Praktikum – iOS-Entwicklung

Sommersemester 2018

Prof. Dr. Linnhoff-Popien

Markus Friedrich, Kyrill Schmid

Crashkurs

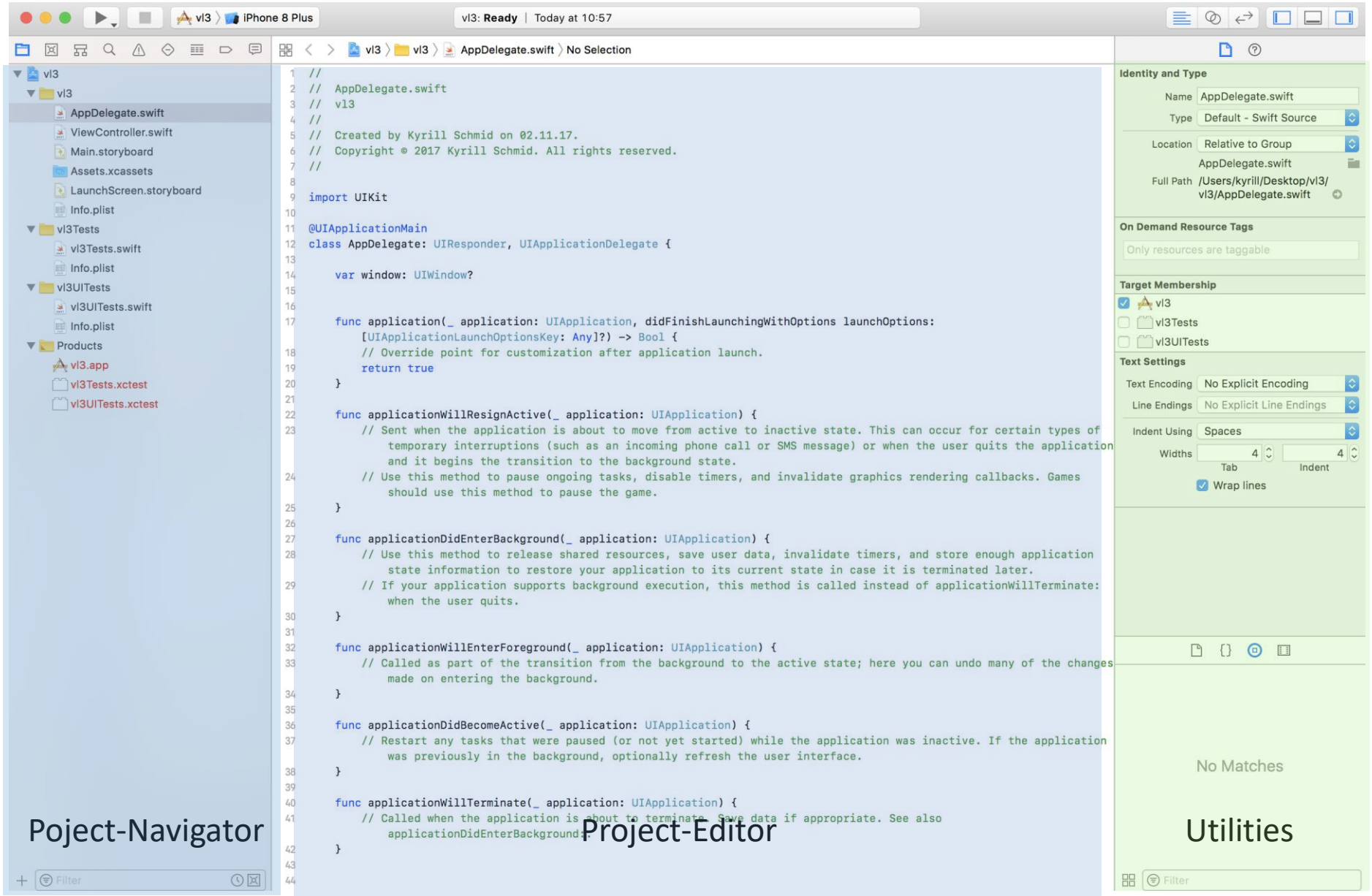
XCode, Projektgrundgerüst, App-Zustände, Views und ViewController

