Flashcards
Lesson 1
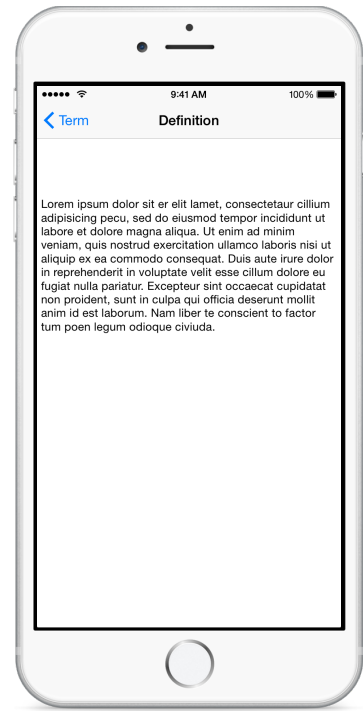
Description

Add a navigation controller to the project, and establish a hierarchy of view controllers with a segue relationship.

Learning Outcomes

- Discover and describe the purpose and behavior of a navigation controller.
- Practice adding view controllers to a storyboard.
- Discover how to bind specific view controller class definitions to view controllers in the storyboard.
- Discover how iOS apps use segues to represent the relationships between view controllers.

Vocabulary

view controller	navigation controller	initial view controller
segue	navigation bar	text view

Materials

- **Flashcards Lesson 1** Xcode project
- **Navigation Controllers** presentation

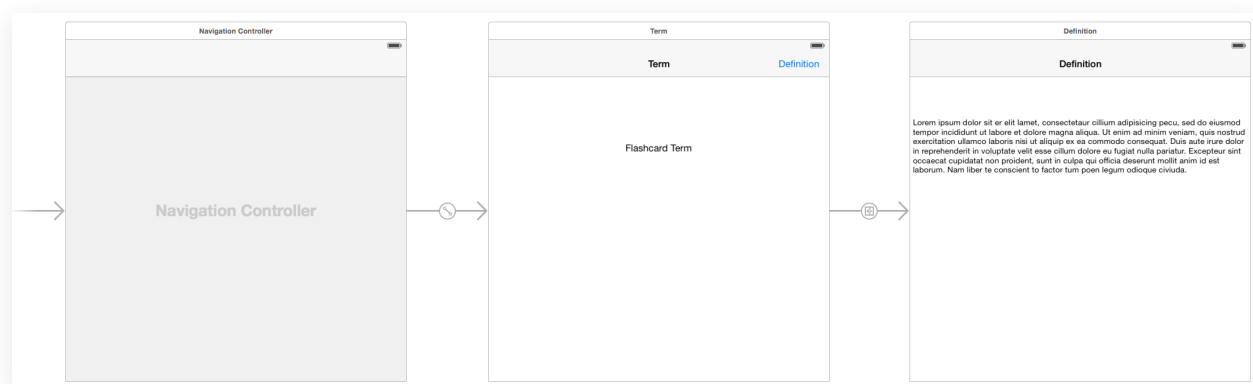## Opening

How can we create views that transition between each other?

Agenda

- Discuss how the app will initially present the front of a flashcard, which will display a particular term.
- Using Interface Builder and the Object Library (⌥⌘L), drag a Label onto the view and change the label to **Flashcard Term**.
- Add layout constraints to the label, using Control-dragging to set its vertical position and to center it within the view.
- Discuss how the user will navigate between the front and back of different flashcards, and how iOS provides navigation controllers to automatically manage transitions between multiple view controllers.
- Present the concept of navigation controllers.
- Using Interface Builder, zoom out by double-clicking in the empty space of the canvas.
- Using the Object Library (⌥⌘L), add a Navigation Controller to the storyboard.
- Notice how Interface Builder adds both a navigation controller and a table view controller.
- Delete the table view controller from the storyboard.
- Arrange the Navigation Controller to the left of the existing View Controller.
- Move the incoming arrow on the main View Controller to the Navigation Controller, to indicate that it is the initial view controller for the app.
- Control drag from the Navigation Controller to the View Controller, and select the `rootViewController` Relationship Segue.
- Explain how the navigation controller acts as a container that manages navigation between different view controllers, such as the flashcard term view controller and the yet-to-be-created definition view controller.
- Explain how Interface Builder automatically adds a navigation bar to the top of any view controllers that the navigation controller manages.
- Select the Flashcard Term label, and use the *Editor > Resolve Auto Layout Issues > Update Frames* (⌥⌘=) menu item to adjust the constraint issue caused by the navigation bar.
- Select the navigation bar at the top of the View Controller, and use the Attributes Inspector (⌥⌘4), set the title to **Term**.
- Using the Document Outline (▢), rename the View Controller to **Term Controller**.
- Using the Object Library (⌥⌘L), drag a new Bar Button Item to the navigation bar in the Term Controller, and change the button title to **Definition**.
- Run the app (⌘R), and observe how the Term Controller view appears with a navigation bar and button at the top.

- Using Interface Builder and the Object Library (⌥⌘L), add another View Controller to the storyboard, placing it to the right of the Term Controller.
- Using the Document Outline (▢), rename the new View Controller to **Definition Controller**.
- Using the Object Library (⌥⌘L), drag a Text View, for holding lots of text, onto the Definition Controller interface.
- Add layout constraints for the text view by Control-dragging from the text view on the canvas to the View in the Document Outline (▢). Create constraints for the leading, trailing, top and bottom space relative to the View.
- Control-drag from the Definition button to the Definition Controller, select the **show** segue, and observe how Interface Builder represents the new relationship with an arrow between the two view controllers.
- Explain how segues represent transitions from one view controller to another.
- Resize the text view by dragging its top edge to align with the bottom of the navigation bar, and update the changed constraint using the *Editor > Resolve Auto Layout Issues > Update Constraints* (⇧⌘=) menu item.
- Using Interface Builder and the Object Library (⌥⌘L), drag a Navigation Item onto the view and set the title to **Definition**.

- Run the app (⌘R), observe the term appear, tap the Definition button, and observe the definition appear. Tap the Term button and observe the transition back to the term view.
- Discuss how the navigation controller automatically manages the back button and the ability to move back from a segue.

Closing

What happens when you select the segue arrow, and use the Attributes Inspector to change the type of the segue?

Modifications and Extensions

- Achieve the same functionality as the existing app without using a storyboard, by creating a navigation controller and the transitions entirely with code. Criticize the benefits and drawbacks of both approaches. Assert your preferred method, and explain your choice.

Resources

Cocoa Application Competencies for iOS: Storyboard http://developer.apple.com/library/ios/documentation/General/Conceptual/Devpedia-CocoaApp/Storyboard.html

Xcode Overview: Design the User Interface of Your App with Storyboards https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/edit_user_interface.html#//apple_ref/doc/uid/TP40010215-CH6-SW1

Setting the Initial Controller for a Storyboard https://developer.apple.com/library/ios/recipes/xcode_help-IB_storyboard/chapters/SetInitialController.html

View Controller Catalog for iOS: Navigation Controllers https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/NavigationControllers.html

Presenting View Controllers from Other View Controllers http://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/ModalViewControllers/ModalViewControllers.html

Adding a Segue Between Scenes in a Storyboard https://developer.apple.com/library/ios/recipes/xcode_help-IB_storyboard/chapters/StoryboardSegue.html

UIKit User Interface Catalog: Navigation Bars https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UINavigationBar.html

UIKit User Interface Catalog: Text Views https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UITextView.html