

# Piscine iOS Swift - Day 07 Siri

Résumé: Ce document contient le sujet du Day 07 de la piscine iOS Swift de 42

# Table des matières

1	1 Teambule	
II	Consignes	3
III	Introduction	4
IV	Exercice 00 : Installation de Cocoapods	5
$\mathbf{V}$	${\bf Exercice}~{\bf 01: First View Controller}$	7
$\mathbf{VI}$	Exercice 02 : pod 'RecastAI'	8
VII	Exercice 03 : pod 'Dark Sky'	9
VIII	${\bf Exercice~04:pod~'JSQMessages View Controller'}$	10

#### Chapitre I

#### Préambule

Voici la page wikipedia sur l'Intelligence Artificielle:

Le terme « intelligence artificielle », créé par John McCarthy, est souvent abrégé par le sigle « I.A. » (ou « A.I. » en anglais, pour Artificial Intelligence). Il est défini par l'un de ses créateurs, Marvin Lee Minsky, comme « la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique ». On y trouve donc le côté « artificiel » atteint par l'usage des ordinateurs ou de processus électroniques élaborés et le côté « intelligence » associé à son but d'imiter le comportement. Cette imitation peut se faire dans le raisonnement, par exemple dans les jeux ou la pratique des mathématiques, dans la compréhension des langues naturelles, dans la perception : visuelle (interprétation des images et des scènes), auditive (compréhension du langage parlé) ou par d'autres capteurs, dans la commande d'un robot dans un milieu inconnu ou hostile.

Même si elles respectent globalement la définition de Minsky, il existe un certain nombre de définitions différentes de l'IA qui varient sur deux points fondamentaux :

Les définitions qui lient la définition de l'IA à un aspect humain de l'intelligence, et celles qui la lient à un modèle idéal d'intelligence, non forcément humaine, nommée rationalité. Les définitions qui insistent sur le fait que l'IA a pour but d'avoir toutes les apparences de l'intelligence (humaine ou rationnelle), et celles qui insistent sur le fait que le fonctionnement interne du système d'IA doit ressembler également à celui de l'être humain et être au moins aussi rationnel.

#### Chapitre II

#### Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Lisez attentivement l'integralité du sujet avant de commencer.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Vos exercices seront corrigés par vos camarades de piscine.
- Le sujet fait foi, ne vous fiez pas toujours à la lettre aux demos qui peuvent contenir des ajouts supplémentaires non demandés.
- Vous devrez rendre une app par jour (sauf pour le Day 01) sur votre depot git, rendez le dossier du projet Xcode.
- Voici le manuel officiel de Swift et de Swift Standard Library
- Il est interdit d'utiliser d'autres librairies, packages, pods... avant le Day 07
- Vous avez une question? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Pensez à discuter sur le forum Piscine de votre Intra!
- Réfléchissez. Par pitié, par Odin! Nom d'une pipe.



Les videos sur l'intra ont été tournées avant Swift 3. Enlevez le prefix "NS" que vous voyez devant les class/struct/function dans le code des videos pour les utiliser en Swift 3.



L'intra indique la date et l'heure de fermeture de vos dépots. Cette date et heure correspond également au début de la période de peer-evaluation pour le jour de piscine correspondant. Cette période de peer-evaluation dure exactement 24h. Une fois ces 24h passées, vos notes peer manquantes seront complétées par des 0.

#### Chapitre III

#### Introduction

Vous le savez surement mais les développeurs sont pour beaucoup des feignants et à juste titre : on ne va pas ré-inventer la roue tous les jours si celle-ci roule déjà parfaitement. Aujourd'hui nous allons vous présenter les pods grace à Cocoapods!

Mais un pod c'est quoi au juste? Un pod ce n'est pas Planet of Death ni The Breeders. Un pod c'est un paquet qui est donc géré par un gestionnaire de paquets, ici Cocoapods.

Dans cette journée vous allez apprendre à utiliser des pods. C'est à dire à les installer ainsi qu'à les utiliser. L'objectif de la journée sera de créer un robot plus communément appelé **bot** (prononcé 'botte') qui vous donnera la météo pour une ville donnée.

Pour cela vous allez utiliser deux API provenant de deux services distincts sur lesquels ils vous faudra créer un compte.

- Recast.AI: une API vous apportera la brique Intelligence Artificielle pour votre bot.
- Dark Sky : une API ouverte qui vous permettra de récupérer la météo pour une latitude et une longitude donnée. (Anciennement Forecast.IO)

Voici les documentations dont vous aurez besoin pour finir la journée :

- Doc Cocoapods
- Pod Recast.AI
- Pod Dark Sky
- Pod JSQMessagesViewController qui vous permettra d'afficher votre conversation sous un format de messagerie.

