```swift
//
//  IntroToParlimentaryProcedure.swift
//  FBLA-QuizME
//
//  Created by Udit Garg on 11/28/18.
//  Copyright © 2018 Udit Garg. All rights reserved.
//

import Foundation
import UIKit
import MessageUI


class IntroToParlimentaryProcedure: UIViewController {

    // Set up variables that represent labels and buttons on the storyboard
    @IBOutlet weak var QuestionLabel: UILabel!
    @IBOutlet weak var Answer1: UIButton!
    @IBOutlet weak var Answer2: UIButton!
    @IBOutlet weak var Answer3: UIButton!
    @IBOutlet weak var Answer4: UIButton!
    @IBOutlet weak var NextQuestion: UIButton!
    @IBOutlet weak var ScoreLabel: UILabel!

    // Create an array of integers that represent the number of questions
    var randomQuestionArray:[Int] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

    override func viewDidLoad() {
        super.viewDidLoad()

        // Hide Initial Next Question Buttons
        NextQuestion.isHidden = true

        //Format the buttons
        Answer1.layer.borderWidth=1
        Answer1.layer.borderColor=UIColor.darkGray.cgColor
        Answer1.layer.cornerRadius=5
        Answer2.layer.borderWidth=1
        Answer2.layer.borderColor=UIColor.darkGray.cgColor
        Answer2.layer.cornerRadius=5
        Answer3.layer.borderWidth=1
        Answer3.layer.borderColor=UIColor.darkGray.cgColor
        Answer3.layer.cornerRadius=5
        Answer4.layer.borderWidth=1
        Answer4.layer.borderColor=UIColor.darkGray.cgColor
        Answer4.layer.cornerRadius=5

        // Set the answers to be incorrect
        Answer1Correct = false
        Answer2Correct = false
        Answer3Correct = false
        Answer4Correct = false

        // As soon as the view loads start generating the questions
        RandomQuestions()

        ScoreNumber = 0
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
```

```
        // Dispose of any resources that can be recreated.
    }

    // If the answer is right then the Next Question button is enabled
    func rightAnswer() {
        NextQuestion.isHidden = false
        ScoreNumber = Int(ScoreNumber) + 2
        ScoreLabel.text = String(format: "%i", ScoreNumber)
    }

    // If the answer is wrong then the Next Question button is hidden
    func wrongAnswer() {
        ScoreNumber = Int(ScoreNumber) - 1
        ScoreLabel.text = String(format: "%i", ScoreNumber)
    }

    // This function randomly generates questions without repeat
    func RandomQuestions(){

        Answer1.isEnabled = true
        Answer2.isEnabled = true
        Answer3.isEnabled = true
        Answer4.isEnabled = true


        // This makes randomIndex represent the number of questions available for this
question
        let randomIndex = Int(arc4random_uniform(UInt32(randomQuestionArray.count)))

        // Generates questions until all of the questions for this topic have been
answered
        if randomQuestionArray.count > -1 {

            switch (randomQuestionArray[randomIndex]) {
            case 0:
                QuestionLabel.text = "Which article in the FBLA bylaws describes the
information about FBLA dues?"