XCode

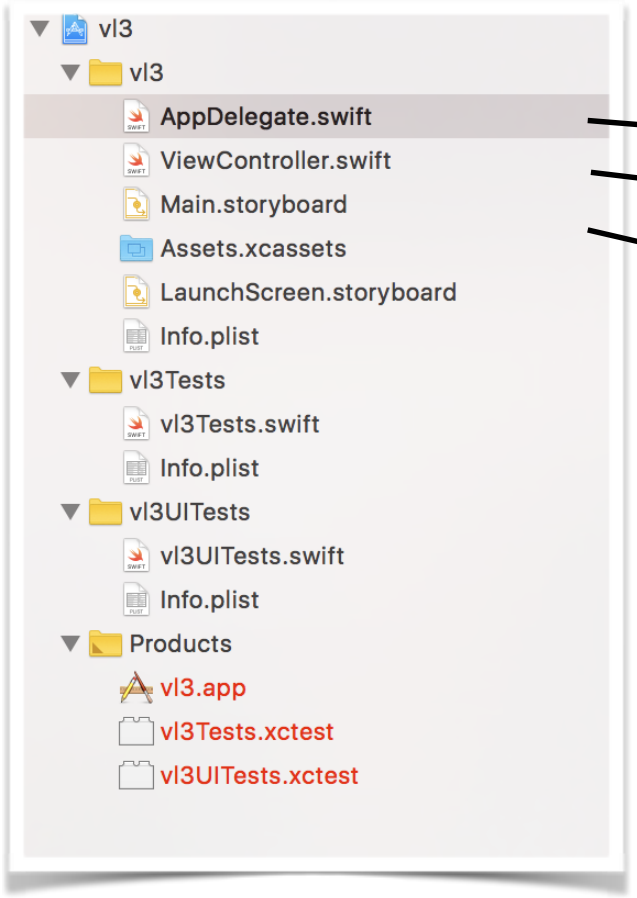
XCode beinhaltet im Grunde schon alles was man zum Entwickeln von iOS-Apps benötigt:

- Software Development Kit (SDK)
- Code Editor
- UI Editor
- Debugging Tools
- Simulator

