

IvanCaldwell / ios-planets-objc
forked from LambdaSchool/ios-planets-objc

No description, website, or topics provided.

Edit

[Manage topics](#)

3 commits

1 branch

0 releases

1 contributor

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 e3892be on Sep 30, 2018

[README.md](#)

Update README.md

5 months ago

README.md



Planets Obj-C

Introduction

The goal of this project is to revisit and remake the Planets project that you've seen before, this time using Objective-C.

Instructions

Please fork and clone this repository, then create an Xcode project in it. Commit regularly as you complete the requirements in this project.

Part 1 - Storyboard

When adding new files, remember to give them a prefix!

1. In the "Main.storyboard", add a navigation controller. Set it to be your initial view controller.
2. Add a collection view controller scene and create an embed segue from the navigation controller to it. Create a Cocoa Touch file for the collection view controller (with your prefix) called `PlanetsCollectionViewController`.
3. The collection view cell should have an image view to show the planet, and a label to show the name of the planet. Create a Cocoa Touch file for the cell and create any outlets and actions you may need.
4. Add a bar button item to the navigation item of the collection view controller. Set its title to "Settings".
5. Add a `UIViewController` scene, then add the following to it:
 - o A `UIButton` titled "Done". Constrain it to the top-right of the view controller.
 - o A `UILabel` with the text "Is Pluto a planet?"
 - o A `UISwitch` that will toggle whether the user wants Pluto to be counted as a planet or not.
6. Create a Cocoa Touch Subclass for this view controller. Call it `SettingsViewController` (again, with your prefix), then create any outlets and/or actions you may need.
7. Create a modally presented segue from the "Settings" button in the collection view controller scene to the `SettingsViewController` scene.

Part 2 - Model/Model Controller

1. Create a model object that represents a planet. It should include the name and image of the planet.
2. Create a model controller that has two hardcoded arrays of planets; the first without Pluto, and the second with Pluto.

Part 3 - View Controller

1. In the `SettingsViewController` , implement the ability to save whether the user would like to show Pluto using `NSUserDefaults` .
2. Implement the required `UICollectionViewDataSource` methods in the `PlanetsCollectionViewController` .

Make sure that the collection view updates showing Pluto or not according to the value saved in `NSUserDefaults` from the switch in the `SettingsViewController` .

Go Further

1. Set the collection view to scroll horizontally, and have the size of the cells be scaled in proportion to the other planet cells. E.g. Jupiter is the largest so its cell would be the largest, while the Mercury cell would be the smallest. [This page](#) includes the size of each planet.