                Answer1.setTitle("Article III", for: .normal)
                Answer2.setTitle("Article IV", for: .normal)
                Answer3.setTitle("Article VI", for: .normal)
                Answer4.setTitle("Article V", for: .normal)
                Answer2Correct = true
            case 1:
                QuestionLabel.text = "To appeal is:"
                Answer1.setTitle("close the meeting", for: .normal)
                Answer2.setTitle("question chair's ruling", for: .normal)
                Answer3.setTitle("agree", for: .normal)
                Answer4.setTitle("modify meaning", for: .normal)
                Answer2Correct = true
            case 2:
                QuestionLabel.text = "Viva voce means:"
                Answer1.setTitle("voice vote", for: .normal)
                Answer2.setTitle("always for the motion", for: .normal)
                Answer3.setTitle("no objection", for: .normal)
                Answer4.setTitle("objection", for: .normal)
                Answer1Correct = true
            case 3:
                QuestionLabel.text = "To delay consideration of the main motion until
the next regular meeting:"
                Answer1.setTitle("lay on the table", for: .normal)
                Answer2.setTitle("limit debate", for: .normal)
```

```
                Answer3.setTitle("postpone indefinitely", for: .normal)
                Answer4.setTitle("postpone definitely", for: .normal)
                Answer4Correct = true
            case 4:
                QuestionLabel.text = "The lack of a second has become immaterial"
                Answer1.setTitle("after the chair has stated the question", for:
.normal)
                Answer2.setTitle("when the vote is taken", for: .normal)
                Answer3.setTitle("after debate has begun", for: .normal)
                Answer4.setTitle("all of the above", for: .normal)
                Answer2Correct = true
            case 5:
                QuestionLabel.text = "A committee composed of national officers and
board members that carefully considers applicants for officers of FBLA is appointed by
the:"
                Answer1.setTitle("CEO", for: .normal)
                Answer2.setTitle("FBLA President", for: .normal)
                Answer3.setTitle("Board of Directors", for: .normal)
                Answer4.setTitle("National Executive Council", for: .normal)
                Answer2Correct = true
            case 6:
                QuestionLabel.text = "If the office of FBLA president becomes vacant,
the _________________ shall automatically become president. "
                Answer1.setTitle("vice president", for: .normal)
                Answer2.setTitle("secretary", for: .normal)
                Answer3.setTitle("other candidate with the next most votes", for:
.normal)
                Answer4.setTitle("vice president from the president's region", for:
.normal)
                Answer4Correct = true
            case 7:
                QuestionLabel.text = "Rescind means:"
                Answer1.setTitle("cancel", for: .normal)
                Answer2.setTitle("agreement that motion be considered", for: .normal)
                Answer3.setTitle("give way to", for: .normal)
                Answer4.setTitle("no objection", for: .normal)
                Answer1Correct = true
            case 8:
                QuestionLabel.text = "The quorum of an assembly"
                Answer1.setTitle("is the largest dependent voting board", for:
.normal)
                Answer2.setTitle("always a majority of the members", for: .normal)
                Answer3.setTitle("is the members needed for business to occur
legally", for: .normal)
                Answer4.setTitle("is two-thirds of the members present at the
meeting", for: .normal)
                Answer3Correct = true
            case 9:
                QuestionLabel.text = "Corrections to the minutes"
                Answer1.setTitle("should only be made on the minutes being corrected",
for: .normal)
                Answer2.setTitle("may be made only after they are read to the
assembly", for: .normal)
                Answer3.setTitle("cannot be made after being accepted by the
assembly", for: .normal)
                Answer4.setTitle("none of the above", for: .normal)
                Answer1Correct = true
            case 10:
                QuestionLabel.text = "Proposed amendments to the bylaws must be
submitted in writing to the"
                Answer1.setTitle("Association President.", for: .normal)
```

```
                Answer2.setTitle("National Executive Council.", for: .normal)
                Answer3.setTitle("FBLA president.", for: .normal)
                Answer4.setTitle("Board of directors.", for: .normal)
                Answer1Correct = true
            case 11:
                QuestionLabel.text = "The secretary must supply at least _____ copies
of the minutes to the necessary authorities."
