ITP 342 Lab 4

Goal

- You will create an iPhone app to display questions and answers using **Flashcards**.
- There is a **model** holding an array of dictionaries for the flashcards.
- **Single tap** on the view to display the question of a random flashcard.
- **Double tap** on the view to display the answer.
- **Swipe right** to show the question of the previous flashcard.
- **Swipe left** to show the question of the next flashcard.
- **Shake** to display the question of a random flashcard.
- Add **animations** when displaying a flashcard.
- **Play a sound** when a flashcard is displayed (one for the question and a different one for the answer).

Assignment

- Open Xcode and create a new Single View application for the iPhone.
- To the view, add a background image or change the background color.
- Add a label for the questions.
- Create a model (Objective-C class) for our data:
  - Name it FlashcardsModel or something similar.
  - The model needs to hold a private property named **flashcards**, which is a mutable array containing dictionary objects. Each dictionary object has a key entitled “question” and a key entitled “answer”. Create static contants of type NSString for the keys.
  - Create an `init` method in the implementation to initialize the flashcards array. (Use the code snippet option from the Library in order to get the correct method signature.) Add at least 5 flashcards.
  - Add the following public methods:

```
+ (instancetype) sharedModel;
- (NSDictionary *) randomFlashcard;
- (NSUInteger) numberOfFlashcards;
- (NSDictionary *) flashcardAtIndex: (NSUInteger) index;
- (void) removeFlashcardAtIndex: (NSUInteger) index;
- (void) insertFlashcard: (NSDictionary *) flashcard;
- (void) insertFlashcard: (NSString *) question
                  answer: (NSString *) answer;
- (void) insertFlashcard: (NSDictionary *) flashcard
                 atIndex: (NSUInteger) index;
```

```
- (void) insertFlashcard: (NSString *) flashcard
                  answer: (NSString *) author
                  atIndex: (NSUInteger) index;
- (NSDictionary *) nextFlashcard;
- (NSDictionary *) prevFlashcard;
```

  - Implement the methods in the implementation.
    - For the remove and insert methods, call the similar NSMutableArray methods.
    - When removing a flashcard, check that the index is less than the number of flashcards.
    - When inserting a flashcard without an index, add it at the end.
    - When inserting a flashcard with an index, make sure the index is less than or equal to the number of flashcards.
    - Create a private property to hold the current index that is used by the nextFlashcard and prevFlashcard methods.
    - If the user swipes to display the next flashcard while displaying the last flashcard, then display the first flashcard.
    - If the user swipes to display the previous flashcard while displaying the first flashcard, then display the last flashcard.
- Edit the View Controller:
  - Create a private property for the model.
  - In the viewDidLoad method, create the model by calling the sharedModel method. Then get the first or a random flashcard and display its question.
  - Use Gesture Recognizers to display a random flashcard if the view is tapped and handle swiping left and right. Display the answer when the view is double tapped.
  - You will probably need to create a private property to help you keep track of the current flashcard or at least the answer.
  - You are unable to test the insertFlashcard:, insertFlashcard:answer:, insertFlashcard:atIndex:, and insertFlashcard:answer:atIndex:, and removeFlashcardAtIndex: methods in the model using the UI. Use XCTest, which is Apple's unit testing framework. You may use the XCTestCase class that is automatically generated for you or you may create a new class. In this class, add a private property of type FlashcardModel and update the setUp method to get a singleton. Create a test method that calls the FlashcardModel methods that need to be tested and uses the XCTAssertEqual and XCTAssertEqualObjects functions. Look at the example in the ITP342_Tests.pdf document on Blackboard under Content.
- Add the shake motion to the View Controller:
  - Add the proper motion-handling method. If the motion detected is a shake motion, then call a method to get a random flashcard and display it.

- Add animations to the View Controller:
  - Use Animations to have the flashcard label animate. Have the old flashcard fade out and then have a new question or answer fade in. You may also make other changes to the label. You may have to make some helper methods. Display the question in one color and the answer in a different color.
- Add audio to the View Controller:
  - Set up the AVAudioPlayer in the viewDidLoad method.
  - Add code to the tap and swipe gesture methods to play the sound file.
  - Add code to play the sound when the device is shaked.

