```swift
//
//  QuizViewController.swift
//  FlagQuiz
//
//  Created by David Hincapie on 11/16/15.
//  Copyright © 2015 David Hincapie. All rights reserved.
//

import UIKit

class QuizViewController: UIViewController, ModelDelegate {
    @IBOutlet weak var flagImageView: UIImageView!
    @IBOutlet weak var questionNumberLabel: UILabel!
    @IBOutlet var segmentedControls: [UISegmentedControl]!
    @IBOutlet weak var answerLabel: UILabel!

    private var model: Model! // reference to the model object
    private let correctColor =
    UIColor(red: 0.0, green: 0.75, blue: 0.0, alpha: 1.0)
    private let incorrectColor = UIColor.redColor()
    private var quizCountries: [String]! = nil // countries in quiz
    private var enabledCountries: [String]! = nil // countries for guesses
    private var correctAnswer: String! = nil
    private var correctGuesses = 0
    private var totalGuesses = 0

    // obtains the app
    override func viewDidLoad() {
        super.viewDidLoad()

        // create Model
        model = Model(delegate: self)
        settingsChanged()
    }

    // SettingsDelegate: reconfigures quiz when user changes settings;
    // also called when app first loads
    func settingsChanged() {
        enabledCountries = model.enabledRegionCountries
        resetQuiz()
    }

    // start a new quiz
    func resetQuiz() {
        quizCountries = model.newQuizCountries() // countries in new quiz
        correctGuesses = 0
        totalGuesses = 0

        // display appropriate # of UISegmentedControls
        for i in 0 ..< segmentedControls.count {
            segmentedControls[i].hidden =
                (i < model.numberOfGuesses / 2) ? false : true
        }
    nextQuestion() // display the first flag in quiz
}

// displays next question
func nextQuestion() {
    questionNumberLabel.text = String(format: "Question %1$d of %2$d",
        (correctGuesses + 1), model.numberOfQuestions)
    answerLabel.text = ""
    correctAnswer = quizCountries.removeAtIndex(0)
    flagImageView.image = UIImage(named: correctAnswer) // next flag

    // re-enable UISegmentedControls and delete prior segments
    for segmentedControl in segmentedControls {
        segmentedControl.enabled = true
        segmentedControl.removeAllSegments()
    }

    // place guesses on displayed UISegmentedControls
    enabledCountries.shuffle() // use Array extension method
    var i = 0

    for segmentedControl in segmentedControls {
        if !segmentedControl.hidden {
            var segmentIndex = 0

            while segmentIndex < 2 { // 2 per UISegmentedControl
                if i < enabledCountries.count &&
                    correctAnswer != enabledCountries[i] {

                        segmentedControl.insertSegmentWithTitle(
                            countryFromFilename(enabledCountries[i]),
                            atIndex: segmentIndex, animated: false)
                        ++segmentIndex
                }
                ++i
            }
        }
    }

    // pick random segment and replace with correct answer
    let randomRow =
    Int(arc4random_uniform(UInt32(model.numberOfGuesses / 2)))
    let randomIndexInRow = Int(arc4random_uniform(UInt32(2)))
    segmentedControls[randomRow].removeSegmentAtIndex(
        randomIndexInRow, animated: false)
    segmentedControls[randomRow].insertSegmentWithTitle(
        countryFromFilename(correctAnswer),
        atIndex: randomIndexInRow, animated: false)
}

// converts image filename to displayable guess String
func countryFromFilename(filename: String) -> String {
    var name = filename.componentsSeparatedByString("-")[1]
    let length: Int = name.characters.count
    name = (name as NSString).substringToIndex(length - 4)
    let components = name.componentsSeparatedByString("_")
    return components.joinWithSeparator(" ")
}
```

```
// called when the user makes a guess
@IBAction func submitGuess(sender: UISegmentedControl) {
    // get the title of the bar at that segment, which is the guess
    let guess = sender.titleForSegmentAtIndex(
        sender.selectedSegmentIndex)!
