EE443 / EE593
Mobile Application Development

Prof. Phillip De Leon

Lecture 3:
Chapter 1 Introduction
Chapter 2 The One-Button App
Chapter 3 Slider and Labels

Prof. Phillip De Leon

Outline

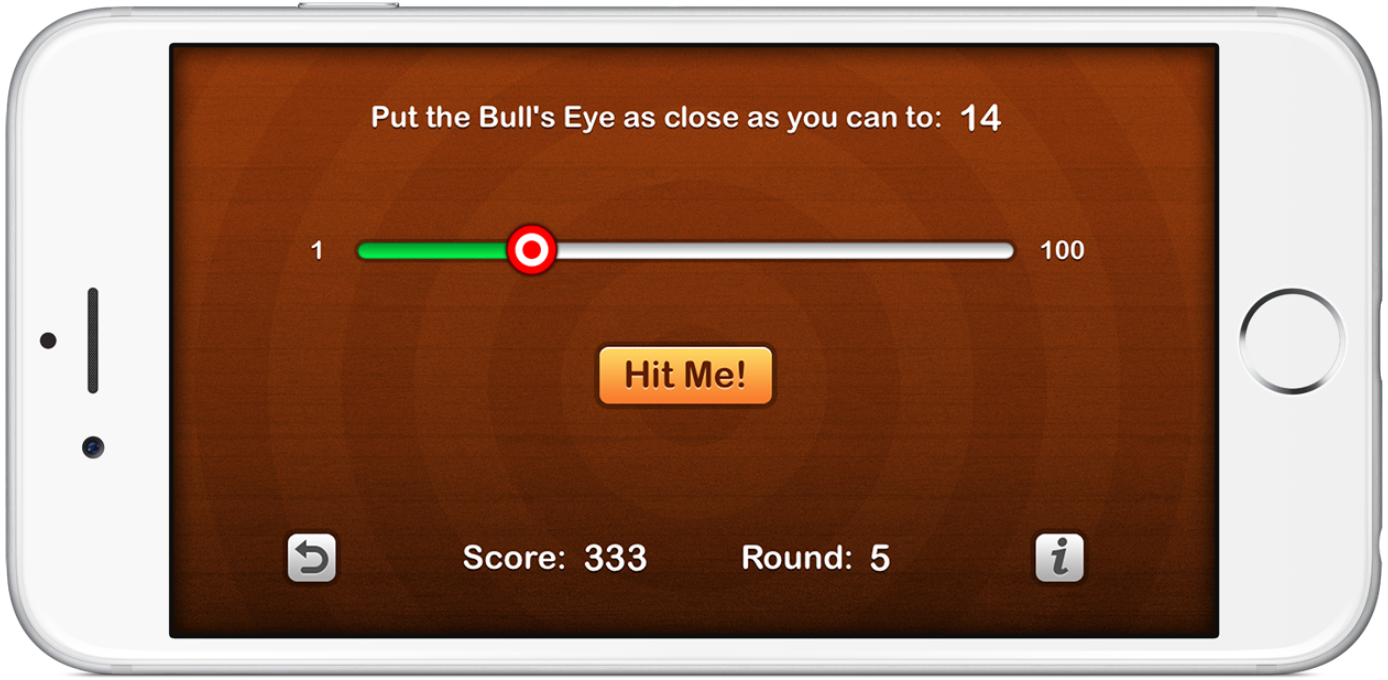
- Chapter 1: Introduction
- Chapter 2: The One-Button App
- Chapter 3: Slider and Labels

iOS Apprentice Apps

- *iOS Apprentice* is spread across four apps, moving from beginning to intermediate topics—we will learn by typing in the code

