

MOBILE DEVELOPMENT

OBJECT ORIENTED PROGRAMMING, CONT.

Rudd Taylor
Founder, SALT

OO PROGRAMMING

LEARNING OBJECTIVES

- Demonstrate common UI elements and hook the IB elements to code
- Identify how to access documentation for IB elements and controllers
- Recognize and implement different view controllers
- Present views pragmatically
- Identify arrays in Swift
- Introduce gestures
- Identify and implement methods associated with view controllers and UI elements inherited from their superclasses

OO PROGRAMMING

REVIEWING OBJECT ORIENTED PROGRAMMING

GETTING STARTED

INTRO TO FUNCTIONS

OO PROGRAMMING

WHAT IS A CLASS?

- A class is a bundle of state and behavior, defines the structure of an instance
- Inherits state and behavior from its parent class (superclass)
- Examples of classes:
 - UIViewController
 - UIView
 - UILabel
 - UITextField

OO PROGRAMMING

WHAT IS OO PROGRAMMING?

- Break up functions and variables into logical bundles of state and behavior
 - e.g. UIView, UIViewController, UIButton, etc are all classes
 - Technical term: *encapsulation*
- Use parent/child relationships between these bundles to share behavior. Children inherit state and methods from their parents
 - e.g. UITextField *is a* UIView
 - Technical term: *inheritance*
- If something operates on a class, it can also operate on any of its subclasses
 - e.g. We can add subviews to a view using addSubview, which takes a UIView. We can pass a UIButton, a subclass of UIView, into that function
 - Technical term: *polymorphism*

OO PROGRAMMING

EXERCISE

- In the same pairs as the last class, take your Player classes and create an app around them
- Two screens
 - Welcome screen has explanation text, 'go to match' button
 - Match screen has a 'result' label and a 'start match' button
 - Result label prints out results of match

OO PROGRAMMING

VIEW CONTROLLERS

GETTING STARTED

INTRO TO FUNCTIONS

OO PROGRAMMING

VIEW CONTROLLER REVIEW

- UIViewControllers are fundamental building blocks of our apps
- They can contain other UIViewControllers
 - e.g. UINavigationController ***is a*** UIViewController, and it contains other UIViewControllers
- They have one

OO PROGRAMMING

VIEW REVIEW

- UIViews can contain other subviews
 - ...which are also UIViews
- UIViews have lots of properties we can change
 - Color
 - Transparency
 - Position
 - ...and lots more

OO PROGRAMMING

**DEMO: PRESENTING AND
DISMISSING VIEW
CONTROLLERS**

GETTING STARTED

INTRO TO FUNCTIONS

OO PROGRAMMING

SHOW VIEW CONTROLLERS WITH LOGIC

- Create a segue between two controllers
- Give it an identifier
- At some point in your view controller, call `performSegueWithIdentifier` using that identifier

OO PROGRAMMING

EXERCISE

- Add a 'good won' screen and a 'bad won' screen
- Display 'good won' screen as a modal if good wins
- Display 'bad won' screen as a modal if bad wins
- Both modals should have a 'dismiss' button which dismisses the modal

OO PROGRAMMING

GESTURES AND CONTROLS

OO PROGRAMMING

GESTURES

- Tap
- Swipe
- Pinch
- Pan
- Edge pan
- Long press
- Rotate
- Can add in interface builder or code

OO PROGRAMMING

CONTROLS

- UISlider, UISwitch, UIButton, UITextField and lots more

OO PROGRAMMING

CONTROLS & GESTURES DEMO

OO PROGRAMMING

EXERCISE

- Add a 'allows supermoves' switch to the match screen. If enabled, players should use a new 'super attack' function
- When the user pinches on the good player or bad player screens, they should be dismissed
 - *note: This is not a standard gesture in the slightest*

TYING IB TO CLASSES

ARRAYS AND TABLE VIEWS

OO PROGRAMMING

TABLE VIEWS

- Table views are a one dimensional list
- Vocabulary:
 - Section: All table views contain multiple sections
 - Row: Every section has a number of rows, which are entries in that section
 - Index path: The combination of a section and row that is a unique entry in a table view
 - Cell: The view that is displayed for an index path (the class UITableViewCell is a subclass of UIView)
- Table views must have a number of sections, a number of cells in each section, and (optionally), the cells themselves
- Table views have a ***data source*** and a ***delegate***
 - Data source: Provides cells, number of cells and sections
 - Delegate: Gets called when things happen to the table view, provides some views (e.g. header and footer)

OO PROGRAMMING

PROTOCOLS

- A group of methods that a class has, encapsulated into its own entity
 - Methods can be required or optional
- Classes can 'meet' as many interfaces as they'd like

OO PROGRAMMING

ARRAYS

- How do we store multiple values in a list?
 - An Array
- Syntax
 - `var array = [1,2,3,4]` // The type of the array is `[Int]`
 - `var array: [String] = []` // You must declare a type for an empty array
 - `array.append("Rudd")` // Insert elements with `append()`
 - `println(array.count)` // count is the number of elements in the array

TYING IB TO CLASSES

**TYING IT TOGETHER WITH
TABLE VIEWS: DEMO**

OO PROGRAMMING

EXERCISE

- Add a name property to your players
- Add a table view to the 'match' screen
- Use the table view to display the names of the players that are about to fight