MATES Computer Science

Senior Capstone Project Bi-Weekly Progress Report

Project Title	Athena
Team Members	Me
Dates Covered by Report	2/12-2/22
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 brain teasers. The app will draw from a database of brain teasers created in SQLite. When the app opens there will be slides that list the game's summary and general instructions. After clicking next past all of that the user will have to create a login so that progress for every user can be tracked. Once you login the screen will change and the player can press the start button on the screen to begin playing. A timer will then start and the screen will display the teaser with a text box to type the answer. The timer can be paused but the teaser is blocked from view if the game is stopped. The user will have 3 strikes before they have to answer correctly. At the end whether or not you answered correctly and your time will be displayed in a pop-up. 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.)

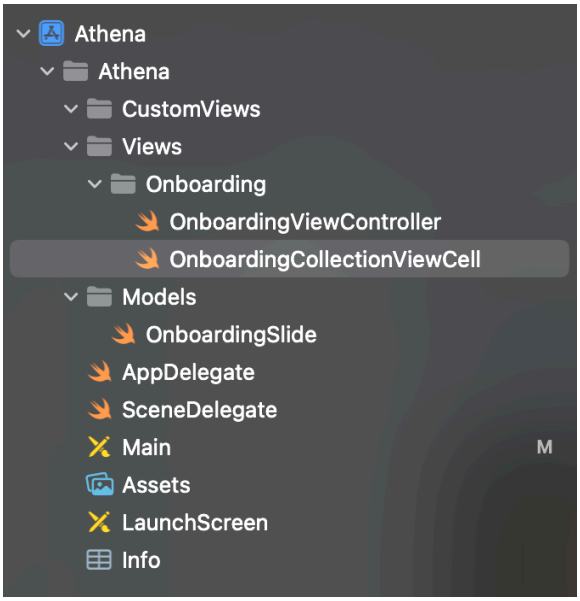
   Over the last two weeks I have been slowly getting used to Swift and Xcode. I started by just experimenting with the two different interfaces and learning where different aspects are located. I then connected my project to Github and learned how to commit/push my code. After tinkering with Xcode, I have moved into actual coding. I have been following along with a tutorial video and creating a food order app in order to learn Swift basics. Along the way, I have

also been working on my own project in congruence so I don't forget how to do certain things. Overall I have created a launch screen and am now working on the slides which are more than halfway done.

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.)

First iPhone image is the main storyboard where the slideshow design is located. It has an image collection up top with two labels and a next button below

The second iPhone is the launch screen story storyboard and contains the loading screen design.



To the left is the collection of groups and code files within my project so far

Below is a struct class that creates that designates the structure of each element in the instruction slides

```
8  import UIKit
9
10 // Struct representing an individual slide in the onboarding flow
11 struct OnboardingSlide {
12     let title: String          // Title of the slide
13     let description: String    // Description of the slide
14     let image: UIImage         // Image displayed on the slide
15 }
```

```swift
// Importing the UIKit framework which provides fundamental building blocks for iOS app development
import UIKit

// Declaring a class named OnboardingViewController, subclassed from UIViewController, responsible for managing the onboarding flow
class OnboardingViewController: UIViewController {

    // Declaring IBOutlet properties representing UI elements
    @IBOutlet weak var collectionView: UICollectionView! // UICollectionView to display slides
    @IBOutlet weak var nextButton: UIButton! // Button to proceed to the next slide
    @IBOutlet weak var pageControl: UIPageControl! // Page control to indicate the current slide

    // Declaring a property to hold onboarding slides
    var slides: [OnboardingSlide] = []

    // Method called after the view controller's view is loaded into memory
    override func viewDidLoad() {
        super.viewDidLoad()

        // Setting up the collection view delegate and data source
        collectionView.delegate = self
        collectionView.dataSource = self

        // Initializing the array of slides with dummy data
        slides = [
            OnboardingSlide(title: "Title 1", description: "Description 1", image: ♨),
            OnboardingSlide(title: "Title 2", description: "Description 2", image: ♨),
            OnboardingSlide(title: "Title 3", description: "Description 3", image: ♨)
        ]
    }

    // Action method called when the next button is clicked
    @IBAction func nextBtnClicked(_ sender: Any) {
        // Implement code to handle navigation to the next slide
    }

}
// Extension of OnboardingViewController to conform to UICollectionViewDelegate, UICollectionViewDataSource, and UICollectionViewDelegateFlowLayout protocols
extension OnboardingViewController: UICollectionViewDelegate, UICollectionViewDataSource, UICollectionViewDelegateFlowLayout {

    // Method to specify the number of items in the collection view
    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
        return slides.count
    }

    // Method to configure and return cells for the collection view
    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
        let cell = collectionView.dequeueReusableCell(withReuseIdentifier: OnboardingCollectionViewCell.identifier, for: indexPath) as! OnboardingCollectionViewCell
        cell.setup(slides[indexPath.row])
        return cell
    }

    // Method to specify the size of items in the collection view
    func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, sizeForItemAt indexPath: IndexPath) -> CGSize {
        return CGSize(width: collectionView.frame.width, height: collectionView.frame.height)
    }
}
```

