UNIVERSITI TUNKU ABDUL RAHMAN

ASSIGNMENT 2 (15%)

**UECS3263 iOS APPLICATION DEVELOPMENT**

BACHELOR OF SCIENCE (HONOURS) SOFTWARE ENGINEERING

|  |  |
| --- | --- |
| Name (as stated in Student Card) | Student ID |
| Gervin Fung Da Xuen | 1801655 |
| Programme | Submission Date |
| Software Engineering | 9/7/2021 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | | **Total Marks** | **Marks Given** |
| User Interface Design – CO2 (4 marks) | |  |  |
|  | Pictures | 1 |  |
|  | Entry of musical instrument names | 1 |  |
|  | Feedback on correctness | 1 |  |
|  | Number of correct answers | 1 |  |
| App Construction and Execution – CO3 (11 marks) | |  |  |
|  | Requirements implemented | 3 |  |
|  | Code design | 4 |  |
|  | Executes successfully | 1 |  |
|  | Originality | 1 |  |
|  | Code readability (e.g. meaningful names) | 1 |  |
|  | User guide | 1 |  |
| Total | | 15 |  |

Table of Contents

[Application Screenshots 1](file:///C:\Users\gervi\Documents\Sem1\IOS\GervinFungDaXuen.docx#_Toc75965887)

[Code Listing 4](file:///C:\Users\gervi\Documents\Sem1\IOS\GervinFungDaXuen.docx#_Toc75965888)

Application Screenshot

1. Before Enter Instruments Name

Graphical user interface, application

Description automatically generated

1. Enter wrong name for Guitar

Graphical user interface, application, Teams

Description automatically generated

1. After click Check Answer, reduce number of tries, and display Feedback

Graphical user interface, application, Teams

Description automatically generated

1. Another 2 attempts for wrong answer, reduce number of tries to 0, display Feedback to user and display correct answer, which user need to slide to see as too long of input box may seems taunting as users (kids) might feel that answer needed is too long, hence a smaller input box would justify that

Graphical user interface, application

Description automatically generated

1. Make correct guess, leading and trailing whitespace or upper-lower case are allowed as the text will trim any leading and trailing whitespace and transform cases to lower case

Graphical user interface, application

Description automatically generated

1. Finish answering

Purposefully answer 2 questions wrongly to show that it actually worked, this alert message will keep on display when user click next instrument and had reached the last instrument

Application

Description automatically generated with low confidence

2. Code Listing

Mostly used drag-and-drop but also used object and class to separate app logic from interface instead of combining all into one

import UIKit

final *class* AppLogic {

    // Immutable

    private let maximumNumberOfTries: *Int*

    private let answerArrayList: *Array*<*String*>

    private let numberOfInstruments: *Int*

    // Mutable

    private var currentQuestion: *Int*

    private var correctAnswered: *Int*

    private var currentNumberOfTries: *Int*

    public *init*() {

        self.maximumNumberOfTries = 3

        self.currentQuestion = 0

        self.correctAnswered = 0

        self.currentNumberOfTries = 0

        self.answerArrayList = ["accordion", "keyboard", "drum", "guitar", "clarinet", "piano", "saxophone", "bongos", "trumpet", "maracas", "trombone", "tambourine", "triangle", "banjo", "cello", "flute", "violin", "recorder", "harp", "xylophone"]

        self.numberOfInstruments = answerArrayList.count

    }

    // String func

    public *func* getImagePath() -> *String* {

        return "instrumentsImg/\(answerArrayList[self.currentQuestion]).jpg"

    }

    public *func* getNQuestion() -> *String* {

        return "Question \(*String*(self.currentQuestion + 1))/\(*String*(self.numberOfInstruments))"

    }

    public *func* getNumberOfTries() -> *String* {

        return "Number of Tries: \(*String*(self.maximumNumberOfTries - self.currentNumberOfTries))"

    }

    public *func* getCurrentCorrectAnswer() -> *String* {

        return self.answerArrayList[self.currentQuestion]

    }

    // Int func

    public *func* getCurrentNumberOfTries() -> *Int* {

        return self.currentNumberOfTries

    }

    public *func* getMaxTries() -> *Int* {

        return self.maximumNumberOfTries

    }

    public *func* getCurrentQuestion() -> *Int* {

        return self.currentQuestion

    }

    public *func* getNumberOfInstruments() -> *Int* {

        return self.numberOfInstruments

    }

    public *func* getNumberOfCorrectAnswer() -> *Int* {

        return self.correctAnswered

    }

    // Void

    public *func* incrementCorrectedAnswer() -> *Void* {

        self.correctAnswered += 1

    }

    public *func* incrementCurrentQuestion() -> *Void* {

        self.currentQuestion += 1

    }

    public *func* incrementNumberOfTries() -> *Void* {

        self.currentNumberOfTries += 1

    }

    public *func* resetNumberOfTries() -> *Void* {

        self.currentNumberOfTries = 0

    }

}

final *class* ViewController: UIViewController, UITextFieldDelegate {

    @IBOutlet weak var nQuestionLabel: UILabel!