Ein neues Projekt anlegen

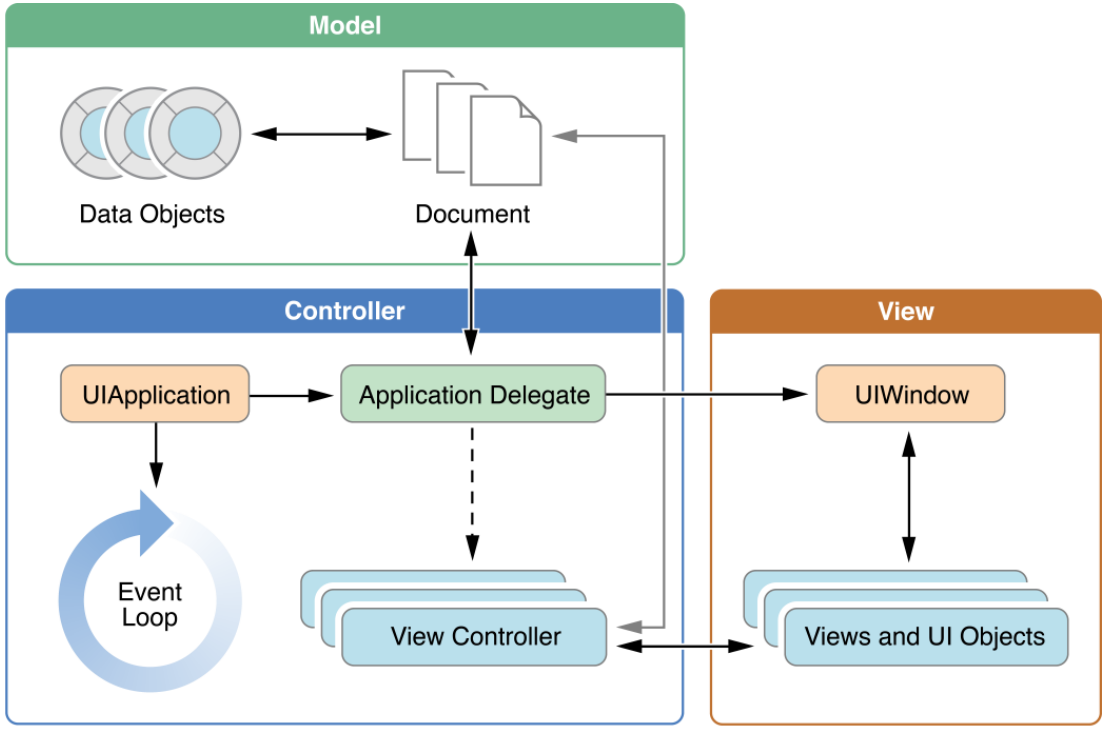

vl3 〉 iPhone 8 Plus
vl3: Ready | Today at 10:57
vl3 〉 vl3 〉 AppDelegate.swift 〉 No Selection
vl3
vl3
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
vl3Tests
vl3Tests.swift
Info.plist
vl3UITests
vl3UITests.swift
Info.plist
Products
vl3.app
vl3Tests.xctest
vl3UITests.xctest
Filter
//
// AppDelegate.swift
// vl3
//
// Created by Kyrill Schmid on 02.11.17.
// Copyright © 2017 Kyrill Schmid. All rights reserved.
//
import UIKit
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
var window: UIWindow?
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
// Override point for customization after application launch.
return true
}
func applicationWillResignActive(_ application: UIApplication) {
// Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.
// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.
}
func applicationDidEnterBackground(_ application: UIApplication) {
// Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.
// If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.
}
func applicationWillEnterForeground(_ application: UIApplication) {
// Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.
}
func applicationDidBecomeActive(_ application: UIApplication) {
// Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.
}
func applicationWillTerminate(_ application: UIApplication) {
// Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:
}
Identity and Type
Name AppDelegate.swift
Type Default - Swift Source
Location Relative to Group
AppDelegate.swift
Full Path /Users/kyrill/Desktop/vl3/vl3/AppDelegate.swift
On Demand Resource Tags
Only resources are taggable
Target Membership
vl3
vl3Tests
vl3UITests
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 Tab 4 Indent
Wrap lines
No Matches
Filter
Poject-Navigator
Project-Editor
Utilities

Neues Projekt: Single View Application

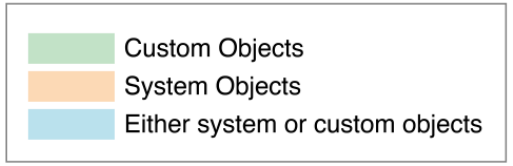


- XCode generiert beim anlegen eines neuen Projekts initiale Files:
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - …

