Piscine Swift - Day 02

Daily Quests & Pomodoro

Michael BRAVE mbrave@student.42.us.org

42 Staff pedago@42.fr

Summary: This document contains the subject for Day 02 for the “Piscine Swift” from 42

Contents

Chapter I

Foreword

“You must vie with time’s swiftness in the speed of using it, and, as from a torrent that rushes by and will not always flow, you must drink quickly.” — Seneca, On the Shortness of Life

Chapter II

General Instructions

- Only this document will serve as reference. Do not trust rumors.
- Read carefully the whole subject before beginning.
- Watch out! This document could potentially change up to an hour before submission.
- This project will be corrected by humans only.
- This course is designed to build on previous days' concepts, try your hardest to finish everyday.
- Each day culminates in a portfolio piece, if you finish the day this is something you can use to get hired.
- When submitting, submit the folder of the Xcode project.
- Only the work submitted on the repository will be accounted for during peer-2-peer correction.
- Here it is the official manual of Swift and the Swift Standard Library
- It is forbidden to use other libraries, packages, pods, etc. Unless otherwise stated in the project.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- You can discuss on the Piscine forum of your Intra!
- By Odin, by Thor! Use your brain!!!

Chapter III

Introduction

Today we are building some productivity apps for ourselves, to help us set time limits and to set habits for ourselves. In doing this we will learn how to create elements of an app that adapt to each other, are reshaped by the content inside them and how to add and remove elements to our pages.

Each assignment was designed to build on the previous assignments knowledge and culminate in a portfolio piece. If we finish the day we will have portfolio pieces that help us to create daily habits, a stopwatch and pomodoro timer.

Hint: some keywords that may help you in your learning: Tableview, Multi Page Navigation, Cell Editing, Dynamic Elements and Saving Data

Chapter IV

Exercise 00 : Stopwatch

Exercise : 00
Stopwatch
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit
Notes : n/a

We are creating what we call a countup timer one that has numeric display, a pause button, a start button and a reset button. When the start button is clicked it should start counting the seconds, minutes and hours as time goes on and visually display these numbers. The pause button should stop the counting, and continue it when pressed again. Reset should set the clock value back to zero.

Hint: Use Swift’s Timer Class

Chapter V

Exercise 01 : Pomodoro Timer

Exercise : 01
Pomodoro Timer
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit
Notes : n/a

Similar to the last assignment but now we are creating a countdown timer. It should have a numeric display, a pause button, a start button and a reset button. When the start button is clicked it should start counting down from 25 min and visually display these numbers down to the second. The pause button should stop the counting, and continue it when pressed again. Reset should set the clock value back to 25 min.

Hint: Use Swift’s Timer Class

Chapter VI

Exercise 02: Editing Lists

Exercise : 02
Editing Lists
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit, UICollectionView
Notes : n/a

In this assignment we are creating the basic functionality of a checklist app. We need to be able to add elements to the list and to delete them. This needs to be visually displayed and functional.

Hint: Cells

Chapter VII

Exercise 03: Dynamic Cell Content

Exercise : 03
Dynamic Cell Content
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit, UICollectionView
Notes : n/a

We need elements that are added to be able to grow dynamically based on what's inside them. As elements grow they affect each other. We need to figure out a way to do this that seems natural for the app. To demonstrate this we will create a todo list, that allows for more than a single line of text for each entry. This should work in both landscape and portrait mode.

Chapter VIII

Exercise 04: Expanding Cells

Exercise : 04
Expanding Cells
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit, UITableViewController
Notes : n/a

So what if we have a title that hides information? We want to click on it and show the hidden info, which stretches and modifies all other data and how it will fit on the screen. We should be able to click on one item in the list to reveal a sublist, this should expand and push down the other elements off the screen, be scrollable, and we should be able to add elements and sub elements to each list.

Chapter XI

Exercise 05: Daily Quests

Exercise : 04
Daily Quests
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit, DateInterval
Notes : n/a

Todays portfolio piece.

Now we put everything we've learned together. We are creating a daily todo app, this is similar in practice to "dailies" from popular MMORPGs. Anything that we wish to do everyday as a habit we will add to the list. On click we will start a timer unique to each element, on completion of the timer it will log that item as completed for the day. The tasks will reset daily. This will allow us to view our collective completion for each daily task in a weekly and monthly view. As an example, workout everyday for 30 min, we click start when we begin our workout and it logs it everyday that we do. We want to be able to see if we have done it every day for a month.

Chapter X

Bonus : Screen Transition

Bonus
Screen Transition
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit, CABasicAnimation
Notes : n/a

Launch logo transition. Take a logo, use it like a loading screen, as it expands eventually to reveal the app behind it.

Hint: Mask, size, alpha