Piscine Swift - Day 05

Tools & Music

Michael BRAVE mbrave@student.42.us.org

42 Staff pedago@42.fr

Summary: This document contains the subject for Day for the "Piscine Swift" from 42

Contents

I Foreword

II General Instructions

III Introduction

IV Exercise 00: CocoaPods

V Exercise 01: CocoaPods 2 - Making Your Own

VI Exercise 02: Music API

VII Exercise 03: Save To Device

VIII Exercise 04: Simple Piano

XI Bonus: Additional Functionality

Chapter I

Foreword

Here is a timeline of technological innovations that influenced the creation of music from Wikipedia's Timeline of music technology

- 1951: Pultec introduces the first passive program equalizer, the EQP-1
- 1952: Harry F. Olson and Herbert Belar invent the RCA Synthesizer
- 1952 : Osmand Kendal develops the Composer-Tron for the Canadian branch of the Marconi Wireless Company
- 1955 : Ampex develops "Sel-Sync" (Selective Synchronous Recording), making audio overdubbing practical
- 1956: Les Paul makes the first 8-track recordings using the "sel-sync" method
- 1956 : Raymond Scott develops the Clavivox
- 1958 : First commercial stereo disk recordings produced by Audio Fidelity
- 1958: Evgeny Murzin along with several colleagues create the ANS synthesizer
- 1958: At Texas Instruments, Jack Kilby creates the first integrated circuit
- 1959 : Daphne Oram develops a programming technique known as Oramics
- 1959 : Wurlitzer manufactures The Sideman, the first commercial electro-mechanical drum machine
- 1963: Keio Electronics (later Korg) produces the DA-20, an early drum machine
- 1963: The Mellotron starts to be manufactured in London
- 1963 : Phillips introduces the Compact Cassette tape format
- 1963: Paul Ketoff designs the SynKet
- 1964 : Ikutaro Kakehashi debuts Ace Tone R-1 Rhythm Ace, the first electronic drum
- 1964: The Moog synthesizer is released
- 1965 : Nippon Columbia patents an early electronic drum machine
- 1966: Korg releases Donca-Matic DE-20, an early electronic drum machine
- 1967 : Ace Tone releases FR-1 Rhythm Ace, the first electronic drum machine to enter popular music
- 1967: First PCM recorder developed by NHK
- 1968 : Sharp engineer Tadashi Sasaki conceives single-chip microprocessor
- 1968 : Release of Shin-ei's Uni-Vibe, designed by Fumio Mieda, an effects pedal with phase shift and chorus effects
- 1968: King Tubby pioneers dub music, an early form of popular electronic music
- 1969 : Matsushita engineer Shuichi Obata invents first direct-drive turntable, Technics SP-10
- 1970: ARP 2600 is manufactured
- 1971 : Busicom's Masatoshi Shima and Intel's Federico Faggin complete 4004, the first commercial microprocessor
- 1972 : Sord Computer Corporation develop Sord SMP80/08, an early microcomputer
- 1973: Yamaha release Yamaha GX-1, the first polyphonic synthesizer
- 1974 : Yamaha build first digital synthesizer
- 1976: Boss, a Roland subsidiary, release Boss CE-1 Chorus Ensemble, the first chorus pedal

- 1977 : Roland release MC-8 Microcomposer, an early microprocessor-driven CV/Gate digital sequencer
- 1977: Apple founder Steve Jobs introduces Apple II, an early home computer
- 1977: Sord Computer Corporation introduces Sord M200, an early home computer
- 1977: Panafacom releases the Lkit-16, an early 16-bit microcomputer
- 1978 : Roland releases CR-78, the first microprocessor-driven drum machine
- 1979 : Casio releases VL-1,[26] the first commercial digital synthesizer
- 1980 : Fujio Masuoka invents flash memory at Toshiba
- 1980: Roland releases TR-808, the most widely used drum machine in popular music
- 1980 : Roland introduces DCB protocol and DIN interface with TR-808
- 1980: Yamaha releases GS-1, the first FM digital synthesizer
- 1980 : Kazuo Morioka creates Firstman SQ-01, the first bass synthesizer with a music sequencer
- 1981: Roland releases TB-303, a bass synthesizer that lays the foundations for acid house music
- 1981: Roland founder Ikutaro Kakehashi conceives MIDI
- 1981 : Toshiba's LMD-649, the first PCM digital sampler, introduced with Yellow Magic Orchestra's Technodelic
- 1981: IBM introduces the IBM PC, a 16-bit personal computer
- 1982 : Sony and Philips introduce compact disc
- 1982 : First MIDI synthesizers released, Roland Jupiter-6 and Prophet 600
- 1983 : Introduction of MIDI, unveiled by Roland's Ikutaro Kakehashi and Sequential Circuits' Dave Smith
- 1983: Roland releases MSQ-700, the first MIDI sequencer
- 1983: Roland releases TR-909, the first MIDI drum machine
- 1983: Roland releases MC-202, the first groovebox
- 1983: Yamaha releases DX7, the first commercially successful digital synthesizer
- 1984 : Apple markets the Macintosh computer
- 1985 : Atari releases the Atari ST computer, designed by Shiraz Shivji
- 1985 : Akai releases the Akai S612, a digital sampler
- 1986: The first digital consoles appear
- 1987: Digidesign markets Sound Tools
- 1988: Akai introduces the Music Production Controller (MPC) series of digital samplers
- 1994: Yamaha unveils the ProMix 01

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 <u>official manual of Swift</u> and the <u>Swift Standard Library</u>
- 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 learning some other practical tools to be used alongside swift, namely dependency injection, the foundations of firebase and how to use tools that other people have made. We will explore this by playing with music, and making a simple piano app.

Chapter IV

Exercise 00: CocoaPods

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

CocoaPods is about dependencies. We are going to learn how to use dependency injection for CocoaPods. We are going to use a MusicTheory pod - https://cocoapods.org/pods/MusicTheory. We will use that to build a sequencer, which will play a loop of notes.

Chapter V

Exercise 01: CocoaPods 2 - Making Your Own

Exercise: 01

CocoaPods 2 - Making Your Own

Files to turn in: .xcodeproj and all necessary files

Allowed functions: Swift Standard Library, UIKit, Firebase Framework SDK

Notes: n/a

We will create a default template of a file to use in all future projects. You will need to set up both a podspec and a LICENSE. For now we will only add what we need to add firebase functionality (useful in future days of the piscine). You will need analytics, a firebase header, module map and firebase configuration.

Chapter VI

Exercise 02: Playing With Sounds

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

We are going to use AudioKit to implement sounds. This assignment will be considered complete if we have two buttons that play two unique sounds. We are also building what's called an arpeggiator which plays a sequence of notes according to sequential parameters.

Chapter VII

Exercise 03: Save To Device

Exercise: 03

Save To Device

Files to turn in: .xcodeproj and all necessary files

 $Allowed\ functions: Swift\ Standard\ Library,\ UIKit,\ AVCapture Audio Data Output,\ AVCapture Output,$

AVCaptureAudioFileOutput

Notes: n/a

So now that we have at least two sounds we want to be able to record what we create. We need to be able to start, stop and pause recording of audio. This output audio needs to be saved to the local storage of the device where it can be played at a later time.

Hint: AVFoundation

Chapter VIII

Exercise 04: Simple Piano

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

Now we pull it all together to build a portfolio piece. We will build a simple piano, that has multiple keys, options for tones, and the ability to record and save to disk.

Chapter XI

Bonus : Additional Functionality

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

Add new functionality, maybe have the piano play different instruments, maybe implement an oscillator. Make it interesting.