Bestandteile einer iOS-App



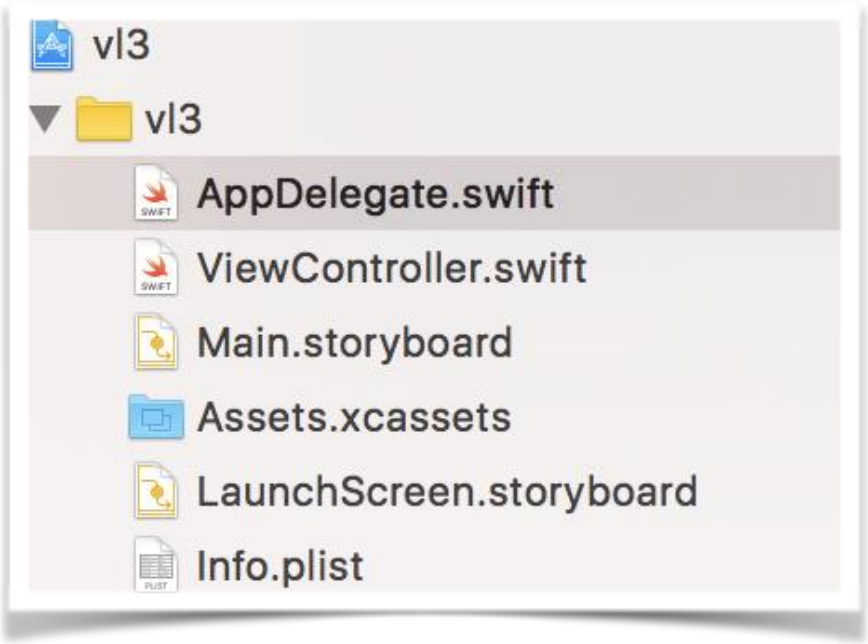
https://developer.apple.com



Mehr zum
Model-View-Controller Pattern in Kürze.

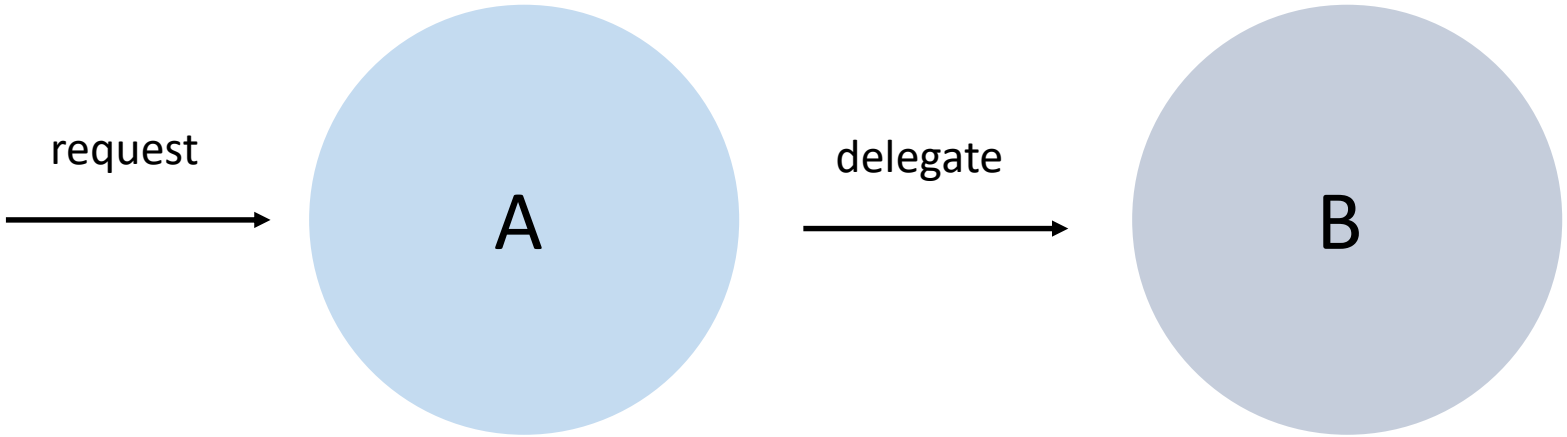
Der AppDelegate

- Der AppDelegate ist ein **Singleton** und das erste customisierbare Objekt einer iOS App.
- AppDelegate ist „Ansprechpartner“ des Betriebssystems für einige wichtige Events.
- AppDelegate ist der **Delegate** der Application.
- Das **Delegation Pattern** ist zentral in iOS und findet an vielen Stellen Verwendung.



Einschub: Das Delegation Pattern

- Delegation ist ein Pattern der Objekt-Orientierten Programmierung um Objekt-Komposition zu unterstützen.
- Bei Delegation behandelt ein Objekt A eine Anfrage indem es an ein weiteres Objekt B delegiert.



- In Swift wird Delegation durch die Verwendung von Protokollen unterstützt.

