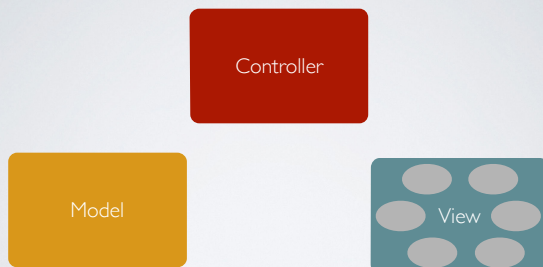# IOS ESSENTIALS

Lecture 04
Jonathan R. Engelsma, Ph.D.

TOPICS

- Model / View / Controller

- Understanding ViewControllers and their life cycle.
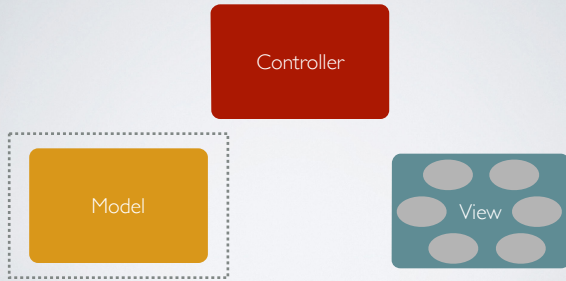
- Overview of commonly used iOS ViewControllers
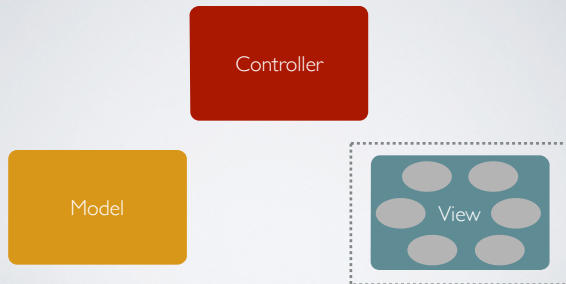

MODEL/VIEW/CONTROLLER

Controller

Model

View

Our iOS code is broken up into 3 different components
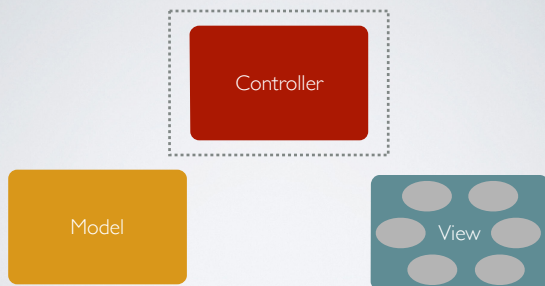(Inspired by Paul Hegarty's very excellent MVC lecture.)

MODEL/VIEW/CONTROLLER

Controller

Model

View

The **MODEL** is the data our app produces/consumes.

MODEL/VIEW/CONTROLLER

Controller

Model

View

The **VIEW** is what the user sees and interacts with on screen.

MODEL/VIEW/CONTROLLER

Controller

Model

View

The **CONTROLLER** *manages* the model and the view.

MODEL/VIEW/CONTROLLER

Controller
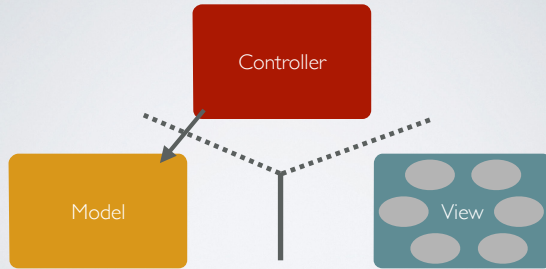
Model

View

Controller can manipulate the model directly.

MODEL/VIEW/CONTROLLER

Controller

Takes place via an "outlet"
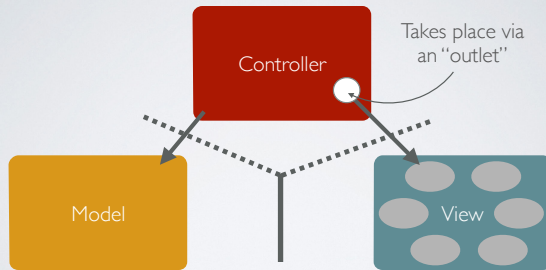
Model

View

Controller can manipulate the view directly.

MODEL/VIEW/CONTROLLER

Controller
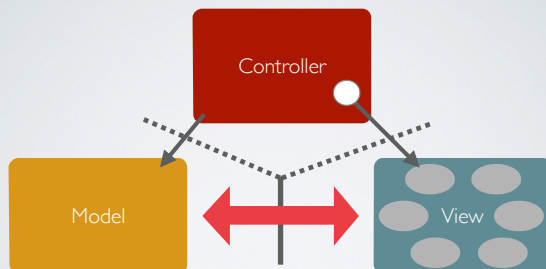
Model

View

Model / View should **NEVER** talk directly to each other!