    let correct = countryFromFilename(correctAnswer)
    ++totalGuesses

    if guess != correct { // incorrect guess
        // disable incorrect guess
        sender.setEnabled(false,
            forSegmentAtIndex: sender.selectedSegmentIndex)
        answerLabel.textColor = incorrectColor
        answerLabel.text = "Incorrect"
        answerLabel.alpha = 1.0
        UIView.animateWithDuration(1.0,
            animations: {self.answerLabel.alpha = 0.0})
        shakeFlag()
    } else { // correct guess
        answerLabel.textColor = correctColor
        answerLabel.text = guess + "!"
        answerLabel.alpha = 1.0
        ++correctGuesses

        // disable segmentedControls
        for segmentedControl in segmentedControls {
            segmentedControl.enabled = false
        }

        if correctGuesses == model.numberOfQuestions { // quiz over
            displayQuizResults()
        } else { // use GCD to load next flag after 2 seconds
            dispatch_after(
                dispatch_time(
                    DISPATCH_TIME_NOW, Int64(2 * NSEC_PER_SEC)),
                dispatch_get_main_queue(), {self.nextQuestion()})
        }
    }
}

// shakes the flag to visually indicate incorrect response
func shakeFlag() {
    UIView.animateWithDuration(0.1,
        animations: {self.flagImageView.frame.origin.x += 16})
    UIView.animateWithDuration(0.1, delay: 0.1, options: [],
        animations: {self.flagImageView.frame.origin.x -= 32},
        completion: nil)
    UIView.animateWithDuration(0.1, delay: 0.2, options: [],
        animations: {self.flagImageView.frame.origin.x += 32},
        completion: nil)
    UIView.animateWithDuration(0.1, delay: 0.3, options: [],
        animations: {self.flagImageView.frame.origin.x -= 32},
        completion: nil)
    UIView.animateWithDuration(0.1, delay: 0.4, options: [],
        animations: {self.flagImageView.frame.origin.x += 16},
        completion: nil)
}

// displays quiz results
```

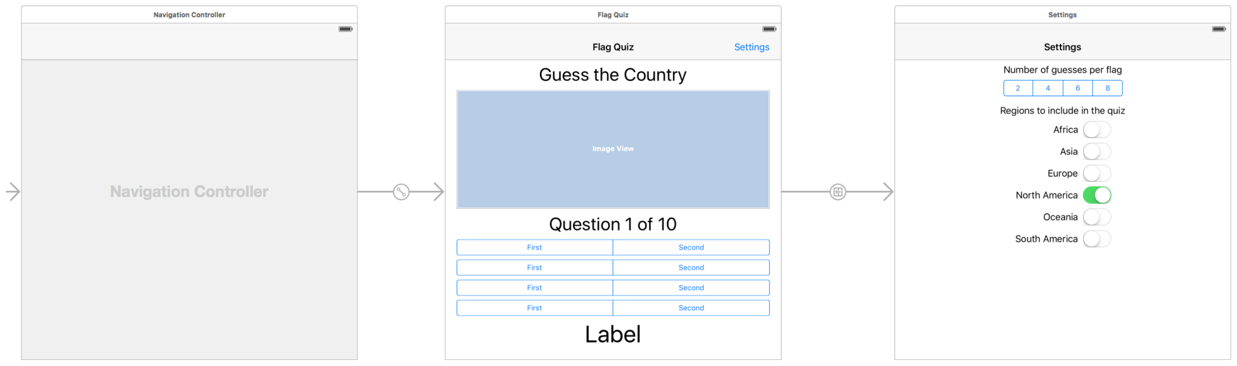
```
    func displayQuizResults() {
        let percentString = NSNumberFormatter.localizedStringFromNumber(
            Double(correctGuesses) / Double(totalGuesses),
            numberStyle: NSNumberFormatterStyle.PercentStyle)

        // create UIAlertController for user input
        let alertController = UIAlertController(title: "Quiz Results",
            message: String(format: "%1$i guesses, %2$@ correct",
                totalGuesses, percentString),
            preferredStyle: UIAlertControllerStyle.Alert)
        let newQuizAction = UIAlertAction(title: "New Quiz",
            style: UIAlertActionStyle.Default,
            handler: {(action) in self.resetQuiz()})
        alertController.addAction(newQuizAction)
        presentViewController(alertController, animated: true,
            completion: nil)
    }

    // called before seque to SettingsViewController
    override func prepareForSegue(segue: UIStoryboardSegue,
        sender: AnyObject?) {

            if segue.identifier == "showSettings" {
                let controller =
                segue.destinationViewController as! SettingsViewController
                controller.model = model
            }
    }
}

// Array extension method for shuffling elements
extension Array {
    mutating func shuffle() {
        // Modern Fisher-Yates shuffle: http://bit.ly/FisherYates
        for first in (self.count - 1).stride(through: 1, by: -1) {
            let second = Int(arc4random_uniform(UInt32(first + 1)))
            swap(&self[first], &self[second])
        }
    }
}
```

```swift
//
//  SettingsViewController.swift
//  FlagQuiz
//
//  Created by David Hincapie on 11/16/15.
