



MATES Computer Science

Senior Capstone Project Bi-Weekly Progress Report

Project Title	Athena
Team Members	Me
Dates Covered by Report	5/6-5/16
Link to Github	https://github.com/AlexBH74/Athena

1. **Summary of Project** (Provide a one paragraph summary of your project. You can largely copy/paste this from one progress report to the next, unless there are significant changes.)

My plan is to create an Apple game for daily trivia. The app will draw from a MySQL database of trivia questions and multiple choice answers. When the app opens there is a login and create account screen. After logging in there is an onboarding screen that has slides that list the game's summary and general instructions. After clicking next past all of the slides the screen changes and there is an interface with easy, medium, and hard modes. Each player will be able to play one question for each mode each day. Once the user presses the mode of the question they will be able to click a start button. A timer will start after the button is clicked and the screen will display the question with four multiple choice answers. The timer can be paused but the trivia question is blocked from view if the game is stopped. At the end whether or not you answered correctly and your time will be displayed in a pop-up. As a reach goal, there will be a separate tab on the app containing a table of each user's career stats, line graphs/trend lines and bar plots showing progress over time

2. Summary of Progress this Period (Provide a high-level, one paragraph overview of what was accomplished this progress period collectively by the team.)

In the past two weeks I have gotten a lot of tasks completed. I first got the time to display on the correct pop up screen through the use of a different timer. This timer fires every 0.1 seconds and updates the text displayed on the correct pop up with the time of the last correct answer for each difficulty. This means that it will continuously update the time as the correct time from the previous day until the new time is added to the list when the question is answered, causing it to change to the most recent time. The timer is invalidated when the user returns to the homescreen.

The next thing I did was get the medium and hard views working. I copied and pasted the items and code from the easy view into their view controllers and then altered all the variables so they were oriented to the difficulty of the controller. I had to reconnect all of the objects which was tedious but not complicated.

After the three different view controllers were set up I moved on to solving how to make only one question available per day. Turns out there's a function that draws on the current date, so I can check the last time a question was answered against the current date. If they are the same the button locks, and you can no longer answer another question of that difficulty for each day.

One of the last things I worked on was making the timer save, before the question is answered so that you can't just exit out and reset your time. I added two defaults to the timer that saves the seconds once the start clock button is pressed and until the question is answered or a new day starts the time will not reset.

The last thing I worked on surprisingly only took me one class period to complete. I was able to save the index of the question until it is answered or it is a new day by adding a if/else statement in the function that loads the question. The statement checks whether the boolean "indexSaves" is true or false, and if true it returns the saved index. It is only turned false when the question is answered or the day changes.

3. **Detailed Progress this Period, separated by Team Member** (Provide detailed information on the progress that you made in the reporting weeks. Include screenshots of code, your game or website, etc. Each team member should have a separate subsection covering their accomplishments. Not including screenshots, this section should be 1-2 pages.)

```
20 override func viewDidLoad() {
21     super.viewDidLoad()
22
23     timer = Timer.scheduledTimer(timeInterval: 0.1, target: self, selector: #selector(correctDisplayCounter), userInfo: nil, repeats: true)
24 }
25
26 @objc func correctDisplayCounter() -> Void {
27     let display = UserDefaults.standard.bool(forKey: "easyCorrectShowing")
28     if display == true {
29         let correctTimes = UserDefaults.standard.object(forKey: "easyCorrectTimes") as? [String]
30
31         let index = correctTimes!.count
32
33         if index != 0 {
34             lastInsertedTime = correctTimes![index-1]
35
36             time = "Time - " + lastInsertedTime
37
38             self.timeLabel.text = time
39         }
40     }
41 }
42
43 @IBAction func homeClicked(_ sender: Any) {
44     UserDefaults.standard.set(false, forKey: "easyCorrectShowing")
45     timer.invalidate()
46     goToHomescrren()
47 }
```

Above is the timer function that updates and displays the time on the easy correct view controller.