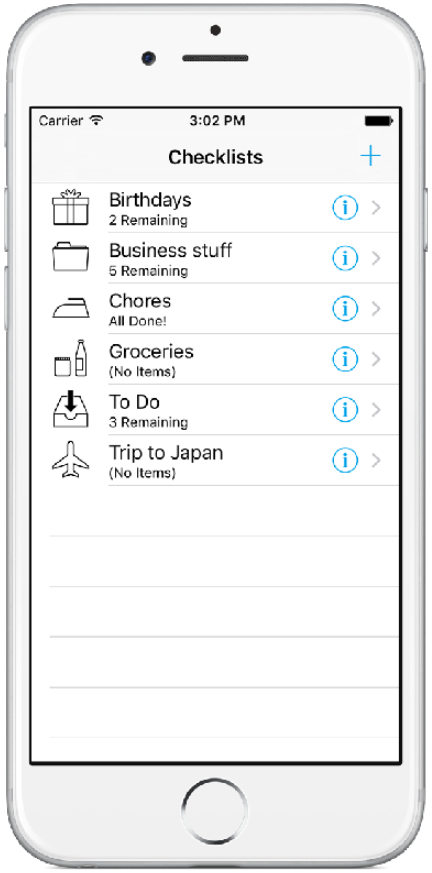
App 1: Bull's Eye (Section 1 Chapter 1 - 8)
App 2: Checklists (Section 2 Chapter 9 - 20)
App 3: MyLocations (Section 3 Chapter 21 - 31)
App 4: StoreSearch (Section 4 Chapter 32 - 40)