Above is the onboarding view controller class that controls what the main storyboard does

```swift
import UIKit

// Custom collection view cell for displaying onboarding slides
class OnboardingCollectionViewCell: UICollectionViewCell {

    // Static property to define a reusable identifier for the cell
    static let identifier = String(describing: OnboardingCollectionViewCell.self)

    // Outlets for UI elements within the cell
    @IBOutlet weak var slideimageView: UIImageView!      // Image view for the slide image
    @IBOutlet weak var slideTitle: UILabel!              // Label for the slide title
    @IBOutlet weak var slideDescription: UILabel!        // Label for the slide description

    // Method to configure the cell with data from an OnboardingSlide
    func setup(_ slide: OnboardingSlide) {
        slideimageView.image = slide.image            // Set the image view's image
        slideTitle.text = slide.title                 // Set the title label's text
        slideDescription.text = slide.description     // Set the description label's text
    }
}
```

Above is the class that controls the cells of the collection view and designates what type of data each cell in the collection view displays

I attached a screen recording to the classroom showing what this code outputs.

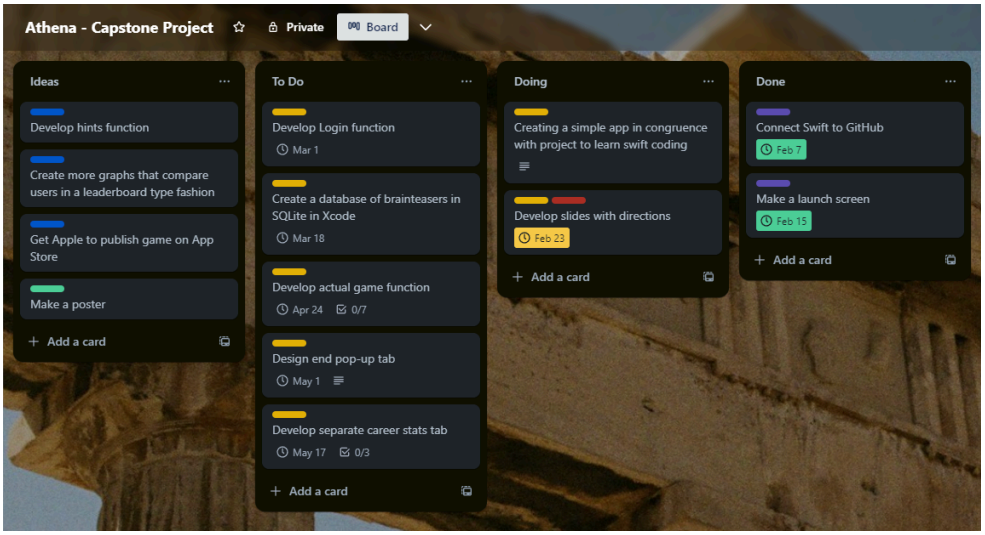
4. **Difficulties Encountered this Progress Period** (Provide detailed information on the difficulties and issues that you encountered in the reporting weeks. Discuss mitigation strategies for how you got around or plan to get around these issues.)

Over the last two weeks I have run into several difficulties, but the main two were my project had a hard time pushing its data to github and the fact that my interface is slightly different then what is used in the tutorial video.

When I watched a video on connecting Xcode to github pushing worked perfectly for the guy in the video, but I kept getting errors. I tried to search the web but nothing worked. When I watched the video for probably the fifth time to try to spot my error I realized the guy did not create a readme file. After creating a new repository and not creating a readme my code pushed perfectly.

The opposing interfaces between me and the tutorial also caused problems. My interface is three generations newer, so I keep having to look up where I can do the same thing they are doing in the video. It's not that big of a deal, though, and I'm starting to get a hang of where the best places to look for more info are.

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/ldYqXQeq/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 will continue to work on finishing up my instructions slideshow which should be done by Monday. Then, I will work on making a login function for all of next week. The last thing I hope to do before the next report is start on creating my database of brain teasers.

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.)