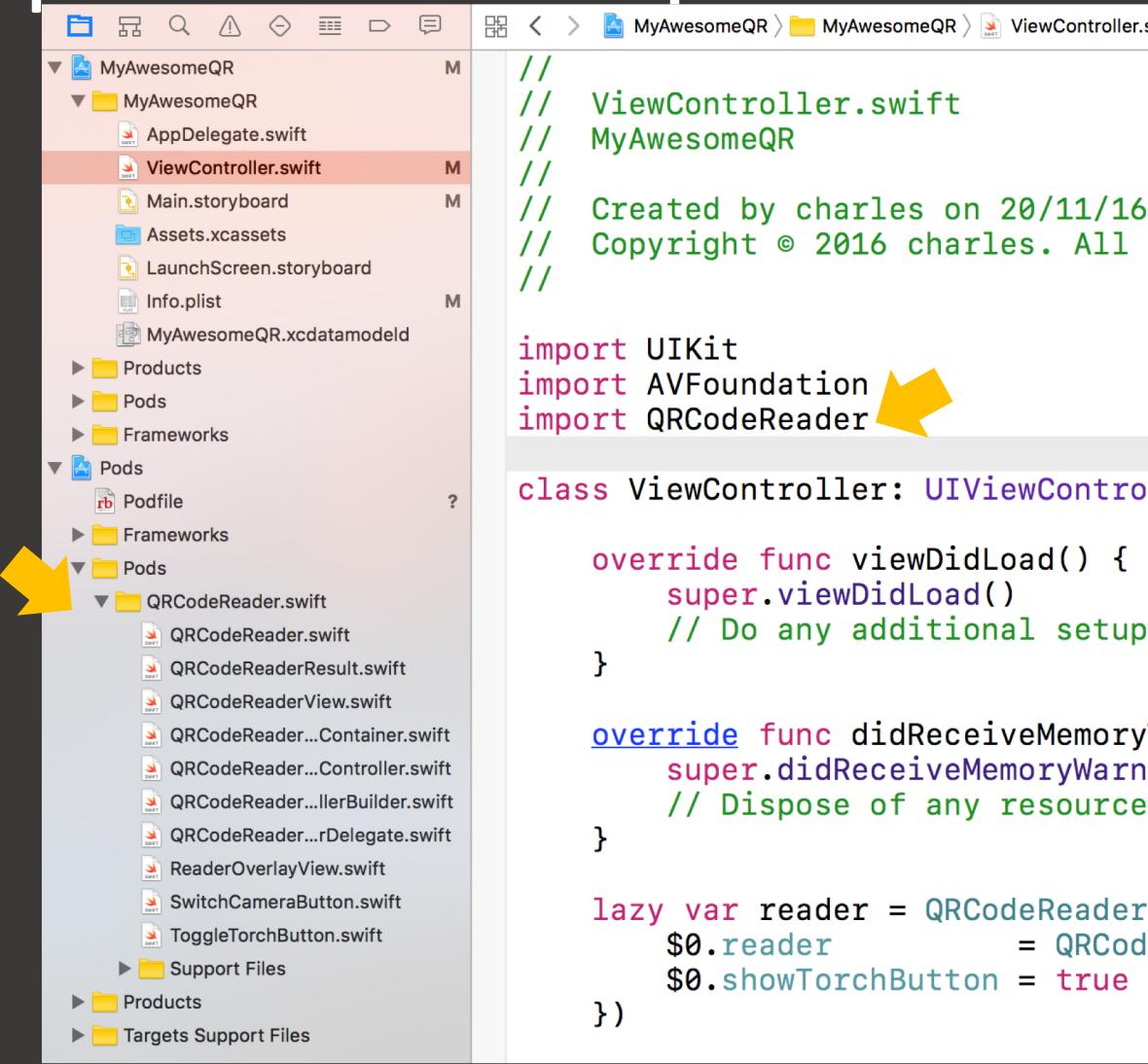
Steps

```
MyProject$ pod install
```

```
Analyzing dependencies  
Downloading dependencies  
Generating Pods project
```



Opening up the xcworkspace



The screenshot shows the Xcode interface with the project 'MyAwesomeQR' open. The left sidebar displays the file structure:

- MyAwesomeQR (target)
- MyAwesomeQR (group)
 - AppDelegate.swift
 - ViewController.swift (selected)
 - Main.storyboard
 - Assets.xcassets
 - LaunchScreen.storyboard
 - Info.plist
 - MyAwesomeQR.xcdatamodeld
- Products
- Pods
 - Podfile
 - Frameworks
 - Pods (group)
 - QRCodeReader.swift (selected)
 - QRCodeReader.swift
 - QRCodeReaderResult.swift
 - QRCodeReaderView.swift
 - QRCodeReader...Container.swift
 - QRCodeReader...Controller.swift
 - QRCodeReader...Builder.swift
 - QRCodeReader...Delegate.swift
 - ReaderOverlayView.swift
 - SwitchCameraButton.swift
 - ToggleTorchButton.swift
- Support Files
- Products
- Targets Support Files

The right pane shows the content of `ViewController.swift`:

```
// ViewController.swift
// MyAwesomeQR
//
// Created by charles on 20/11/16.
// Copyright © 2016 charles. All rights reserved.

import UIKit
import AVFoundation
import QRCodeReader // Yellow arrow points here

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup required after loading the view.
    }

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

    lazy var reader = QRCodeReaderView()
        $0.reader      = QRCodeReader()
        $0.showTorchButton = true
    }
}
```

MAD2 Oct 2018

Alternative Dependency Manager

- Carthage
 - Carthage is another simple dependency manager for the Cocoa application. It downloads and build the dependencies **but will not change Project file Or Xcode project** build setting like CocoaPods. We have to manually drag '.framework' binaries to "Linked Frameworks and Libraries".

```
$ brew install carthage
```

Carthage

- Pros
 - Carthage won't touch your Xcode settings or project files. It just download and build the dependencies so you have proper control on what you are doing.
 - Decentralised
 - Supports submodules
- Cons
 - Unstable and Slow
 - Small Community, not many contributors
 - Lot of manual steps to perform on Xcode to get everything setup

Alternative Dependency Manager

- Swift Package Manager
 - The Swift Package Manager is a tool for managing the distribution of Swift code. It's integrated with the Swift build system to automate the process of downloading, compiling, and linking dependencies.
 - It's Official Package Manager for Swift
- Swift Package Manager is
 - Command Line Based tool
 - Being Cross-Platform, Swift Package Manager doesn't need Xcode to create package
 - It's decentralised
 - Swift Package Manager is open-source and source code is available on Github

```
$ swift build -- version  
Apple Swift Package Manager – Swift 4.2.0 (swiftpm-14460.2)
```

Summary

Understand Alert View with UIAlertController

Understand the Dependancy Manager concepts, Libraries & Frameworks : CocoaPod