



Piscine iOS Swift - Day 05

Kanto

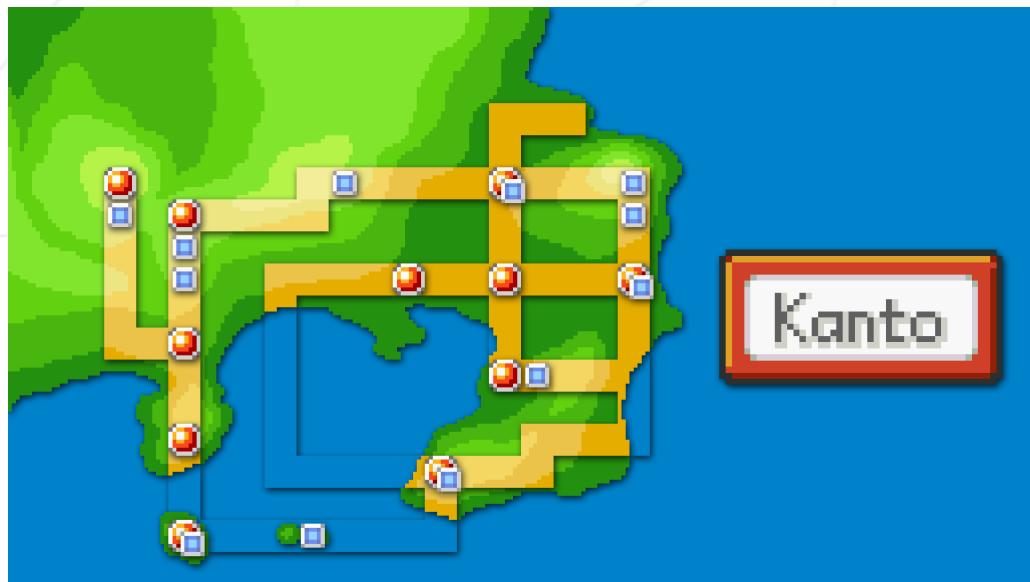
*Summary: This document contains the subject for Day 05 of the iOS Swift piscine of
the 42*

Contents

I	Preamble	2
II	Instructions	3
III	Introduction	5
IV	Exercise 00: Tab Bar	6
V	Exercise 01: Table View	7
VI	Exercise 02: MapKit	8
VII	Exercise 03: SegmentControlBar	9
VIII	Exercise 04: Geo-tracking	10
IX	Exercise 05: Selecting a location	11
X	Exercise 06: Pin colors	12

Chapter I

Preamble



Chapter II

Instructions

- Only this page will serve as reference. Do not trust rumors.
- Read attentively the whole document before beginning.
- Your exercises will be corrected by your piscine colleagues.
- The document can be relied upon, do not blindly trust the demos which can contain not required additions.
- You will have to deliver an app every day (except for Day 01) on your git repository, where you deliver the file of the Xcode project.
- Here it is the official manual of [Swift](#) and of [Swift Standard Library](#)
- It is forbidden to use other libraries, packages, pods...before Day 07
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Think about discussing on the forum Piscine of your Intra!
- Use your brain!!!



The videos on Intra were produced before Swift 3. Remove the prefix "NS" which you see in front of the class/struct/function in the code in the videos to use them in Swift 3.



Intra indicates the date and the hour of closing for your repositories. This date and hour also corresponds to the beginning of the peer-evaluation period for the corresponding piscine day. This peer-evaluation period lasts exactly 24h. After 24h passed, your missing peer grades will be completed with a 0.

SPECIFIC instructions. Will only apply to specific days

Chapter III

Introduction

Geo-tracking is a weapon of choice in any good iOS developer's kit. You have to know how to use it.

Apple provides different frameworks such as **MapKit** that allows you to use a map a very simple way or **CoreLocation** that helps you manage the tracking of a user.

Those frameworks will become your best friends today.

You will create an geo-tracking application for several locations: utilisant :

A TabBarController: to set up your different views.

An MKMapView: for the map.

A CLLocationManager: to geo-track the user.

A SegmentedControlBar: to change the map style.

Some MKAnnotationView: to customize the map's pop up.

Chapter IV

Exercise 00: Tab Bar

	Exercice : 00
	Tab Bar
	Files to turn in : Swift Standard Library, UIKit
	Authorised functions : n/a
	Notes : n/a

Start creating a project using *Tabbed Application*.

Personalize the icons of both views contained in the *MainStoryboard*.

One of the list will feature several places in a **table view**. The other will be the map on which these places will be displayed.

Chapter V

Exercise 01: Table View

	Exercice : 01
	Table View
	Files to turn in : Swift Standard Library, UIKit
	Authorised functions : n/a
	Notes : n/a

You will now add a **table view** to one of those views. It will have to display at least 3 different places.



To ease the development in the following exercises, try to organize your data correctly. You still can add new files to your project.