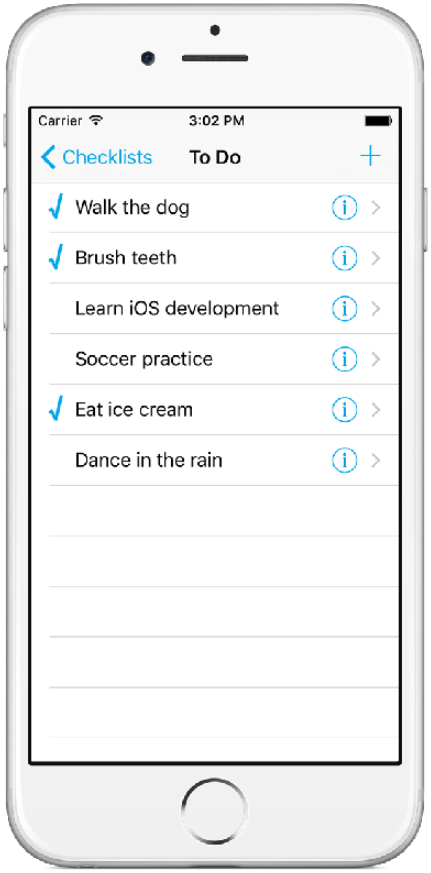
App 1: Bull's Eye

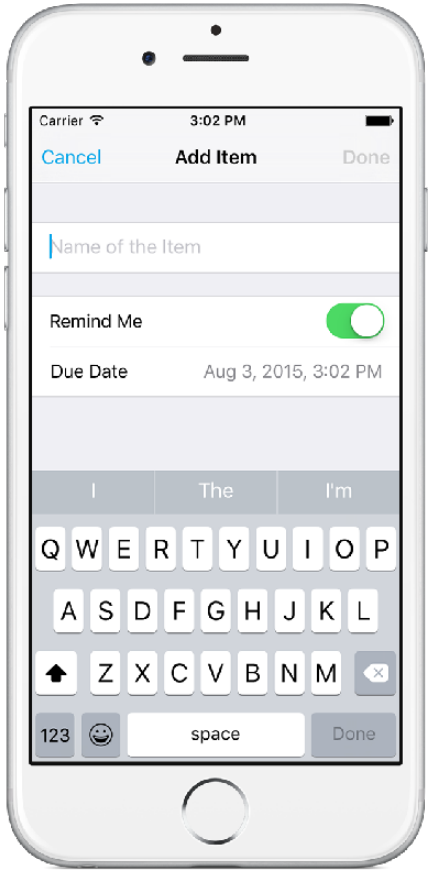


- Xcode
- Interface Builder (IB)
- Swift

App 2: Checklists

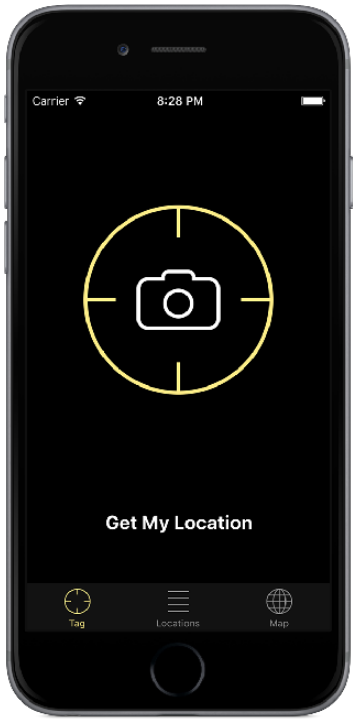


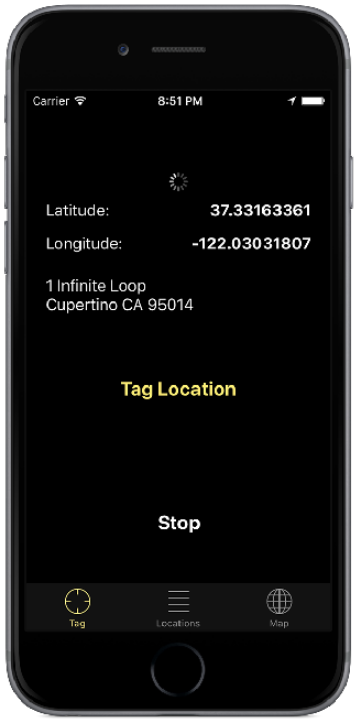


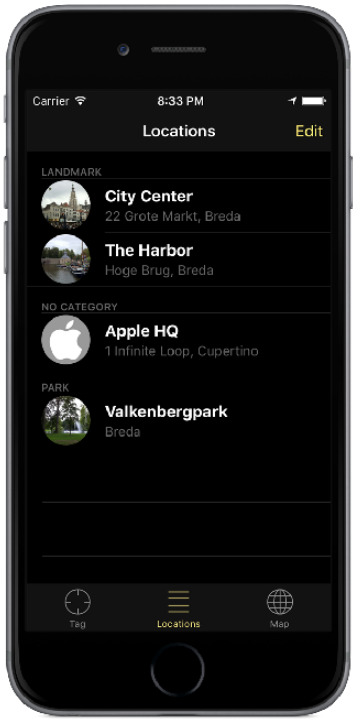


- Design patterns
- Table views