                Answer1.setTitle("four", for: .normal)
                Answer2.setTitle("two", for: .normal)
                Answer3.setTitle("three", for: .normal)
                Answer4.setTitle("one", for: .normal)
                Answer2Correct = true
            case 12:
                QuestionLabel.text = "Action of the National Executive Council must be
included in the minutes of the"
                Answer1.setTitle("adjourned meeting.", for: .normal)
                Answer2.setTitle("special meeting.", for: .normal)
                Answer3.setTitle("regular meeting.", for: .normal)
                Answer4.setTitle("next regular meeting.", for: .normal)
                Answer4Correct = true
            default:
                break
            }
            // Removes the possibility of the question that was just shown to be shown
again
            randomQuestionArray.remove(at: randomIndex)
        }

        // If the user is on the last question then show them that they have reached
the last question
        if (randomQuestionArray.count < 1) {
            let alert = UIAlertController(title: "Wow!", message: "You have reached
the last question for Introduction to Parlimentary Procedure! Nice Job! Complete this
question and then click on 'Your Score' for a rating!", preferredStyle: .alert)
            alert.addAction(UIAlertAction(title: "Continue", style: .default, handler:
{ action in
                switch action.style{
                case .default:
                    print("default")
                case .cancel:
                    print("cancel")
                case .destructive:
                    print("destructive")
                }}))
            self.present(alert, animated: true, completion: nil)
            NextQuestion.isEnabled = false
        }

        if (randomQuestionArray.count == 0) {
            let alert = UIAlertController(title: "Wow!", message: "You got
\(ScoreNumber) out of 13 questions correct nice job! To see a detailed breakdown of
your score, click-Your Score-next to your score number.", preferredStyle: .alert)
            alert.addAction(UIAlertAction(title: "Continue", style: .default, handler:
{ action in
                switch action.style{
                case .default:
                    print("default")
                case .cancel:
                    print("cancel")
                case .destructive:
                    print("destructive")
```

```
                }}))
            self.present(alert, animated: true, completion: nil)
        }
    }

    // These 4 functions tell the user if they got the correct answer or if they got
the incorrect answer
    @IBAction func Answer1(_ sender: Any) {
        if Bool(Answer1Correct) == true {
            rightAnswer()
            Answer1.layer.backgroundColor = UIColor.green.cgColor
            Answer1.isEnabled = false
            Answer2.isEnabled = false
            Answer3.isEnabled = false
            Answer4.isEnabled = false
        } else {
            wrongAnswer()
            Answer1.layer.backgroundColor = UIColor.red.cgColor
            Answer1.isEnabled = false
        }
    }

    @IBAction func Answer2(_ sender: Any) {
        if Bool(Answer2Correct) == true {
            rightAnswer()
            Answer2.layer.backgroundColor = UIColor.green.cgColor
            Answer2.isEnabled = false
            Answer1.isEnabled = false
            Answer3.isEnabled = false
            Answer4.isEnabled = false
        } else {
            wrongAnswer()
            Answer2.layer.backgroundColor = UIColor.red.cgColor
            Answer2.isEnabled = false
        }
    }

    @IBAction func Answer3(_ sender: Any) {
        if Bool(Answer3Correct) == true {
            rightAnswer()
            Answer3.layer.backgroundColor = UIColor.green.cgColor
            Answer3.isEnabled = false
            Answer1.isEnabled = false
            Answer2.isEnabled = false
            Answer4.isEnabled = false
        } else {
            wrongAnswer()
            Answer3.layer.backgroundColor = UIColor.red.cgColor
            Answer3.isEnabled = false
        }
    }

    @IBAction func Answer4(_ sender: Any) {
        if Bool(Answer4Correct) == true {
            rightAnswer()
            Answer4.layer.backgroundColor = UIColor.green.cgColor
            Answer4.isEnabled = false
            Answer1.isEnabled = false
            Answer2.isEnabled = false
            Answer3.isEnabled = false
        } else {
```

```
            wrongAnswer()
            Answer4.layer.backgroundColor = UIColor.red.cgColor
            Answer4.isEnabled = false
        }
    }

    // Resets the colors and answers and generates another question
    @IBAction func NextQuestion(_ sender: Any) {
        Answer1.layer.backgroundColor = UIColor.white.cgColor
        Answer2.layer.backgroundColor = UIColor.white.cgColor
        Answer3.layer.backgroundColor = UIColor.white.cgColor
        Answer4.layer.backgroundColor = UIColor.white.cgColor
        NextQuestion.isHidden = true
        Answer1Correct = false
        Answer2Correct = false
        Answer3Correct = false
        Answer4Correct = false
        RandomQuestions()
    }
}
```