MODEL/VIEW/CONTROLLER

Controller defines an *action*.

Controller

View is wired up to call action.

Model

View

Views can sort of talk to controller, but only in a very controlled manner!

MODEL/VIEW/CONTROLLER

Controller

Delegate

Data Source

Model

View

Controller serves as the View's delegate and data source via Protocols
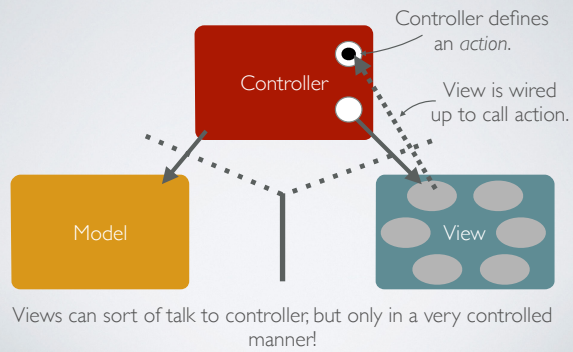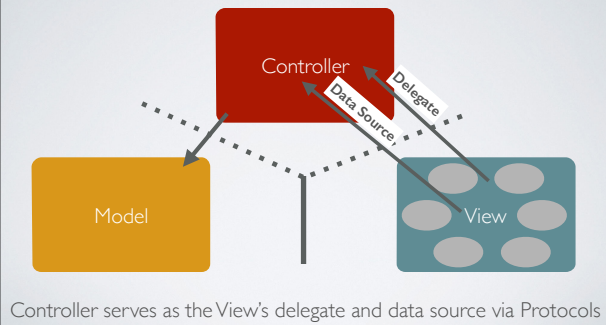
GREETING DEMO

DEMO!!

VIEW CONTROLLERS

- View controllers manage a screen's view hierarchy and the segues between view controllers.

- The "glue" between views and models in MVC.

- View controllers:

  - initialize and setup models

  - populate the view hierarchy with views

  - coordinate with the view hierarchy (delegate)

  - format data for the views (data source)

VIEW CONTROLLER LIFECYCLE

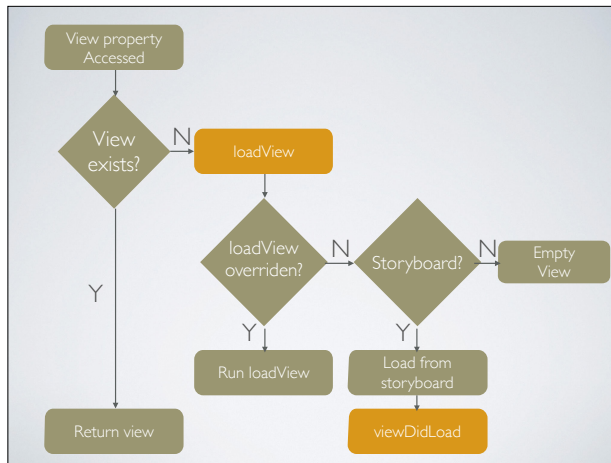- Recall the methods generated when we created our first app:

```swift
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    self.configureView()
}

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

- These are methods that help us coordinate with the controller's lifecycle.

VIEW CONTROLLER LIFE CYCLE

- Understanding the view controller life cycle allows us to properly manage the controller's models and views.

- Whenever a view controller's view property is accessed a chain of events is triggered.

https://developer.apple.com/library/IOS/featuredarticles/ViewControllerPGforiPhoneOS/ViewLoadingandUnloading/ViewLoadingandUnloading.html

Flowchart:
- View property Accessed
- View exists? — N → loadView
- View exists? — Y → Return view
- loadView overriden? — N → Storyboard?
- loadView overriden? — Y → Run loadView
- Storyboard? — N → Empty View
- Storyboard? — Y → Load from storyboard
- viewDidLoad

VIEW CONTROLLER METHODS

Method Name	Description
viewDidLoad	View has finished loading.
viewWillAppear	View is about to appear.
viewDidAppear	View just appeared.
viewWillDisappear	View is about to disappear.
viewDidDisappear	View just disappeared
didRecieveMemoryWarning	Low memory conditions detected.

UITabBarController   UICollectionViewController   UITableViewController

UINavigationController

READING ASSIGNMENT

- Chapter 6: Programming iOS8 (Neuburg)

LIFECYCLE & STOPWATCH DEMOS

DEMO!!