- Navigation controllers
- Delegates

App 3: MyLocations

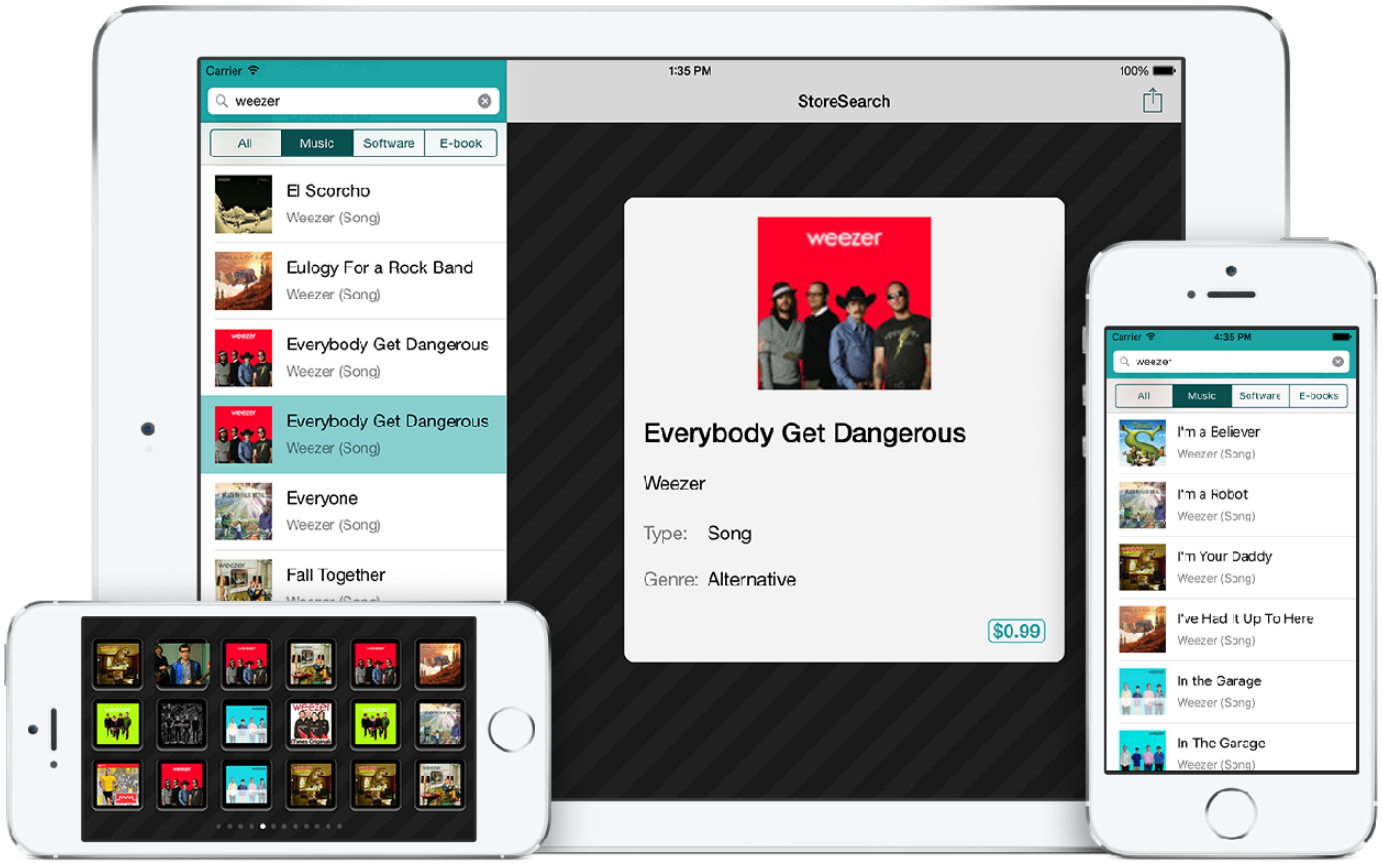






- TabBarController
- CoreLocation
- MapKit
- CoreData

App 4: StoreSearch



- Networking
- HTTP requests
- JSON

Outline

- Chapter 1: Introduction
- Chapter 2: The One-Button App
- Chapter 3: Slider and Labels