Protokolle in Swift

- Protokolle sind ähnlich zu Interfaces in z.B. Java.
- Eine Klasse (oder Struktur) kann Protokoll konform sein d.h. die Anforderungen des Protokolls werden umgesetzt.

```
protocol SomeProtocol {
    // protocol definition goes here
    func requiredFunction()
}

class SomeClass: SomeProtocol{
    func requiredFunction() {
        print("Have to do that")
    }
}
```

Protokolle und Delegation

```
protocol Assistant {
    func writeEmail()
}
```

Protokolle und Delegation

```
protocol Assistant {
    func writeEmail()
}

class WorkerA : Assistant {
    func writeEmail() {
        print("WorkerA: I wrote an Email!")
    }
}
```

Protokolle und Delegation

```
protocol Assistant {
    func writeEmail()
}

class WorkerA : Assistant {
    func writeEmail() {
        print("WorkerA: I wrote an Email!")
    }
}

class Boss {
    var delegate : Assistant?

    func work(){
        print("Boss: I have no time to write Emails!“)
        delegate?.writeEmail()
    }
}
```

Protokolle und Delegation

```
let boss = Boss()
let worker = WorkerA()
boss.delegate = worker
boss.work()

// Boss: I have no time to write Emails!
// WorkerA: I wrote an Email!
```

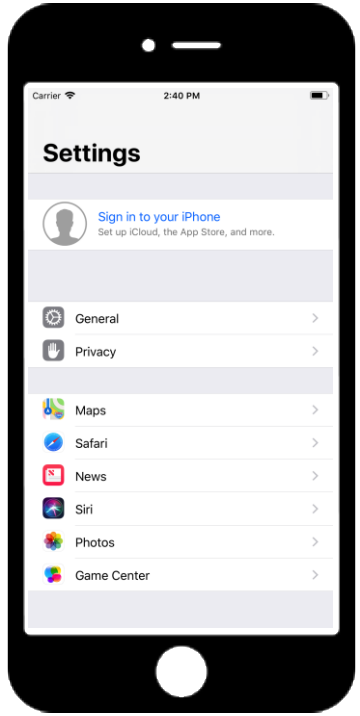
AppDelegate als Delegate der Application

- AppDelegate ist der **Delegate** der Application.
- Das bedeutet konkret, dass er **UIApplicationDelegate** konform ist.
- Eine wichtige Verantwortlichkeit, die der AppDelegate übernimmt ist das Reagieren auf Zustands-Transitionen der App**.**

App Zustands-Transition

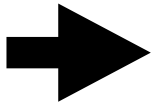
- Wenn die App in den Zustand ‚Background' übergeht müssen u.U. noch…
  - …Daten gesichert werden.
  - …Speicher freigegeben werden (reduce memory footprint).
- Die letzte Chance dazu ergibt sich in der **applicationDidEnterBackground(_:)** Methode des AppDelegate