The same code is in the other two difficulties as well.

```
10 class HomescreenViewController: UIViewController {
11
12     private var currentDate = Date()
13     let format = DateFormatter()
14     private var easy = UserDefaults.standard.bool(forKey: "easyDone")
15     private var medium = UserDefaults.standard.bool(forKey: "mediumDone")
16     private var hard = UserDefaults.standard.bool(forKey: "hardDone")
17
18     @IBOutlet weak var questionLabel: UILabel!
19     @IBOutlet weak var easyLockImage: UIImageView!
20     @IBOutlet weak var mediumLockImage: UIImageView!
21     @IBOutlet weak var hardLockImage: UIImageView!
22     @IBOutlet weak var easyBtn: UIButton!
23     @IBOutlet weak var medBtn: UIButton!
24     @IBOutlet weak var hardBtn: UIButton!
25
26     override func viewDidLoad() {
27         super.viewDidLoad()
28
29         self.easyLockImage.isHidden = true
30         self.mediumLockImage.isHidden = true
31         self.hardLockImage.isHidden = true
32
33         format.dateFormat = "yyyy-MM-dd"
34         let date = format.string(from: currentDate)
35
```

Above in the home screen view controller are the date function variables and then the formatter that converts the date into a string.

```
35     //UserDefaults.standard.set(nil, forKey: "easyLastDate") //comment out
36     var easyLastDate = UserDefaults.standard.string(forKey: "easyLastDate")
37     if easyLastDate == nil {
38         easyLastDate = "2024-01-01"
39         print(easyLastDate!)
40     } else {
41         print(easyLastDate!)
42     }
43
44     if date == easyLastDate {
45         print(date)
46         easy = UserDefaults.standard.bool(forKey: "easyDone")
47     } else {
48         print(date)
49         UserDefaults.standard.set(false, forKey: "easyDone")
50         if date != UserDefaults.standard.string(forKey: "easyLastReset") {
51             UserDefaults.standard.set(false, forKey: "easyIndexSaves")
52             UserDefaults.standard.set(0, forKey: "easyTimerNum")
53             UserDefaults.standard.set(date, forKey: "easyLastReset")
54         }
55         easy = UserDefaults.standard.bool(forKey: "easyDone")
56     }
57
58     if easy == true {
59         easyBtn.isEnabled = false
60         self.easyLockImage.isHidden = false
61     } else {
62         easyBtn.isEnabled = true
63         print(easy)
64     }
65 }
```

Above is the code that makes it so only one easy question can be answered per day. The code is the same for the medium and hard difficulties.

```
@objc func timerCounter() -> Void {
    timeSaves = UserDefaults.standard.bool(forKey: "easyTimeSaves")

    if timeSaves == true {
        num = UserDefaults.standard.integer(forKey: "easyTimerNum")
        UserDefaults.standard.set(false, forKey: "easyTimeSaves")
        print(num)
    }

    num = num + 1
    UserDefaults.standard.set(num, forKey: "easyTimerNum")

    let time = secondsToMinutesSeconds(seconds: num)
    timeString = makeTimeString(minutes: time.0, seconds: time.1)
    if timeString == "60:00" {
        stopCounting()
        answerIncorrect()
    } else {
        self.timerLabel.text = timeString
    }
}
```

Above is the modified code for the easy timer that allows the timer to be saved. The code is the same for the medium and hard difficulties.

```
private func answerCorrect() {
    print("Correct!")
    correctTimes.append(timeString)
    correctTimes = [] //comment out to make correct times save
    print(correctTimes)
    UserDefaults.standard.set(true, forKey: "easyCorrectShowing")
    UserDefaults.standard.set(true, forKey: "easyDone")
    UserDefaults.standard.set(currentDate, forKey: "easyLastDate")
    UserDefaults.standard.set(0, forKey: "easyTimerNum")
    UserDefaults.standard.set(false, forKey: "easyTimeSaves")
    UserDefaults.standard.set(false, forKey: "easyIndexSaves")
    UserDefaults.standard.set(currentDate, forKey: "easyLastReset")
    self.correctPopUp.isHidden = false
    navigationItem.setHidesBackButton(true, animated: true)
    self.homeBtn.isHidden = true
}

private func answerIncorrect() {
    print("Incorrect!")
    resetTimer()
    UserDefaults.standard.set(true, forKey: "easyIncorrectShowing")
    UserDefaults.standard.set(true, forKey: "easyDone")
    UserDefaults.standard.set(currentDate, forKey: "easyLastDate")
    UserDefaults.standard.set(0, forKey: "easyTimerNum")
    UserDefaults.standard.set(false, forKey: "easyIndexSaves")
    UserDefaults.standard.set(currentDate, forKey: "easyLastReset")
    UserDefaults.standard.set(false, forKey: "easyTimeSaves")
    self.incorrectPopUp.isHidden = false
    navigationItem.setHidesBackButton(true, animated: true)
    self.homeBtn.isHidden = true
}
```