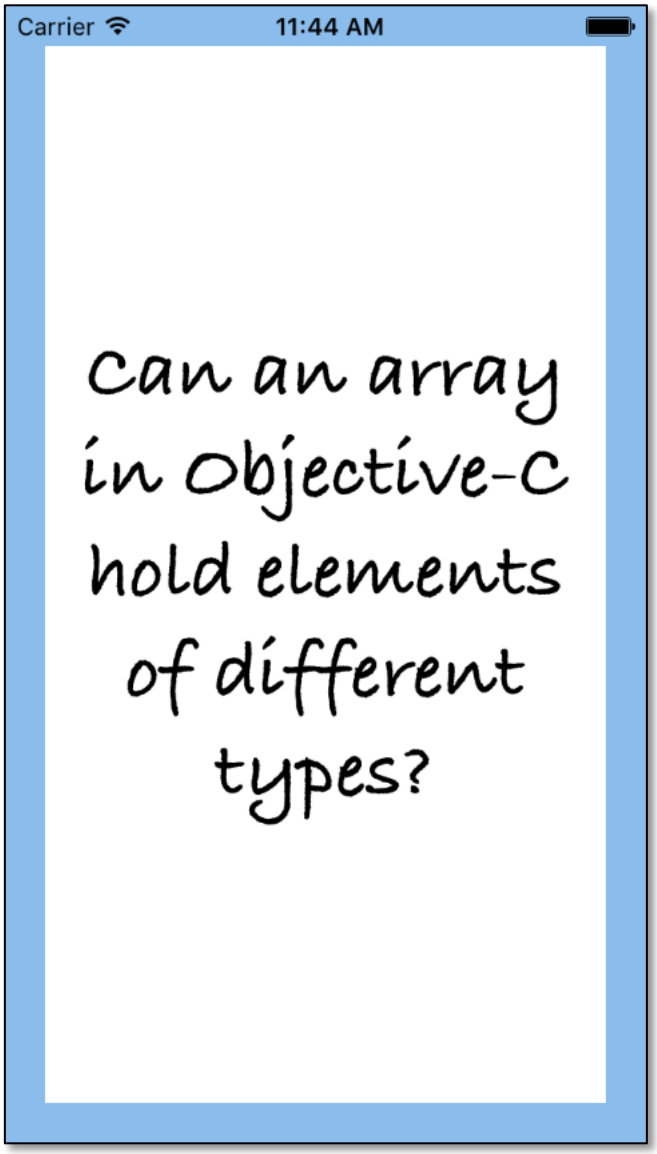
Submission

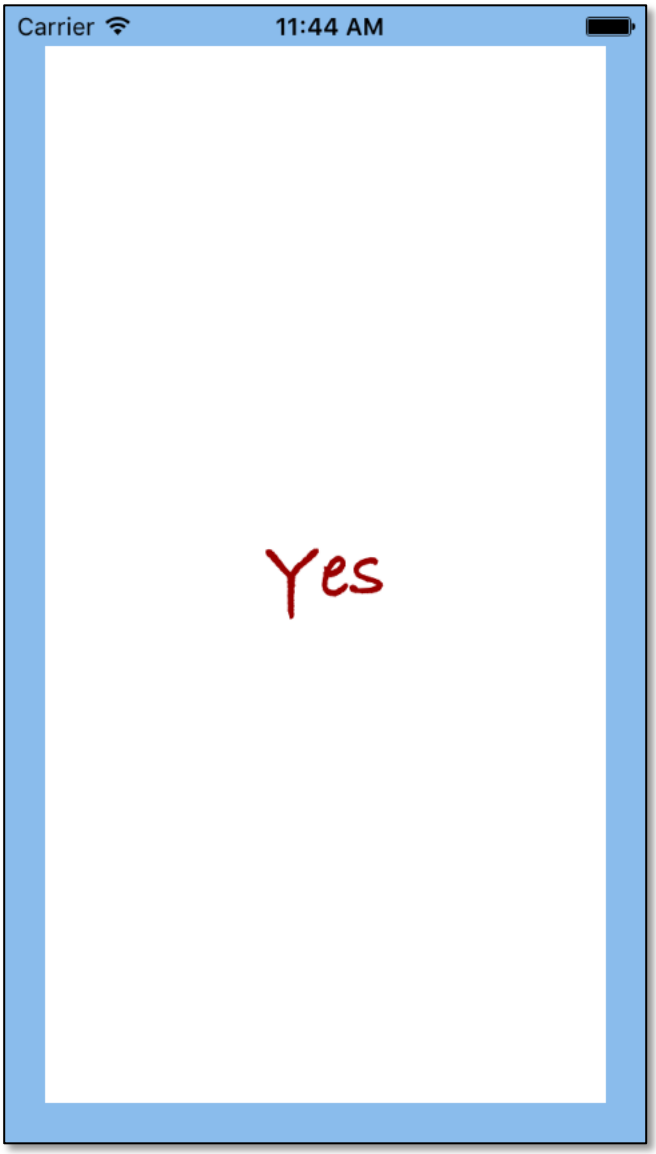
- Compress your project folder.
- Rename the zip to *LastnameFirstname*Lab4.zip.
- Submit the .zip file on Blackboard under Assignments.

Grading (60 pts)

- Model: 15 pts – quotes array, public methods, singleton
- Unit Testing of Model: 10 pts
- Gesture Recognizers: 10 pts
- Layout: 5 pts (should work in iPhone 5, iPhone 6, and iPhone 6plus)
- Shake Motion: 5 pts
- Animation (different color or something for answer): 5 pts
- Play sounds (different for answer): 5 pts
- App icon: 5 pts

Example


Carrier
11:44 AM
Can an array in Objective-C hold elements of different types?


Carrier
11:44 AM
Yes