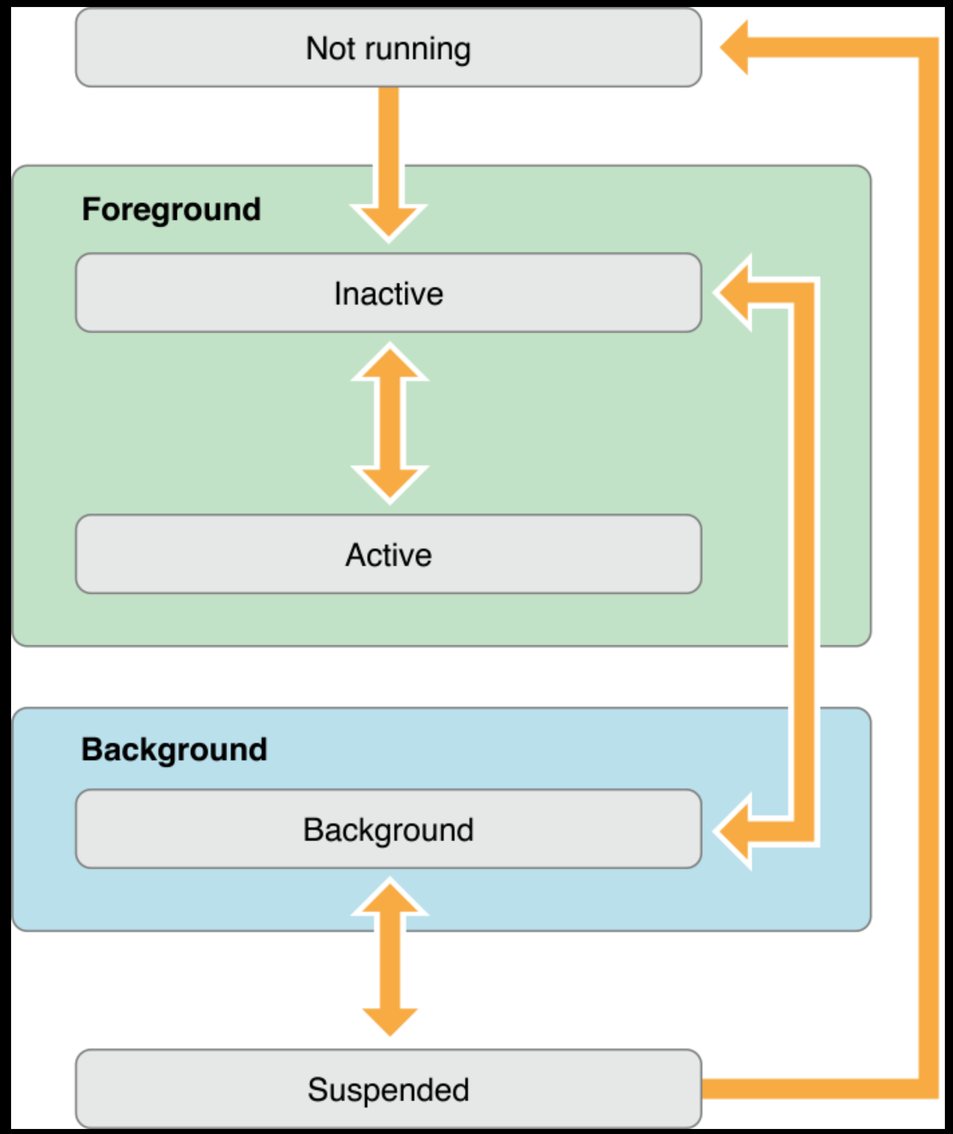
iOS Life Cycle, ARC, MVC

Session 103

Alex Telek
Shazam iOS Engineer

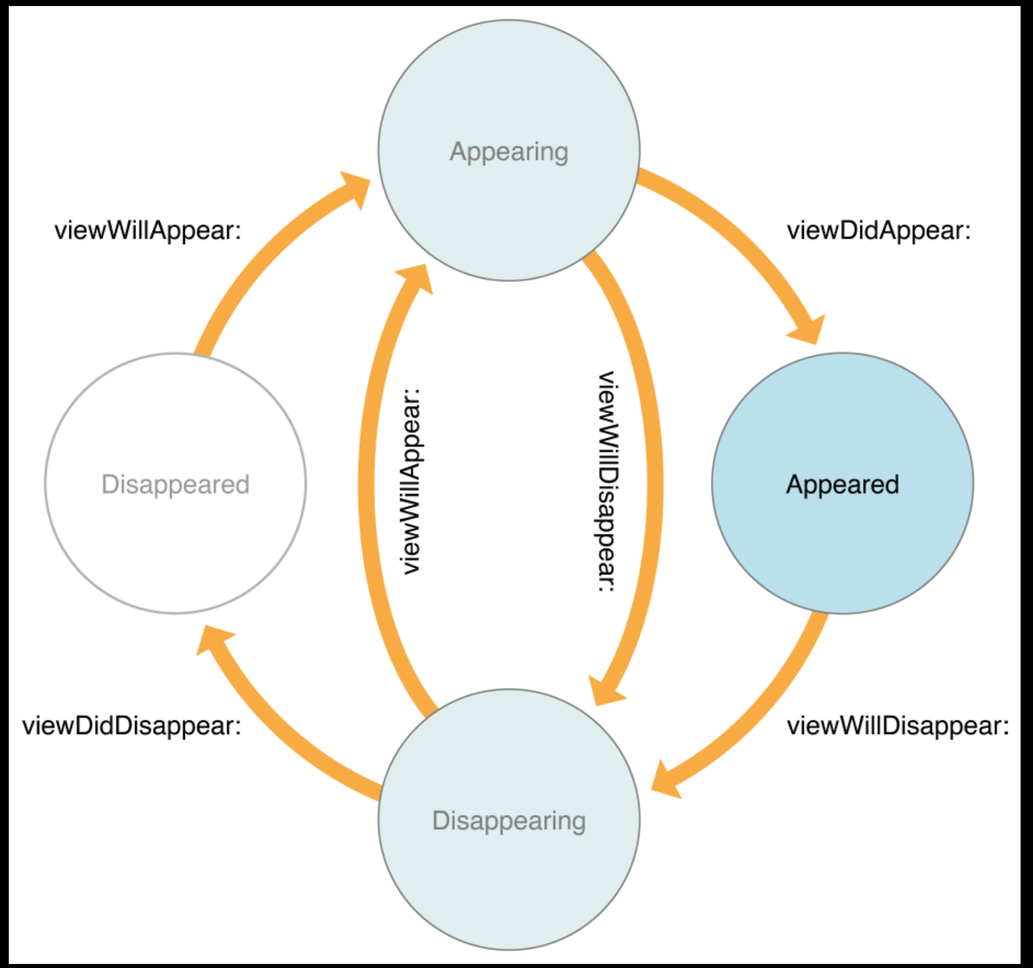
App States

- application:willFinishLaunchingWithOptions:
- application:didFinishLaunchingWithOptions:
- applicationDidBecomeActive:
- applicationDidEnterBackground:
- applicationWillEnterForeground:
- applicationWillTerminate:



View states

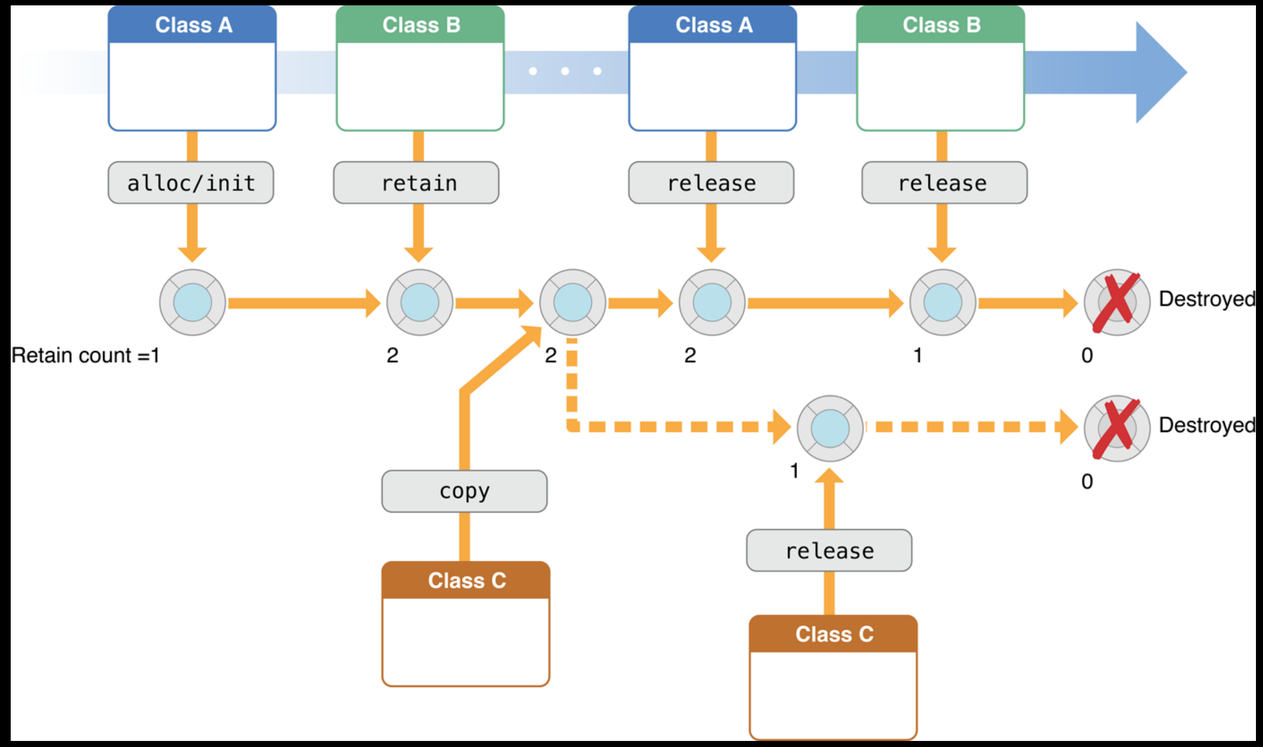
- viewDidLoad:
- viewWillAppear:
- viewDidAppear:
- viewWillDisappear:
- viewDidDisappear:



ARC

Automatic Reference Counting

Memory management



Strong and Weak

Strong variables increase the reference.

Weak variable do not increase the reference.

Retain Cycle and Memory Leaks

A has multiple instances of an entity B and each B entity is associated with an A entity.

```
class Book: {

    var pages = [Page]()
}
```

```
class Page: {

    var book: Book
}
```

Protocols

A protocol defines a blueprint of methods, properties, and other requirements that suit a particular task or piece of functionality. It can be adopted by a class, structure or enumeration.

```
protocol RandomNumberGenerator {

    var lastRandomNumber: Double { get }

    func random() -> Double

    mutating func changeSomething()
}
```

Protocols

```
class Game: RandomNumberGenerator, OtherProtocol {

    var lastRandomNumber: Double = 0.0

    func random() -> Double { … }

    mutating func changeSomething() { … }
}
```

Delegation

Delegation is a design patter that enables a class or structure to hand off some of its responsibilities to an instance of another type.

```
protocol DiceGame {

    func play()
}

protocol DiceGameDelegate {

    func didStartGame(_ game: DiceGame)
    func didEndGame(_ game DiceGame)
}
```

Delegation

```
class SnakesAndLadders: DiceGame {

    weak var delegate: DiceGameDelegate?

    func play() {

        delegate?.gameDidStart(self)

        // Play game

        delegate?.gameDidEnd(self)
    }
}
```

```
class DiceGameTracker: DiceGameDelegate {

    myGame.delegate = self

    func didStartGame(_ game: DiceGame) { … }
    func didEndGame(_ game: DiceGame) { … }
}
```

MVC

A pattern of app design in which view controllers serve as the communication pipeline between views and the data model.

Video

Stanford CS193P Fall 2017 -18

KCL TECH SOCIETY