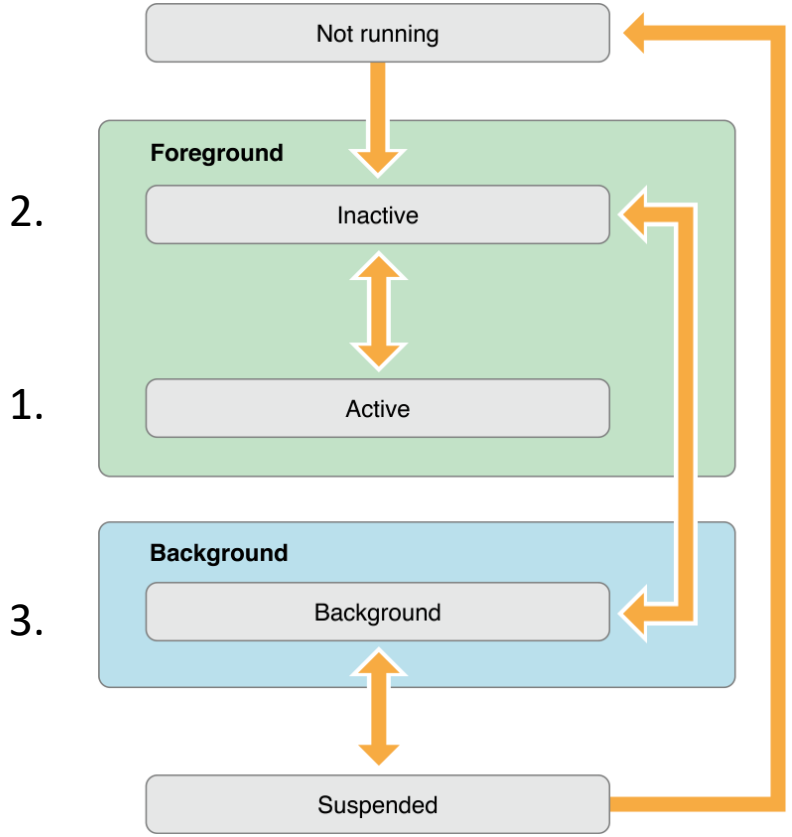
Home Button Press



App Zustands-Transition

User drückt Home Button





App Zustands-Transition

Weitere Methoden des AppDelegate:

applicationWillTerminate(_:)

application(_:willFinishLaunchingWithOptions:)

applicationDidBecomeActive(_:)

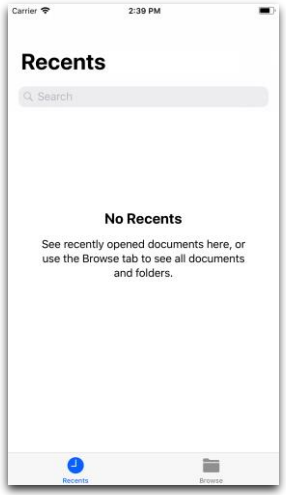
applicationWillResignActive(_:)

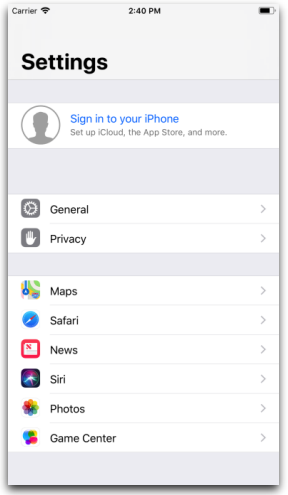
applicationWillEnterForeground(_:)

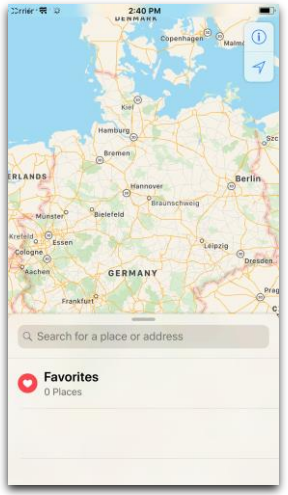
application(_:didFinishLaunchingWithOptions:)

Views

- Eine App besteht aus einer Menge von **Views** durch die der Nutzer navigieren kann.