# Chapitre IV

Exercice 00 : Installation de Cocoapods



Exercice: 00

Installation de Cocoapods

Fichiers à rendre : .xcodeproj et tous les fichiers necessaires Fonctions Autorisées : Swift Standard Library, UIKit, Cocoapods

Remarques : n/a

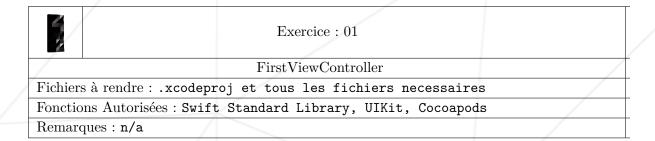
Pour commencer il vous faut installer Cocoapods.

Faites un tour sur le site de Cocoapods et suivez la procédure. Vérifiez que Cocoapods est bien installé en tapant 'pod' dans la console :

```
>pod
Usage:
     $ pod COMMAND
       CocoaPods, the Cocoa library package manager.
Commands:
     + cache
                     Manipulate the CocoaPods cache
     + init
                     Generate a Podfile for the current directory.
                     Install project dependencies to Podfile.lock versions Inter-process communication  \begin{tabular}{ll} \hline \end{tabular} \label{table_podfile}
     + install
     + ipc
     + lib
                     Develop pods
     + list
                     List pods
     + outdated
                     Show outdated project dependencies
     + plugins
                      Show available CocoaPods plugins
                     Manage spec-repositories
Search for pods.
Setup the CocoaPods environment
     + repo
     + search
     + setup
                     Manage pod specs
     + spec
                      Interact with the CocoaPods API (e.g. publishing new specs)
     + trunk
     + try
+ update
                      Try a Pod!
                     \label{thm:policy} \mbox{\sc Update outdated project dependencies and create new Podfile.lock}
Options:
                      Show nothing
     --silent
                      Show the version of the tool
     --version
                      Show more debugging information
     --verbose
                     Show output without ANSI codes
Show help banner of specified command
     --no-ansi
     --help
```

#### Chapitre V

#### Exercice 01: FirstViewController



Pour cette journée vous aurez besoin d'un controller vous permetant de faire vos requêtes à Recast & Dark Sky.

Faites un controller contenant :

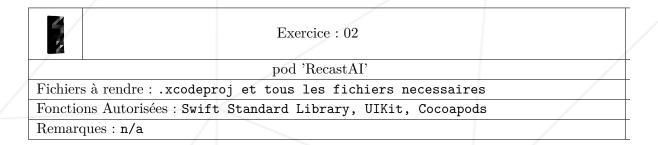
- $\bullet\,$  Un  ${\bf Bouton}$  pour faire la requête a Recast
- Un **TextField** pour écrire le texte que vous enverrez
- Un Label pour afficher la réponse



Pensez à l'Autolayout!

### Chapitre VI

Exercice 02: pod 'RecastAI'



Il est temps de créer un compte sur Recast.AI. Installez le pod 'RecastAI' pour pouvoir

effectuer des requêtes et utilisez un bot Slackbot dans la communauté avec l'intention weather puis avec le token faites une requête via le boutton en passant en parametre le text du **TextField**.

Le **label** doit afficher l'intention retournée par **Recast** ou "Error" si aucune intention n'est retournée.

### Chapitre VII

Exercice 03: pod 'Dark Sky'



Maintenant nous allons afficher la météo en fonction de la localisation retournée. Pour cela créer un compte sur Dark Sky pour obtenir un token. Installez le pod 'Dark Sky' puis faites une requête a Dark Sky une fois la requête Recast terminée. Le label doit afficher la météo retournée par l'API de Dark Sky en fonction de la localisation retournée par l'API de Recast.

## Chapitre VIII

# Exercice 04 : pod 'JSQMessagesViewController'

2	Exercice: 04	
	pod 'JSQMessagesViewController'	/
Fichiers	sàrendre: .xcodeproj et tous les fichiers necessaires	/
Fonctio	ns Autorisées : Swift Standard Library, UIKit, Cocoapods	/
Remarc	ques : n/a	/

Maintenant que tout cela est fini nous allons pouvoir peaufiner notre interface. Pour cela nous allons utiliser le pod 'JSQMessagesViewController' pour afficher nos requêtes sous forme de conversation avec notre bot.

Vous devrez aussi rajouter un boutton pour faire des requêtes avec la voix.