



## Compte rendu programmation mobile

Auteur: Lucas Blanchard

## Sommaire

<b>Introduction</b>	<b>3</b>
<b>Conception du modèle de données</b>	<b>3</b>
<b>Création du Storyboard et Programmation de l'application</b>	<b>3</b>
Mise en place des différentes vues	3
Différents moyens d'interaction	3
MyViewController	3
SecondViewController	4
ThirdViewController	4
Lien avec MyViewController	4
<b>Conclusion</b>	<b>4</b>

## Introduction

L'objectif de ces TP est de développer une application complète de type Gestionnaire de tâches. Le développement de cette application sera fait en respectant le modèle MVC et se décomposera en plusieurs phases, chacune d'elles permettant d'ajouter de nouvelles fonctionnalités. Le but de ces TP est de se familiariser avec le développement d'applications iOS et d'appliquer ce que nous avons vu en CM et en TD.

## Conception du modèle de données


La première chose à réaliser n'est autre que la classe ToDo qui servira de modèle de données à l'application. La classe ToDo possède donc une variable nom et une variable desc qui sont tous deux de type String.

## Création du Storyboard et Programmation de l'application

### Mise en place des différentes vues

Comme vu en TD un objet TableView a été mis en place sur la vue disponible. Nous y avons ensuite ajouté différents labels et TextView.

Tout d'abord un label a été ajouté afin d'afficher le nom de l'application, c'est-à-dire "My To-Do List". Ensuite des TextField ont été ajoutés afin d'afficher le titre et la description de la tâche.

Nous savons que plus tard dans ce TP nous devrons ajouter des tâches, en supprimer et afficher les tâches terminées. Tout cela à l'aide de boutons, nous ajoutons donc un bouton "+" afin d'ajouter une tâche, un bouton  afin de cocher une tâche terminée et, en ce qui concerne la suppression d'une tâche, lors d'une séance de TP le professeur ayant parlé d'une fonction pour supprimer une tâche en glissant celle-ci sur le côté aucun bouton ne sera donc ajouté.

Nous ajoutons alors les deux autres vues demandées.

Tout d'abord une seconde vue devra afficher les détails d'une tâche, nous y ajoutons donc un label permettant d'afficher le nom de la tâche (le titre) et un TextView permettant d'afficher sur plusieurs lignes la description de cette tâche.

En ce qui concerne la dernière vue demandée, celle-ci doit permettre d'ajouter de nouvelles tâches. Nous ajoutons un bouton "Retour" et un bouton "Enregistrer". Nous ajoutons aussi un TextField pour le nom et un TextView pour la description.

### Différents moyens d'interaction

#### MyViewController

Le professeur ayant parlé d'une fonction permettant la suppression de tâche, une recherche internet nous permet de trouver cette fonction que nous implémentons donc.

Cette fonction permet de retirer une tâche de todo (qui est une instantiation de la classe ToDo). Cela se fait en glissant le doigt sur une tâche.

## SecondViewController

La classe SecondViewController est le ViewController de la deuxième vue, vue permettant d'afficher le détail d'une tâche. Le nom est un label et la description un TextView comme dit au début de ce compte rendu.

Si val est l'instanciation todo alors le nom et la description de la tâche sont ajoutés aux variables nom et desc de val.

## ThirdViewController

Comme dit précédemment, on retrouve sur la troisième vue de l'application un TextField et un TextView pour le nom et la description de la tâche.

## Lien avec MyViewController

Dans le cas où un clic est effectué sur une tâche le todo du SecondViewController est égal au todo ayant été cliqué, de ce fait on retrouvera les informations de la tâche (le nom et la description) dans la seconde vue (vue permettant d'afficher le détail de la tâche), de ce fait si la description était trop longue pour être entièrement affichée sur la première vue elle pourra l'être entièrement sur la deuxième vue.


On met des identifiants sur les événements UnwindSegue lorsque l'on clique sur le bouton Retour et sur le bouton Enregistrer de la troisième vue.

Dans le cas où l'identifiant est égal à "retour" cela signifie que l'on a cliqué sur le bouton retour et de ce fait on ferme la troisième vue pour revenir à la première vue que l'on peut qualifier de vue principale.

Dans le cas où l'identifiant est égal à "enregistrer" cela signifie que l'on a cliqué sur le bouton Enregistrer et de ce fait on récupère dans une variable ajout\_nom le texte de la case correspondant au nom sur la troisième vue ainsi que dans une variable ajout\_desc le texte de la case correspondant à la description sur la troisième vue, c'est-à-dire le nom et la description de la tâche à ajouter.

On crée alors une variable ajout\_todo de type Todo avec comme paramètres le nom ajout\_nom et la description ajout\_desc.

On ajoute ensuite ajout\_todo à l'ensemble des todo de l'application puis on recharge l'affichage du Tableview afin d'afficher cette nouvelle tâche.

Enfin, le bouton  présent sur la première vue permet de "cocher" une tâche terminée.

## Conclusion

Au cours de ce TP nous avons progressé en swift, nous avons pu mettre au point une application de type ToDoList qui possède différentes tâches avec la possibilité de supprimer celles-ci mais aussi d'en ajouter d'autres. Par manque de temps sur des appareils MAC il n'a pas été possible de terminer l'ensemble de ce qui était demandé. De plus, par manque de temps, l'avancement du sujet a été privilégié sur les commentaires présents dans le code.