EE443 / EE593
Mobile Application Development

Prof. Phillip De Leon

Lecture 5:
Chapter 5: Rounds and Score
Chapter 6: Polish

Prof. Phillip De Leon

Outline

- Chapter 5: Rounds and Score
- Chapter 6: Polish
- Chapter 7: The New Look

Programming To-Do List
Chapter 5

~~(Ch. 4) Fix slider initial value bug~~

~~(Ch. 4) Generate a random number at the start of each round and display it on the screen. This is the target value.~~

~~(Ch. 4) Compare the value of the slider to that random number and calculate a score based on how far off the player is. You show this score in the alert popup.~~

(Ch. 5) Compute the score for each round

(Ch. 5) Compute and display the total score

(Ch. 5) Keep track of and display the round

Get the Difference, Compute the Score

- score = 100 – abs(targetValue - currentValue)

```
@IBAction func showAlert() {
  let difference = abs(targetValue - currentValue)
  let points = 100 - difference
  let message = "You scored \(points) points”
...
}
```

Compute and Display the Total Score

- Add an instance variable to keep the total score: `var score = 0`
- Update score in showAlert(): `score += points`
- Create an outlet to the UILabel for the score:

```
@IBOutlet weak var scoreLabel: UILabel!
```

- Connect to outlet to the UILabel
- Add to updateLabels(): `scoreLabel.text = String(score)`

Keep Track of and Display the Round

- Add an instance variable to keep the round: `var round = 0`
- Create an outlet to the UILabel for the score:

```
@IBOutlet weak var roundLabel: UILabel!
```

- Connect to outlet to the UILabel
- Add to updateLabels(): `roundLabel.text = String(round)`
- Where should we increment the round? `round += 1`

Outline

- Chapter 5: Rounds and Score
- Chapter 6: Polish
- Chapter 7: The New Look

Programming To-Do List
Chapter 6

~~(Ch. 5) Compute the score for each round~~

~~(Ch. 5) Compute and display the total score~~

~~(Ch. 5) Keep track of and display the round~~

(Ch. 6) UI tweaks to make the game look and function better

(Ch. 6) Update alert view functionality so screen updates *after* alert goes away

(Ch. 6) Reset the game to start afresh

UI Tweaks:
Game Feedback in UIAlert

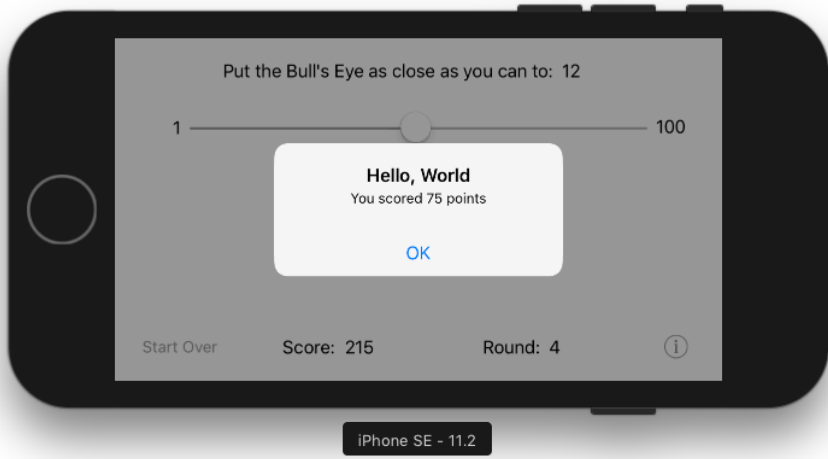
- Change UIAlertController title and message:

```
let title: String
if difference == 0 {
      title = "Perfect!"
} else if difference < 5 {
      title = "You almost had it!"
} else if difference < 10 {
      title = "Pretty good!"
} else {
      title = "Not even close..."
}

let message = "You scored \(points) points"
```

declaration

initialization




```
let alert = UIAlertController(title: title,
                              message: message,
                              preferredStyle: .alert)
```

UI Tweaks:
Bonus Points (Skip)

```
var points = 100 - difference

let title: String
if difference == 0 {
  title = "Perfect!"
  points += 100
} else if difference < 5 {
     title = "You almost had it!"
     if difference == 1 {
        points += 50 }
} else if difference < 10 {
      title = "Pretty good!"
} else {
      title = "Not even close..."
}
score += points
```

Update Alert View Functionality So Screen Updates *After* Alert Goes Away

- Demo screen updating as soon as Hit Me! is tapped
- Start new round after UIAlert dismissed
- Move startNewRound() as UIAlertAction closure — performed when OK tapped

```
let action = UIAlertAction(title: "OK", style: .default,
                           handler: { action in self.startNewRound()
})
```

`self` allows the ViewController to refer to itself inside closures we have to use `self` to refer to the VC

Reset the Game to Start Afresh

- When you hit the Start Over button, reset score, round, and startNewRound()

```
@IBAction func startNewGame() {
    score = 0
    round = 0
    startNewRound()
}
```

- Connect the outlet
- Also change viewDidLoad() to startNewGame() instead of startNewRound()
- Check out the connections for the ViewController

Outline

- Chapter 5: Rounds and Score
- Chapter 6: Polish
- Chapter 7: The New Look

Programming To-Do List
Chapter 7

~~(Ch. 6) UI tweaks to make the game look and function better~~

~~(Ch. 6) Update alert view functionality so screen updates *after* alert goes away~~

~~(Ch. 6) Reset the game to start afresh~~

(Ch. 7) Remove iPhone status bar while in landscape orientation (@homework)

(Ch. 7) Spice up the graphics (@homework)

(Ch. 7) The About Screen (get Info button)

The About Screen

- Since we need a new "screen" we need another ViewController (add the Swift file to the project and ViewController object to the storyboard)
- Get UIButton and TextView objects (with new text) onto canvas
- Create a seque (Info button to AboutViewController)

Dismiss the About Screen

- Create a "dismissal" IBAction

```
@IBAction func close() {
    dismiss(animated: true, completion: nil)
}
```

- Need to connect the AboutViewController on the Storyboard to the actual AboutViewController class in the Project
- Connect the Close button to the close() action

Add a WebView for HTML Content

- Replace the TextView with a WebView (for a URL or .html file)
- Add BullsEye.html to project
- Add an outlet for the WebView

```
@IBOutlet weak var webView: UIWebView!
```

- Connect the AboutViewController to the WebView
- Add to viewDidLoad()

```
if let url = Bundle.main.url(forResource: "BullsEye", withExtension: "html") {
    if let htmlData = try? Data(contentsOf: url) {
        let baseURL = URL(fileURLWithPath: Bundle.main.bundlePath)
        webView.load(htmlData, mimeType: "text/html", textEncodingName: "UTF-8", baseURL: baseURL)
  }
}
```