//  Copyright © 2015 David Hincapie. All rights reserved.
//
import UIKit

class SettingsViewController: UIViewController {
    @IBOutlet weak var guessesSegmentedControl: UISegmentedControl!
    @IBOutlet var switches: [UISwitch]!

    var model: Model! // set by QuizViewController
    private var regionNames = ["Africa", "Asia", "Europe",
        "North_America", "Oceania", "South_America"]
    private let defaultRegionIndex = 3

    // used to determine whether any settings changed
    private var settingsChanged = false

    // called when SettingsViewController is displayed
    override func viewDidLoad() {
        super.viewDidLoad()

        // select segment based on current number of guesses to display
        guessesSegmentedControl.selectedSegmentIndex =
            model.numberOfGuesses / 2 - 1

        // set switches based on currently selected regions
        for i in 0 ..< switches.count {
            switches[i].on = model.regions[regionNames[i]]!
        }
    }

    // update guesses based on selected segment's index
    @IBAction func numberOfGuessesChanged(sender: UISegmentedControl) {
        model.setNumberOfGuesses(2 + sender.selectedSegmentIndex * 2)
        settingsChanged = true
    }

    // toggle region corresponding to toggled UISwitch
    @IBAction func switchChanged(sender: UISwitch) {
        for i in 0 ..< switches.count {
            if sender === switches[i] {
                model.toggleRegion(regionNames[i])
                settingsChanged = true
            }
        }

        // if no switches on, default to North America and display error
        if model.regions.values.lazy.filter({$0 == true}).count == 0 {
            model.toggleRegion(regionNames[defaultRegionIndex])
            switches[defaultRegionIndex].on = true
            displayErrorDialog()
        }
    }

    // display message that at least one region must be selected
    func displayErrorDialog() {
```

```
// create UIAlertController for user input
        let alertController = UIAlertController(
            title: "At Least One Region Required",
            message: String(format: "Selecting %@ as the default region.",
                regionNames[defaultRegionIndex]),
            preferredStyle: UIAlertControllerStyle.Alert)

        let okAction = UIAlertAction(title: "OK",
            style: UIAlertActionStyle.Cancel, handler: nil)
        alertController.addAction(okAction)

        presentViewController(alertController, animated: true,
            completion: nil)
    }

    // called when user returns to quiz
    override func viewWillDisappear(animated: Bool) {
        if settingsChanged {
            model.notifyDelegate() // called only if settings changed
        }
    }
}

    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation
before navigation
    override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
        // Get the new view controller using segue.destinationViewController.
        // Pass the selected object to the new view controller.
    }
    */
```

```swift
//
//  Model.swift
//  FlagQuiz
//
//  Created by David Hincapie on 11/16/15.
//  Copyright © 2015 David Hincapie. All rights reserved.
