



Piscine iOS Swift - Day 08

CoreData

Maxime LEMORT mlemort@student.42.fr

PE LIEB plieb@student.42.fr

42 Staff pedago@42.fr

Résumé: Ce document contient le sujet du Day 08 de la piscine iOS Swift de [42](#)

Table des matières

I	Préambule	2
II	Consignes	3
III	Introduction	4
IV	Exercice 00 : Création du pod	5
V	Exercice 01 : Podspec	6
VI	Exercice 02 : xcmodeld	7
VII	Exercice 03 : Class Article	8
VIII	Exercice 04 : Class ArticleManager	9
IX	Exercice 05 : ViewController	10

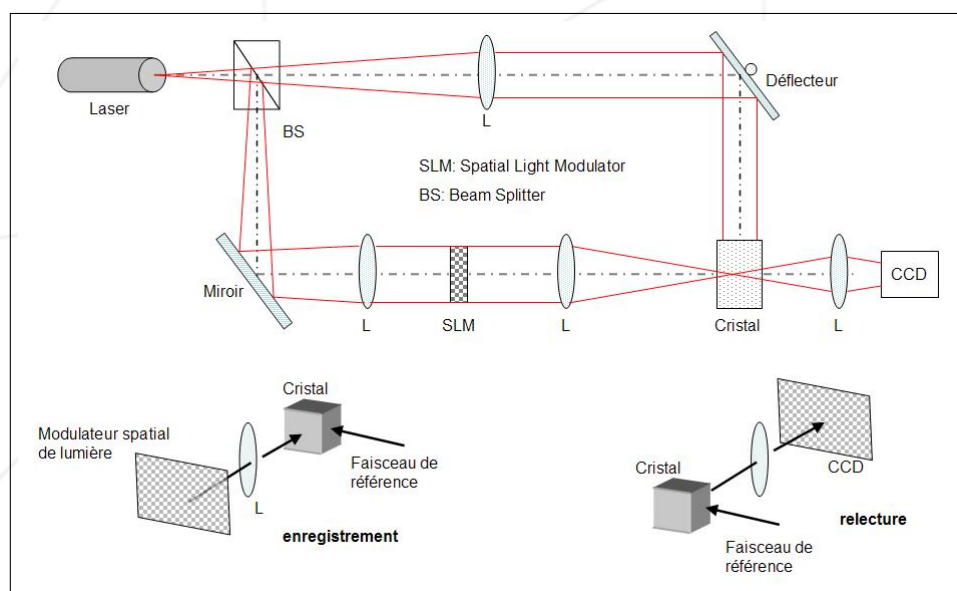
Chapitre I

Préambule

Voici la page wikipedia sur la Memoire Holographique :

La mémoire holographique est une technique de mémoire de masse utilisant l'holographie pour stocker de hautes densités de données dans des cristaux ou des polymères photosensibles.

La mémoire holographique est souvent désignée comme étant la prochaine génération de stockage optique des données. En effet, les techniques utilisées pour les CD ou les DVD atteignent leurs limites physiques (à cause de la taille des rayons d'écriture limitée par la diffraction). L'holographie permet d'utiliser le volume du support au lieu de se limiter à la surface pour enregistrer des données. De plus, les données peuvent être multiplexées dans le volume d'enregistrement en ajoutant un angle au faisceau enregistreur par rapport au faisceau de référence, ou encore en modifiant sa fréquence ou sa phase.



Chapitre II

Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Lisez attentivement l'intégralité 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](#)
- Voici le manuel officiel 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 !
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet...
- Réfléchissez. Par pitié, par Odin ! Nom d'une pipe.



L'intra indique la date et l'heure de fermeture de vos dépôts. 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

Maintenant que vous avez découvert les pods nous allons apprendre à en créer un. Toujours grace à [Cocoapods](#) !

Si vous avez déjà oublié ce qu'est un pod c'est un **paquet** qui est donc géré par un **gestionnaire de paquets** ici **Cocoapods**.


L'objectif de la journée sera d'en créer un qui utilisera le framework **CoreData** pour apprendre a utiliser la persistance des données et les modeles. Le but de la journée étant de créer un gestionnaire d'articles qui servira d'interface pour le D09.

