Piscine Swift - Day 01

Tappy Swimmer

Michael BRAVE mbrave@student.42.us.org

42 Staff pedago@42.fr

Summary: This document contains the subject for Day 00 for the “Piscine Swift” from 42

Contents

Chapter I

Foreword

Object Oriented Programming

Jeff Goodell:
Would you explain, in simple terms, exactly what object-oriented software is?

Steve Jobs:
Objects are like people. They're living, breathing things that have knowledge inside them about how to do things and have memory inside them so they can remember things. And rather than interacting with them at a very low level, you interact with them at a very high level of abstraction, like we're doing right here.

Here's an example: If I'm your laundry object, you can give me your dirty clothes and send me a message that says, "Can you get my clothes laundered, please." I happen to know where the best laundry place in San Francisco is. And I speak English, and I have dollars in my pockets. So I go out and hail a taxicab and tell the driver to take me to this place in San Francisco. I go get your clothes laundered, I jump back in the cab, I get back here. I give you your clean clothes and say, "Here are your clean clothes."

You have no idea how I did that. You have no knowledge of the laundry place. Maybe you speak French, and you can't even hail a taxi. You can't pay for one, you don't have dollars in your pocket. Yet, I knew how to do all of that. And you didn't have to know any of it. All that complexity was hidden inside of me, and we were able to interact at a very high level of abstraction. That's what objects are. They encapsulate complexity, and the interfaces to that complexity are high level.

From a 1994 Rolling Stone interview of Steve Jobs

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 working on learning the basics of object oriented programming.

Today we will be learning how to use spritekit, object oriented programming, using timers, and then putting it all together.

Each assignment was designed to build on the previous assignments knowledge and culminate in a portfolio piece. At the end of the day we will be building a game named tappy swimmer which is highly inspired by flappy bird.

So go forth and learn about OOP (encapsulation, abstraction, inheritance & polymorphism).

Chapter IV

Exercise 00: Collisions & Objects

Exercise : 00
Collisions & Objects
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit (CAEmitterLayer / CAEmitterCell)
Notes : n/a

Now we're cooking with objects. We are creating a simple collision game, your character will move, and have to not crash into some things, while collecting others. In game speak this will be a very simple top down game where your character moves automatically and enemies and coins spawn randomly. There should be a score for points and an end to the game on crashing.

All are game objects and will inherit some similar classes, but will have different interpretations of how to handle that.

Chapter V

Exercise 01 : SpriteKit 101

Exercise : 01
SpriteKit 101
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit, SpriteKit
Notes : n/a

In order to learn spritekit we are going to make a spaceship that flies away from us. To zoom form large to small, then disappears (is no longer visible after certain constraints are met).

Chapter VI

Exercise 02 : Sisyphus

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

Sisyphus, in Greek mythology, the cunning king of Corinth who was punished in Hades by having repeatedly to roll a huge stone up a hill only to have it roll down again as soon as he had brought it to the summit.

We are going to make a button that counts when we click it, and a timer that decreases the count over time so that we are exercising futility, a lot like how sisyphus did.

You will need to show

1. A button to click that causes the number to increase
2. A visualization of the number value
3. A timer that decreases the number over time.

Chapter VII

Exercise 03: Tappy Swimmer

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

Now we pull it all together. We are creating a game that causes a character to rise as we tap and sink over time. There will be obstacles to avoid and collet. This will function much like the game flappy bird.

Chapter XI

Bonus : Adding Animations & Graphics

Bonus
Adding Animations & Graphics
Files to turn in: .xcodeproj and all necessary files
Allowed functions : Swift Standard Library, UIKit
Notes : n/a

Although the game is functional it might be cool if animations helped it to come alive. Do your best, make it real.