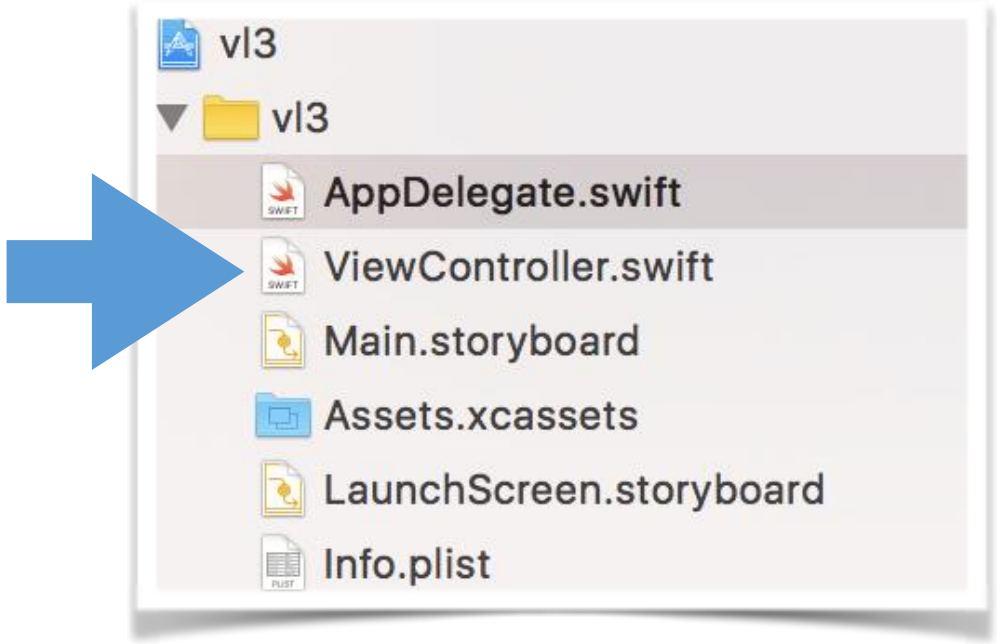






- **Views** zeigen aber lediglich Daten an.
  - Daten kommen aus Modell und werden vom **ViewController** übermittelt.
  - Benutzerinteraktion wird vom **ViewController** interpretiert und an das **Model** übermittelt.

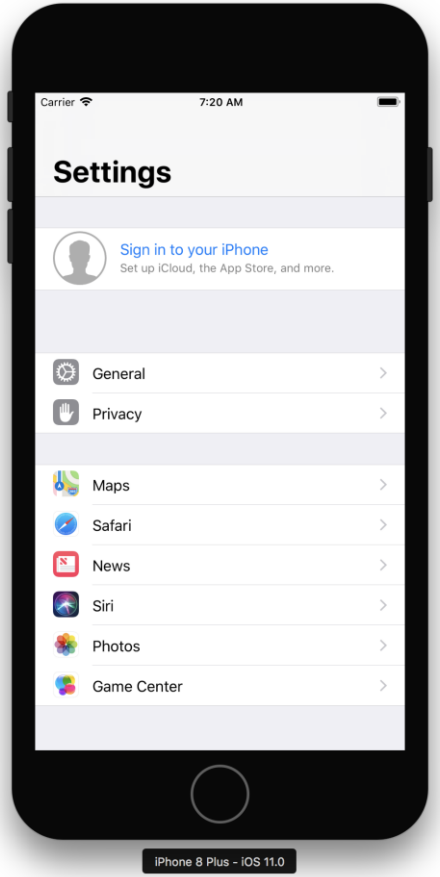
ViewController

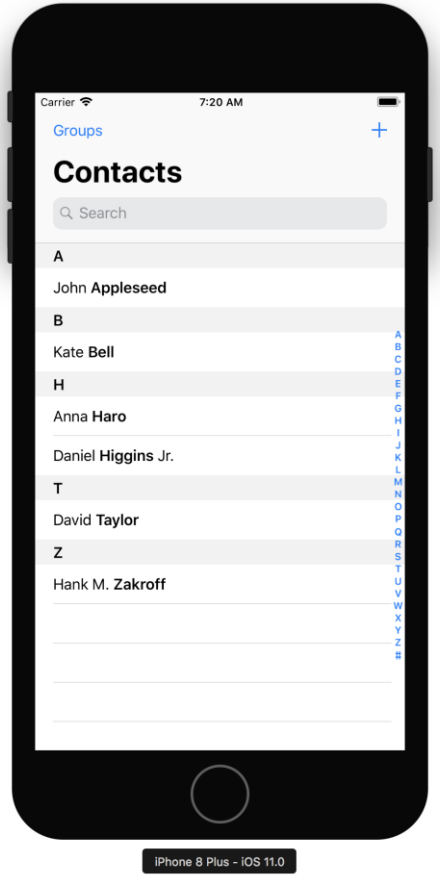


- Jede **View** hat einen zugehörigen **Controller** -> **ViewController**
- ViewController sind zentrale Objekte für die iOS-Programmierung und übernehmen viele wichtige Aufgaben.
- Geben eine bewährte Struktur vor für die Gestaltung eines Interface.
- Viele verschiedene ViewController Typen, die sofort verwendet werden können (NavigationView, TableView, CollectionView, etc.).