    @IBOutlet weak var musicInstrumentImg: UIImageView!

    @IBOutlet weak var numberOfTriesLabel: UILabel!

    @IBOutlet weak var answerTextField: UITextField!

    @IBOutlet weak var feedbackLabel: UILabel!

    @IBOutlet weak var nxtBtn: UIButton!

    @IBOutlet weak var checkAnsBtn: UIButton!

    //constants

    private let appLogic = AppLogic()

    override *func* viewDidLoad() {

        super.viewDidLoad()

        feedbackLabel.isHidden = true

        nxtBtn.isHidden = true

        answerTextField.delegate = self

        nQuestionLabel.text = appLogic.getNQuestion()

        numberOfTriesLabel.text = appLogic.getNumberOfTries()

        musicInstrumentImg.image = UIImage(named: appLogic.getImagePath())

        nxtBtn.setTitle("Next Instrument", for: .normal)

        checkAnsBtn.setTitle("Check Answer", for: .normal)

    }

    private *func* setPropertiesOfFeedbackLabel(isRed: *Bool*, text: *String*) -> *Void* {

        feedbackLabel.isHidden = false

        feedbackLabel.textColor = isRed ? UIColor.red :  UIColor.green

        feedbackLabel.text = text

    }

    @IBAction *func* checkButtonPressed(\_ *sender*: *Any*) {

        let answer = answerTextField.text?.lowercased().trimmingCharacters(in: .whitespacesAndNewlines)

        if (appLogic.getCurrentNumberOfTries() < appLogic.getMaxTries()) {

            if (answer == appLogic.getCurrentCorrectAnswer()) {

                self.setPropertiesOfFeedbackLabel(isRed: false, text: "Hooray! You get the correct answer!")

                nxtBtn.isHidden = false

                answerTextField.isUserInteractionEnabled = false

                appLogic.incrementCorrectedAnswer()

            } else {

                self.setPropertiesOfFeedbackLabel(isRed: true, text: "Answer entered is Wrong!")

                appLogic.incrementNumberOfTries()

                numberOfTriesLabel.text = appLogic.getNumberOfTries()

                answerTextField.text = nil

            }

        }

        if (appLogic.getCurrentNumberOfTries() == appLogic.getMaxTries()) {

            nxtBtn.isHidden = false

            self.setPropertiesOfFeedbackLabel(isRed: true, text: "You had used up the tries for this question!")

            answerTextField.textColor = UIColor.red

            answerTextField.text = "The Correct Answer is: \(appLogic.getCurrentCorrectAnswer())"

        }

    }

    @IBAction *func* nextButtonPressed(\_ *sender*: UIButton) {

        if (appLogic.getCurrentQuestion() < appLogic.getNumberOfInstruments() - 1) {

            feedbackLabel.isHidden = true

            nxtBtn.isHidden = true

            answerTextField.isUserInteractionEnabled = true

            appLogic.incrementCurrentQuestion()

            nQuestionLabel.text = appLogic.getNQuestion()

            appLogic.resetNumberOfTries()

            numberOfTriesLabel.text = appLogic.getNumberOfTries()

            answerTextField.text = nil

            answerTextField.textColor = UIColor.black

            musicInstrumentImg.image = UIImage(named: appLogic.getImagePath())

        } else {

            let alertController = UIAlertController(title: "Quiz is completed!", message: "Congratulations! Your score is \(appLogic.getNumberOfCorrectAnswer())/\(appLogic.getNumberOfInstruments())", preferredStyle: UIAlertController.Style.alert)

            let cancelAction = UIAlertAction(title: "OK", style: UIAlertAction.Style.cancel, handler: nil)

            alertController.addAction(cancelAction)

            self.present(alertController, animated: true, completion: nil)

        }

    }

*func* textFieldShouldReturn(\_ *textField*: UITextField) -> *Bool* {

        textField.resignFirstResponder()

        return true

    }

}