//
import Foundation

// adopted by delegate so it can be notified when settings change
protocol ModelDelegate {
    func settingsChanged()
}

class Model {
    // keys for storing data in the app's NSUserDefaults
    private let regionsKey = "FlagQuizKeyRegions"
    private let guessesKey = "FlagQuizKeyGuesses"

    // reference to QuizViewController to notify it when settings change
    private var delegate: ModelDelegate! = nil

    var numberOfGuesses = 4 // number of guesses to display
    private var enabledRegions = [ // regions to use in quiz
        "Africa" : false,
        "Asia" : false,
        "Europe" : false,
        "North_America" : true,
        "Oceania" : false,
        "South_America" : false
    ]

    // variables for maintaining quiz data
    let numberOfQuestions = 10
    private var allCountries: [String] = [] // list of all flag names
    private var countriesInEnabledRegions: [String] = []

    // initialize the Settings from the app's NSUserDefaults
    init(delegate: ModelDelegate) {
        self.delegate = delegate

        // get the NSUserDefaults object for the app
        let userDefaults = NSUserDefaults.standardUserDefaults()

        // get number of guesses
        let tempGuesses = userDefaults.integerForKey(guessesKey)
        if tempGuesses != 0  {
            numberOfGuesses = tempGuesses
        }

        // get Dictionary containing the region settings
        if let tempRegions = userDefaults.dictionaryForKey(regionsKey) {
            self.enabledRegions = tempRegions as! [String : Bool]
        }

        // get a list of all the png files in the app's images group
        let paths = NSBundle.mainBundle().pathsForResourcesOfType(
            "png", inDirectory: nil)

        // get image filenames from paths
```

```swift
for path in paths {
            if !(path as NSString).lastPathComponent.hasPrefix("AppIcon") {
                allCountries.append((path as NSString).lastPathComponent)
            }
        }

        regionsChanged() // populate countriesInEnabledRegions
    }

    // loads countriesInEnabledRegions
    func regionsChanged() {
        countriesInEnabledRegions.removeAll()

        for filename in allCountries {
            let region = filename.componentsSeparatedByString("-")[0]

            if enabledRegions[region]! {
                countriesInEnabledRegions.append(filename)
            }
        }
    }

    // returns Dictionary indicating the regions to include in the quiz
    var regions: [String : Bool] {
        return enabledRegions
    }

    // returns Array of countries for only the enabled regions
    var enabledRegionCountries: [String] {
        return countriesInEnabledRegions
    }

    // toggles a region on or off
    func toggleRegion(name: String) {
        enabledRegions[name] = !(enabledRegions[name]!)
        NSUserDefaults.standardUserDefaults().setObject(
            enabledRegions as NSDictionary, forKey: regionsKey)
        NSUserDefaults.standardUserDefaults().synchronize()
        regionsChanged() // populate countriesInEnabledRegions
    }

    // changes the number of guesses displayed with each flag
    func setNumberOfGuesses(guesses: Int) {
        numberOfGuesses = guesses
        NSUserDefaults.standardUserDefaults().setInteger(
            numberOfGuesses, forKey: guessesKey)
        NSUserDefaults.standardUserDefaults().synchronize()
    }

    // called by SettingsViewController when settings change
    // to have model notify QuizViewController of the changes
    func notifyDelegate() {
        delegate.settingsChanged()
    }

    // return Array of flags to quiz based on enabled regions
    func newQuizCountries() -> [String] {
        var quizCountries: [String] = []
        var flagCounter = 0
        // add 10 random filenames to quizCountries
        while flagCounter < numberOfQuestions {
            let randomIndex = Int(arc4random_uniform(
                UInt32(enabledRegionCountries.count)))
            let filename = enabledRegionCountries[randomIndex]

            // if image's filename is not in quizCountries, add it
            if quizCountries.filter({$0 == filename}).count == 0 {
                quizCountries.append(filename)
                ++flagCounter
            }
        }

        return quizCountries
    }
}
```


Navigation Controller
Navigation Controller
Flag Quiz
Flag Quiz
Settings
Guess the Country
Image View
Question 1 of 10
First
Second
First
Second
First
Second
First
Second
Label
Settings
Settings
Number of guesses per flag
2
4
6
8
Regions to include in the quiz
Africa
Asia
Europe
North America
Oceania
South America