Beispiel: TableView

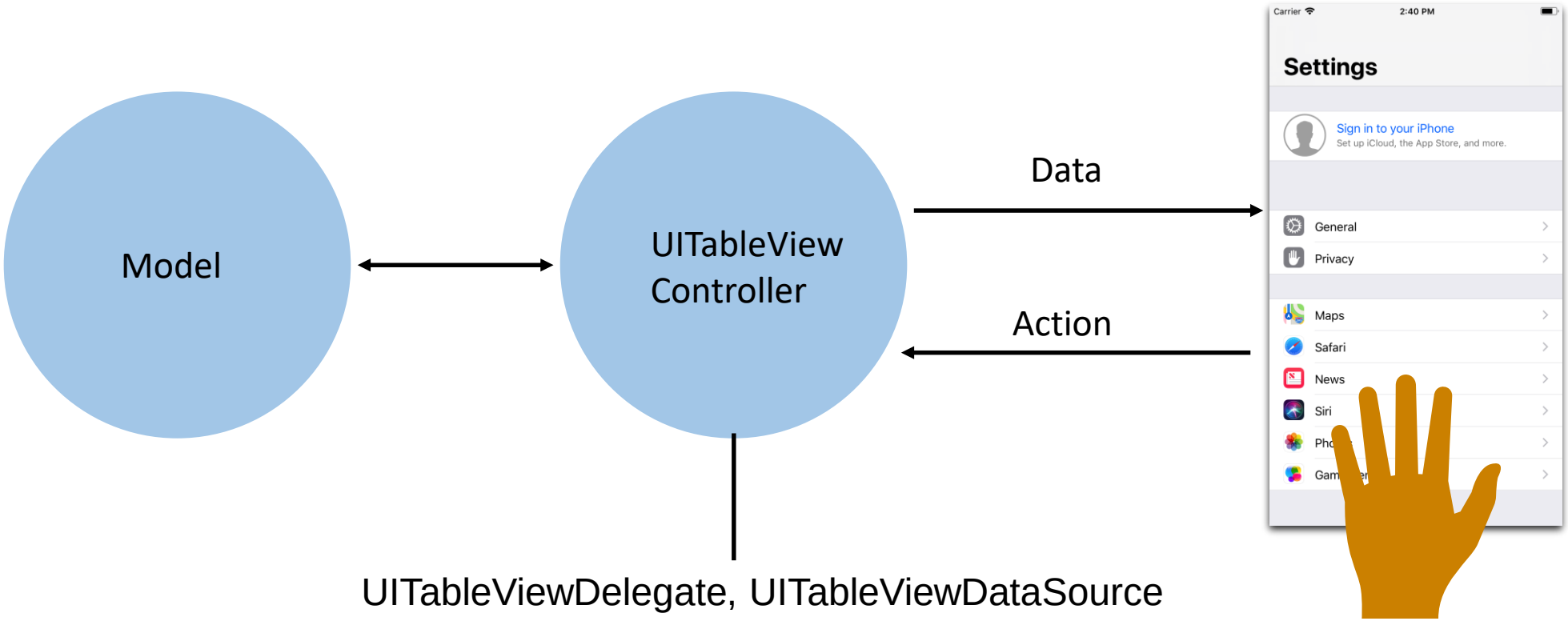
- Oft bietet sich eine Tabellen-Artige Darstellung von Daten an z.B. für Kontakte, Einstellungen, …
- Dafür gibt es einen vordefinierten Controller: **UITableViewController**





TableViewController

Der **UITableViewController** muss also Daten liefern und auf Benutzerinteraktion reagieren können.



TableViewController Beispiel

Daten liegen in Form eines Arrays vor (Namen).

- Wir wollen Namen aus der Liste löschen
- Liste umsortieren