The One-Button App

- Demo: Bull's Eye app
- Demo: Bull's Eye app at the end of Chapter 2

Programming To-Do List

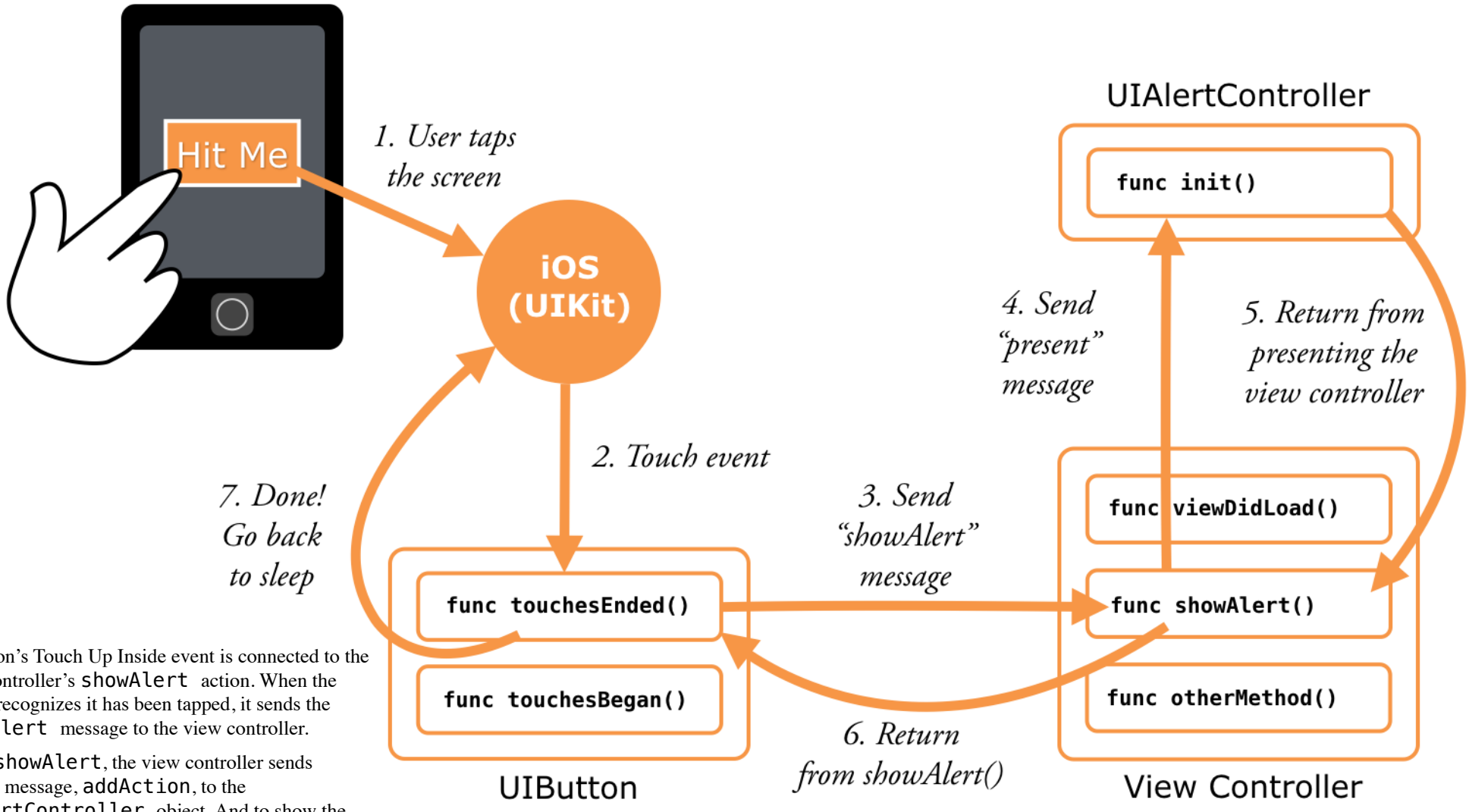
- (Ch 2) The app needs to put the "Hit Me!" button on the screen and show an alert popup when the user presses it.

Storyboard and ViewController

- The Main.storyboard and ViewController.swift files together form the design and implementation of a view controller
- The job of a view controller is to manage a single screen in your app
- The keyword @IBAction means that this method can be connected to a control in Interface Builder

The Anatomy of an App

- An app is made up of objects that send messages to each other
  - Many of the objects in your app are provided by iOS, e.g. UIButton, UIAlertController, UIAlertAction
  - Some objects you will have to program yourself, such as the view controller
  - When we tap the Hit Me! button, the UIButton object sends a showAlert() message to the view controller
  - Inside showAlert(), the view controller sends an addAction message to the UIAlertController object
  - The view controller sends the present() message



UIButton's Touch Up Inside event is connected to the view controller's `showAlert` action. When the button recognizes it has been tapped, it sends the `showAlert` message to the view controller.

Inside `showAlert`, the view controller sends another message, `addAction`, to the `UIAlertController` object. And to show the alert, the view controller sends the `present` message.

Outline

- Chapter 1: Introduction
- Chapter 2: The One-Button App
- Chapter 3: Slider and Labels

Programming To-Do List
Chapter 3 App Demo

~~(Ch. 2) The app needs to put the "Hit Me!" button on the screen and show an alert popup when the user presses it.~~

(Ch. 3) Put the app in landscape orientation

(Ch. 3) Put UI controls on the Storyboard including UILabels "Score", UIButtons "Start over", and UISlider