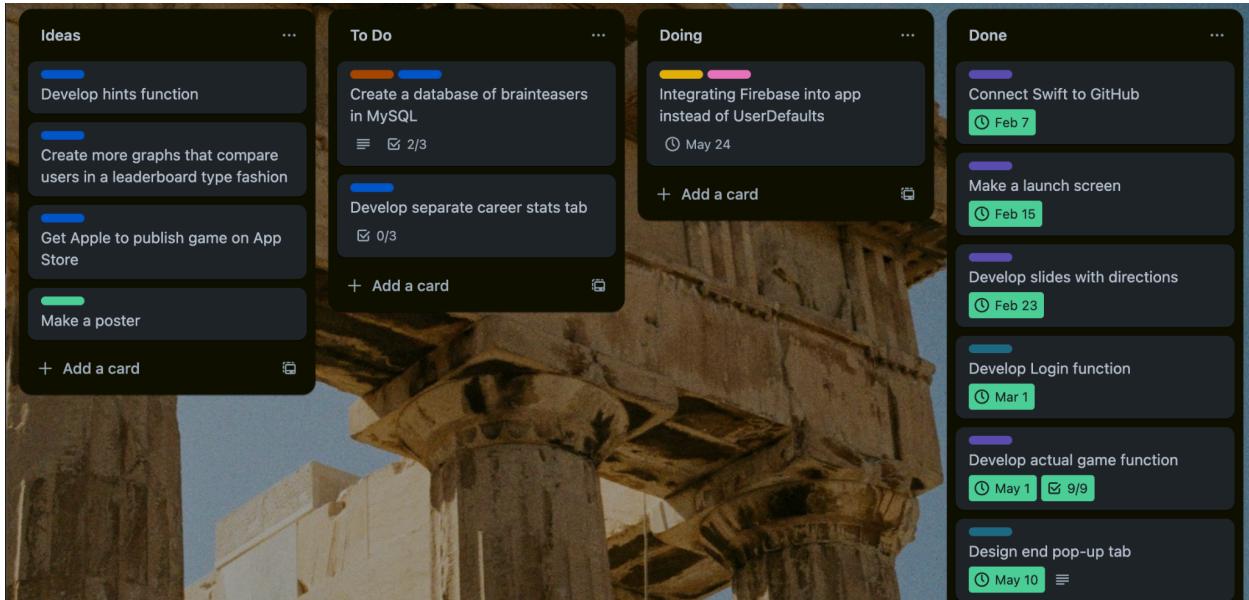
Above is the modified code for easy answer correct and incorrect functions. There are some added UserDefaults. The code is the same for the medium and hard difficulties.

I attached the screen recording to the classroom showing the swift code output from the homescreen onwards.

4. **Difficulties Encountered this Progress Period** (Provide detailed information on the difficulties and issues that you encountered in the reporting weeks. Discuss mitigation strategies for how you got around or plan to get around these issues.)

The most major difficulty I encountered was implementing the timers to update the time displayed on the correct view controller. Working the logic and implementing the defaults in the right places took a few days to figure out.

5. **Updated Trello Board and Discussion** (Provide screenshot of and link to updated Trello board. Discuss any changes made to board since last progress report and why.)
<https://trello.com/b/lYqXeq/athena-capstone-project>



6. **Tasks to Be Worked on in Next Progress Period** (Discuss the tasks to be worked on in the following two weeks. Discuss who is working on each.)

I attempt to implement firebase instead of user defaults.

7. **Additional Information** (Provide any additional information that you want to provide in this section; for example, one of your teammates is going away next week, your Github account is gone, etc. It could be good news as well.)