42

Saint Ouen

Grenoble

Reims

Moldavie

Circuit

Map List More

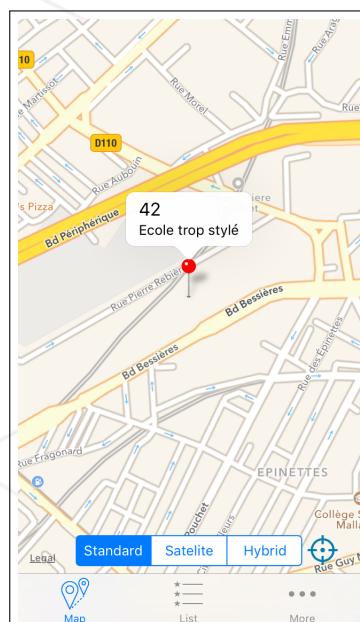
Chapter VI

Exercise 02: MapKit

	Exercice : 02
	MapKit
	Files to turn in : Swift Standard Library, UIKit, MapKit
	Authorised functions : n/a
	Notes : n/a

Let's get to it. With this exercise, you will have to:

- Add a map in the second view.
- Display a *pin* on Ecole 42.
- Add a title and a subtitle to this *pin*. These informations must appear when you click the pin.
- When you get to the map, it must be zoomed on the Ecole 42.



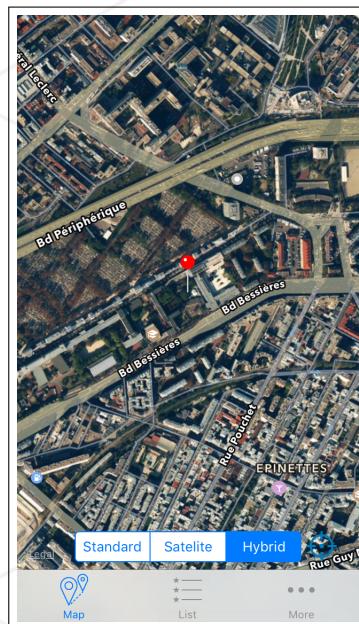
Chapter VII

Exercise 03: SegmentControlBar

	Exercice : 03
	SegmentControlBar
	Files to turn in : Swift Standard Library, UIKit, MapKit
	Authorised functions : n/a
	Notes : n/a

Now you've managed to display the map as you intended to, you can add a **segmented control bar** that must allow you to select the map mode.

There are 3 map display modes: *Hybrid*, *Satellite* or *Standard*. The **segmented control bar** must allow you to change the mode anytime.



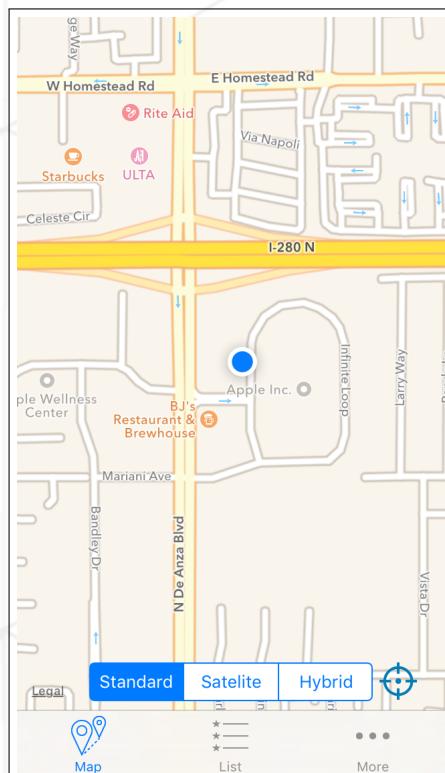
Chapter VIII

Exercise 04: Geo-tracking

	Exercice : 04
	Geo-tracking
	Files to turn in : Swift Standard Library, UIKit, MapKit, CoreLocation
	Authorised functions : n/a
	Notes : n/a

Now your map is operational, it would be nice to have a button to help you geo-track yourself.

Add a button that must recenter the map on your position, setting the scale so it's zoomed on you.



Chapter IX

Exercise 05: Selecting a location

	Exercice : 05
	Selecting a location
	Files to turn in : Swift Standard Library, UIKit, MapKit, CoreLocation
	Authorised functions : n/a
	Notes : n/a

You must now make you list functional passing variables between your views:

- All the locations in your list must be featured in the map's view as *pins* with a title and a subtitle.
- One click on one of your map's locations must send you back on the map's view and zoom on the selected location.



You must not instantiate a controller and a view again.

Chapter X

Exercise 06: Pin colors

	Exercice : 06
	Pin colors
	Files to turn in : Swift Standard Library, UIKit, MapKit, CoreLocation
	Authorised functions : n/a
	Notes : n/a

You will now personalize the map's *pins* giving them different colors.

