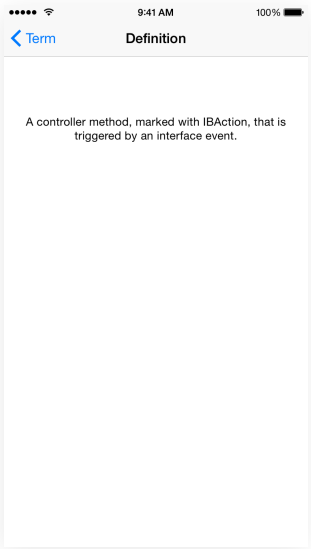
Flashcards
Lesson 6

Description

Override the term controller `viewWillAppear:animated:` method to draw a random card each time the controller is presented.



Learning Outcomes

- Analyze app behavior and relate behavior to controller life cycle methods.
- Discover the `UIViewController` `viewWillAppear:animated:` method and distinguish it from other controller life cycle methods.
- Practice overriding a controller method to customize app behavior.

Vocabulary

controller life cycle events	`UIViewController`	method override

Materials

- **Flashcards Lesson 6** Xcode project

Opening

How can we display a different flashcard `term` when returning from the definition controller?

Agenda

- Discuss the desire for a new random term to be displayed every time the `TermController` is presented.
- Discuss possible solutions, such as customizing the behavior of the Term (back) button to cause the `TermController` to pick a new `Flashcard`.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIViewController` class reference. Explore methods such as `willMoveToParentViewController:` and properties such as `parentViewController` and `presentingViewController`.
- Discuss a simple approach of having the `TermController` choose a random `Flashcard` `term` to display every time the view controller will appear, rather than during `viewDidLoad`.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIViewController` `viewWillAppear:animated:` method.
- Update the `TermController`, moving the `Flashcard` selection to a new implementation of `viewWillAppear:animated:`.

```
override func viewDidLoad() {
   super.viewDidLoad()
}

override func viewWillAppear(animated: Bool) {
   super.viewWillAppear(animated)
   if let flashcard = deck.randomCard {
      self.flashcard = flashcard
      termLabel.text = flashcard.term
   }
}
```

- Discuss how `viewWillAppear:animated:` will be called when the view first appears, and each time it reappears when transitioning back from the definition controller.
- Run the app (⌘R), tap the Definition button, observe the corresponding definition, tap the Term (back) button, and observe a new (likely, due to the random `Flashcard` selection) `term` appear. Move back and forth between term and definition to observe the changes.

Closing

What if you use some of the other controller life cycle methods to display the `term` of different flashcards? How much code might be involved? What are the benefits and drawbacks to the different approaches?

Modifications And Extensions

- Use modal transitions and gestures to move between the view controllers.
- Add a **.plist** file to the project to store a dictionary of flashcard data, and populate the `Deck` using this data instead of a hard-coded array.

Resources

View Controller Programming Guide for IOS: Responding to Display-Related Notifications http://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/RespondingtoDisplay-Notifications/RespondingtoDisplay-Notifications.html

UIViewController Class Reference https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController_Class/index.html