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

- [Doc Cocoapods](#)
- [CoreData](#)

Chapitre IV

Exercice 00 : Création du pod


	Exercice : 00
Création du pod	
Fichiers à rendre : <code>.xcodeproj</code> et tous les fichiers nécessaires	
Fonctions Autorisées : Aucune	
Remarques : n/a	

Pour commencer il vous faut créer un pod **Cocoapods**.

Faites un tour sur le site de [Cocoapods](#) et suivez la procédure. Vous trouverez aussi plein de tutoriels sur internet. Votre pod doit etre en **Swift**, contenir un **example**, vous n'avez pas besoin de **tests**. Le nom de votre pod sera votre **login suivi de l'année courante** (ex : **mlemort2016**).

Chapitre V

Exercice 01 : Podspec

	Exercice : 01
Podspec	
Fichiers à rendre : <code>.xcodeproj</code> et tous les fichiers nécessaires	
Fonctions Autorisées : Aucune	
Remarques : n/a	

Maintenant que votre pod est crée il faut que vous vous occupiez de son fichier **podspec**.

Votre fichier **podspec** doit contenir :

- Une **Description**
- Un **Résumé**
- Un **Framework** CoreData


La seule chose dont nous vous dispensons est l'url du projet sur github.



`pod lib lint VOTRE_POD !`

Chapitre VI

Exercice 02 : xcdatamodeld


	Exercice : 02
xcdatamodeld	
Fichiers à rendre : .xcodeproj et tous les fichiers nécessaires	
Fonctions Autorisées : Aucune	
Remarques : n/a	

Il est temps de créer votre modèle de donnée pour **CoreData** dans votre pod. Ajouter un fichier `article.xcdatamodeld` et ajoutez-y :

- Un **Titre**
- Un **Content**
- Une **Langue**
- Une **Image**
- Une **Date de création**
- Une **Date de modification**

Chapitre VII

Exercice 03 : Class Article


	Exercice : 03
Class Article	
Fichiers à rendre : .xcodeproj et tous les fichiers necessaires	
Fonctions Autorisées : Swift Standard Library, CoreData	
Remarques : n/a	

Maintenant créez votre classe **Article** qui étend de **NSManagedObject**. Votre classe doit contenir les attributs suivants :

- Un **Titre** de type **String** ?
- Un **Content** de type **String** ?
- Une **Langue** de type **String** ?
- Une **Image** de type **NSData** ?
- Une **Date de création** de type **NSDate** ?
- Une **Date de modification** de type **NSDate** ?
- Un **Override** de **Description**

Chapitre VIII

Exercice 04 : Class ArticleManager

	Exercice : 04
Class ArticleManager	
Fichiers à rendre : .xcodeproj et tous les fichiers necessaires	
Fonctions Autorisées : Swift Standard Library, CoreData	
Remarques : n/a	

Maintenant que tout cela est fait nous allons pouvoir créer notre classe **ArticleManager**. Cette classe doit contenir les méthodes suivantes :


- **newArticle** qui permet de créer un nouvel article et le renvoie.
- **getAllArticles** qui renvoie tous les articles stockés.
- **getArticles(withLang lang : String)** qui renvoie tous les articles stockés avec la langue donnée.
- **getArticles(containString str : String)** qui renvoie tous les articles stockés contenant la string passée en parametre.
- **removeArticle(article : Article)** qui supprime l'article.
- **save** qui sauvegarde toutes les modifications.



Lors de la création de votre `NSManagedObjectContext` pensez à utiliser `NSBundle(forClass: AnyClass)` pour charger le bon **Bundle** !

Chapitre IX

Exercice 05 : ViewController

	Exercice : 05
ViewController	
Fichiers à rendre : .xcodeproj et tous les fichiers necessaires	
Fonctions Autorisées : Swift Standard Library, UIKit, CoreData	
Remarques : n/a	

Pour finir créez plusieurs articles dans le **ViewDidLoad** de votre **ViewController** et affichez les dans le debug au lancement de l'application.



Lorsque vous lancez plusieurs fois l'application les anciens articles doivent persister.