IvanCaldwell / **ios-sprint6-challenge**
forked from LambdaSchool/ios-sprint6-challenge

Sprint challenge for Sprint 6 of Lambda's iOS course

Edit

Manage topics

5 commits | **1** branch | **0** releases | **2** contributors

Branch: **master** ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

This branch is even with LambdaSchool:master. Pull request Compare

SpencerCurtis Update README.md Latest commit `204445d` on Oct 5, 2018

Sprint 6 Challenge	Add starter project with locked and unlocked images. Add README with …	5 months ago
.gitignore	Initial commit	6 months ago
README.md	Update README.md	3 months ago

README.md

Lock Control

Instructions

Please read this entire README to make sure you understand what is expected of you before you begin.

This sprint challenge is designed to ensure that you are competent with the concepts taught throughout Sprint 6.

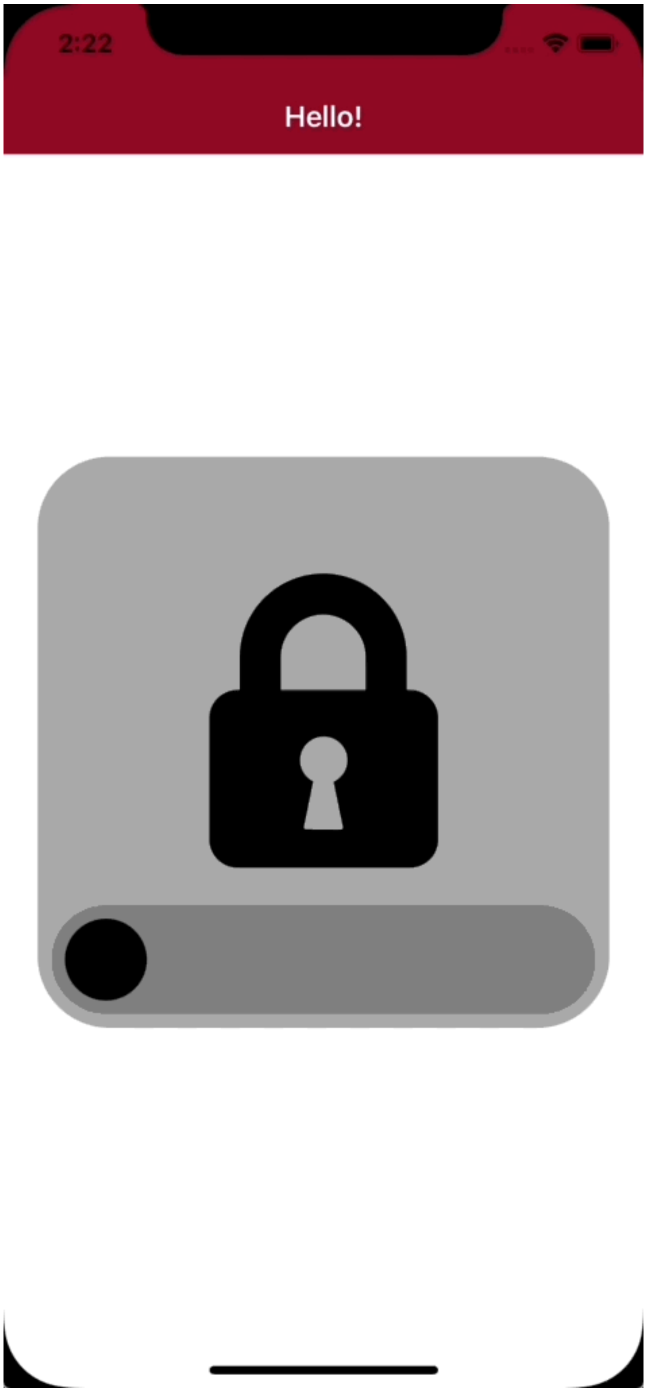
Begin by forking this repository. Clone your forked repository to your machine. Use the provided Xcode project in the repository as it contains the two images you will need. Commit as appropriate while you work. Push your final project to GitHub, then create a pull request back to this original repository.

You will have 3 hours to complete this sprint challenge

Good luck!

Screen Recording

Please view the screen recordings so you will know what your finished project should look like:



(The gif is fairly large in size. It may take a few seconds for it to appear)

Requirements

The goal of this Sprint Challenge is to create a custom `UIControl` . The control is a lock that is unlocked by dragging a subview. This similar to the old "swipe to unlock" feature the iOS device lockscreen used to have. Along with that, you will demonstrate your knowledge of `UIAppearance` by applying a simple theme to the app.

The requirements for this project are as follows:

1. Your view controller should have:
   - A navigation bar.
   - A bar button item in the navigation bar that will reset the lock.

    - A view for your lock control.

2. Apply a theme for your application using `UIAppearance` . This should change the look of at least the navigation bar and the bar button item.
3. Implement the lock custom `UIControl` . The control should:
    - Show an image indicating whether it is locked or unlocked. You will find the images in the assets folder.
    - Move according to the movement of the touch if it's touching the thumb view (the black circle in the screen recording). See the screen recording for how the view should move.
    - Upon ending the touch, the thumb view should animate by springing back to the starting position if it's below 80% of the way down the track, or unlock if it's past 80% of the track. Change the lock's image to the `unlocked` image from the assets folder.
    - Once unlocked, user interaction should be disabled. Create a `reset` method that will "lock" the control by putting the thumb view back to its starting position and resetting the image back to the locked image.
    - The reset button should appear when the button is unlocked, and disappear when the reset button is tapped. This should be animated.