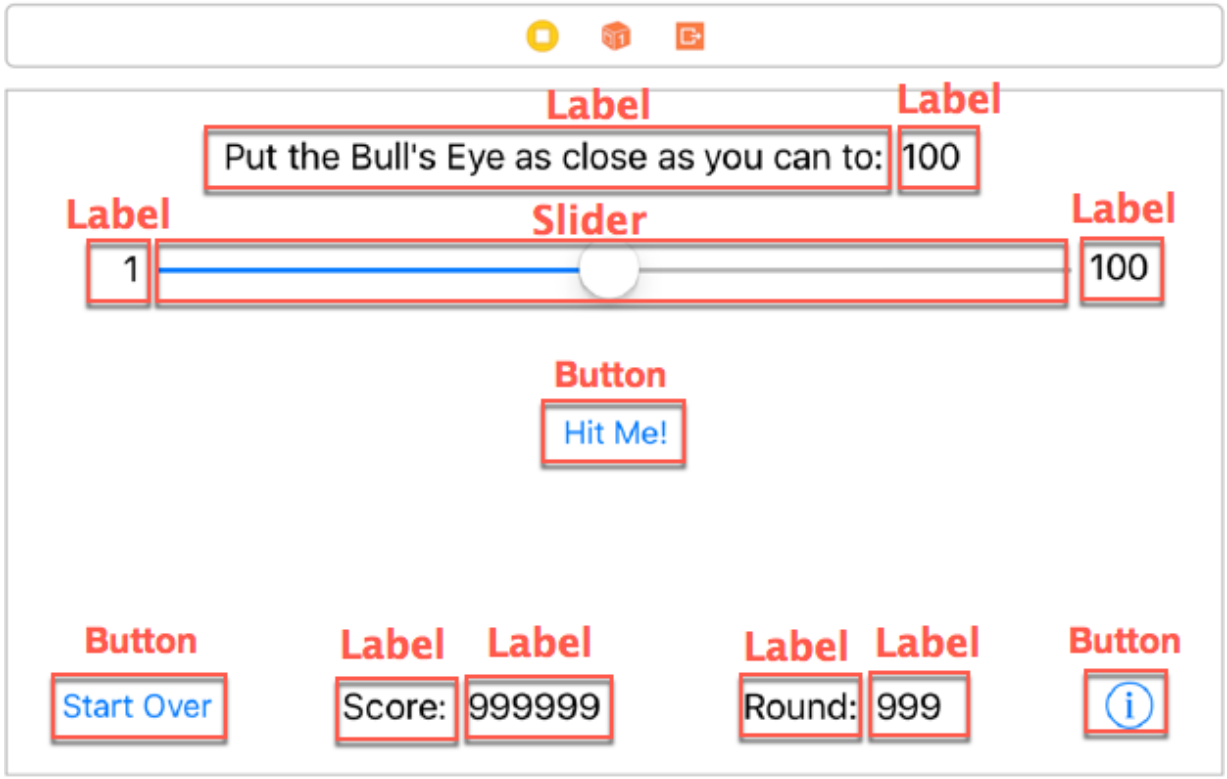
(Ch. 3) Read the value of the slider after the user presses the Hit Me button.

Points vs. Pixels

- (iPhone - iPhone 3) 1 point = 1 pixel (big chunky pixels, low resolution)
- (iPhone 4 - iPhone 8, iPads) 1 point = 2 pixels x 2 pixels (2x resolution)
- (iPhone 6+ - iPhone X, iPad Pros) 1 point = 3 pixels x 3 pixels (3x resolution)
- Developers work in points (no extra work as resolutions increase)

Device	Form factor	Screen dimension in points
iPhone 4s and older	3.5”	320 x 480
iPhone 5, 5c, 5s, SE	4”	320 x 568
iPhone 6, 6s, 7, 8	4.7”	375 x 667
iPhone 6, 6s, 7, 8 Plus	5.5”	414 x 736
iPhone X	5.8”	375 x 812
iPad, iPad mini	9.7” and 7.9”	768 x 1024
iPad Pro	10.5”	834 x 1112
iPad Pro	12.9”	1024 x 1366

Add UI Controls



Slider min 1, max 100, value 50

Read Value of Slider after HitMe!

- Create an IBAction

```
@IBAction func sliderMoved(_ slider: UISlider) {
    print("The value of the slider is now: \(slider.value)")
}
```

- Connect the UISlider to [send the sliderMoved() message to] the ViewController

Display SliderValue on UIAlert

- Create a variable to hold the slider's currentValue

```
var currentValue: Int = 0
```

- Comment out the print and set the currentValue in sliderMoved()

```
currentValue = lroundf(slider.value)
```

- Change UIAlertController message and UIAlertAction title

```
let message = "The value of the slider is: \(currentValue)"
let alert = UIAlertController(title: "Hello, World", message: message,
                                                    preferredStyle: .alert)
let action = UIAlertAction(title: "OK", style: .default, handler: nil)
```