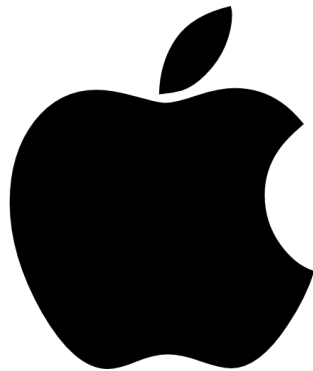


Rapport de projet de programmation mobile, Swift

DEVOUCOUX Maxime, 4A ILC



L'objectif de ce projet est de nous faire réaliser une application ToDoList permettant de lister des tâches, d'en ajouter, d'en supprimer et d'autres fonctionnalités listées par après. Cette application sera développée en langage swift.

1 - Conception du modèle de données.

Nous allons implémenter une classe Todo permettant de matérialiser les tâches que nous allons manipuler dans notre application. La classe Todo est définie dans le fichier Todo.swift et est constituée d'un minimum d'informations essentielles. Parmi celles-ci nous retrouvons deux variables string permettant de stocker le nom et la description de la tâche. De plus, une variable booléenne permet de gérer le statut de la tâche, à savoir faite ou non faite. Nous retrouvons ensuite un constructeur pour initialiser le nom et la description par paramètres et la variable de statut est toujours mise à false par défaut.

En plus de cette classe Todo j'ai décidé de créer une classe ToDoList dans le fichier ToDoList.swift qui est en fait un tableau d'objets Todo car je préfère cette façon de procéder pour manipuler les données. Cette classe est composée d'un tableau et d'un nom de liste.

2 - Création de storyboard et programmation de l'application.

La première étape est de disposer les différents éléments qui vont composer l'interface graphique sur la première vue. J'ai donc choisi d'utiliser un tableView dans lequel j'ai placé un label permettant d'afficher le nom de la tâche ainsi qu'un bouton permettant de passer la tâche de non faite à faite ou inversement. De plus, j'ai placé un bouton en haut de la TableView permettant d'ajouter une tâche. Je crée également une variable myData de type ToDoList qui servira à contenir les objets Todo qui seront affichés dans le TableView. Je peux l'initialiser grâce à la fonction viewDidLoad() qui va initialiser 10 tâches génériques pour l'exemple. Pour cela je crée 10 objets Todo différents que j'ajoute dans la variable myData. Grâce à la fonction tableView() je peux afficher les éléments dans le tableView.

La gestion du comportement de la cellule se fera dans le fichier `TableViewCell.swift`. La fonction principale est la fonction `changementEtat()` qui permet de changer la couleur du bouton entre rouge et vert ainsi que de changer la valeur de la variable booléenne d'état de la tâche.

La prochaine étape est de créer une nouvelle vue qui va afficher la description de la tâche. Je crée donc un nouveau `ViewController` se nommant `DetailViewController`. J'y place 2 labels pour afficher le nom et la description de la tâche. J'y ajoute également un bouton permettant de supprimer la tâche. Dans la fonction `viewDidLoad()` nous allons afficher les informations lorsque la vue sera chargée. J'affiche alors dans les labels le contenu du titre et de la description de la cellule cliquée sur la première vue. J'affiche également le statut de la tâche en fonction de l'état de la variable de statut. Les données sont transmises depuis grâce à la fonction `prepare()` dans le fichier `ViewController` qui détermine la ligne sur laquelle l'utilisateur a cliqué et qui transmet l'objet `Todo` correspondant. Je peux alors utiliser ces informations pour remplir les labels de la vue de détails. Enfin, la suppression de la tâche sur fait par appui sur un bouton qui est liée à un `exit` sur la fonction `supprimer()` dans le fichier `ViewController`. Celle-ci va également récupérer la ligne sélectionnée et va supprimer l'élément dans la classe `ToDoList` à la position associée. En actualisant l'affichage du `TableView` on constate que la tâche est bien supprimée.

La dernière fonction principale sera effectuée dans une autre vue. J'ai donc créé la vue `AjouterViewController` dans laquelle j'ai placé deux labels, un bouton ainsi que deux `textEdit` qui serviront à récupérer les informations saisies par l'utilisateur. La fonction `prepare()` du fichier `AjouterViewController` () permet de récupérer le nom et la description contenus dans les `TextEdit`. Pour cela je crée un nouvel objet `Todo` auquel je donne le nom et la description récupérés depuis les `textEdit`. Le bouton permettant de valider est lié à la fonction `valider()`. Cette fonction a pour effet d'ajouter à l'objet `ToDoList` l'objet créé dans la fonction précédente. Après actualisation de la `TableView` l'objet est ajouté à la liste.