

TD4

La séance commence par un rappel sur les modes de navigation, à l'aide des contrôleurs de contrôleurs : les UINavigationController et les UITabBarController.

Elle se poursuit par la description de la procédure permettant d'exécuter une application sur un appareil physique, plutôt que dans le simulateur. Il faut pour cela :

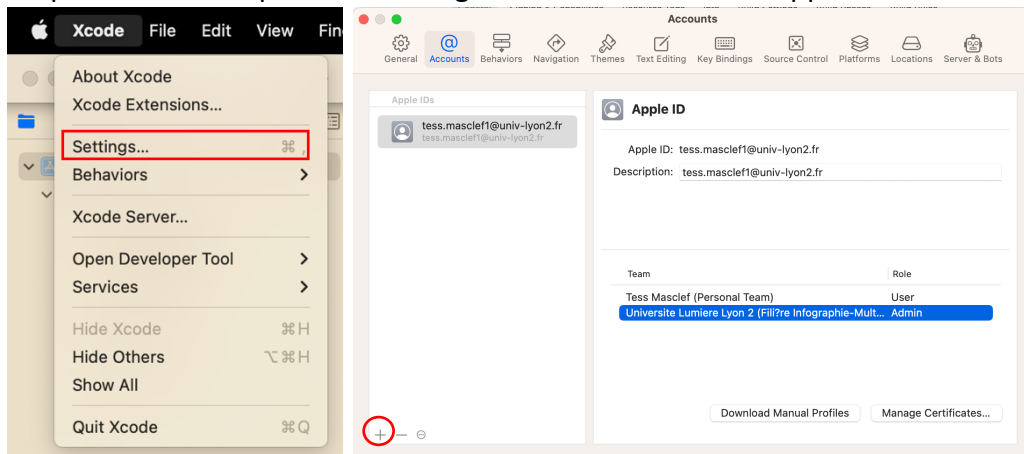
- Signer son application avec un certificat développeur
- Être enrôlé dans une équipe de développement
- Voir son appareil enregistré dans les appareils valides de l'équipe de développement
- Télécharger un profil d'approvisionnement, qui contient toutes les clefs autorisant l'exécution d'une application donnée, compilée par un utilisateur donné sur un appareil reconnu.

Deux exercices permettent ensuite de manipuler les structures de données importantes en Swift : chaînes de caractères et tableaux.

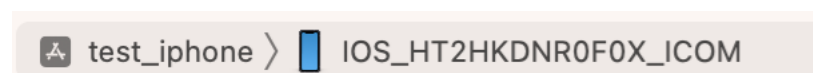
Déploiement sur un device physique

Pour obtenir votre certificat développeur, il faut commencer par accepter l'invitation que vous avez reçue par mail le 29 ou le 30 novembre sur votre adresse Lyon 2. Créez un AppleID associé à cette adresse mail. Puis connectez-vous sous Xcode avec cet AppleID, et choisissez l'équipe de développement "Filière Infographie-Multimédia".

1. Créez un nouveau projet Xcode comme vu dans les TD précédents, que vous nommerez « TP4_exercice ».
2. Vous devez ensuite vous enregistrer sous Xcode avec l'AppleID et choisir une équipe. Cliquez sur le « + » pour vous enregistrer et sélectionnez « AppleID ».



3. Les iPhones de l'ICOM sont des iPhone 12 actuellement installés sous iOS 16.2. Vous devrez donc changer la version d'iOS dans « Minimum Deployments » en 16.2. Vous devrez également choisir l'iPhone plutôt qu'un simulateur, comme cible de la compilation.



Intégration des différents modes de navigation

4. Dans le Storyboard, commencez par ajouter 3 vues (« écrans ») et changez la couleur de leur arrière-plan.
5. Sélectionnez la première vue pour en faire la vue principale du contrôleur de barre d'onglets. Cliquez sur l'icône « Embed In » et sélectionnez « TabBarController ».



6. Changez le titre des onglets (« Accueil ») ainsi que l'icône associée (dans l'inspecteur d'attributs).
7. Associez les deux autres vues au Tab Bar Controller (ctrl + clic souris depuis le Tab Bar Controller vers les deux autres vues, et sélectionnez « Relationship segue → View controllers »). Changez les titres des onglets (« Miroir », « Jours ») et leurs icônes.

Testez votre application (le test se fera donc sur l'iPhone et non plus sur le simulateur)

8. Nous allons maintenant combiner le « TabBarController » et un « UINavigationController ». Ajoutez deux boutons à la vue « Accueil », puis ajoutez deux vues (dont vous changerez les couleurs) et faites une transition entre les boutons et vues (ctrl + clic souris → show).
9. Faites de la vue « Accueil » la racine du « UINavigationController ».

Testez votre application

Exercice sur les chaînes de caractères

10. Sur la vue « Miroir », ajoutez un label « Veuillez entrer une chaîne de caractère et cliquez sur Ok », un champ de texte et un bouton « Ok ». (Attention de laisser assez de place pour l'affichage du clavier).
11. Ajoutez un nouveau fichier « ViewController » comme vu dans le TD précédent que vous associerez avec la vue. (Vous le nommerez « ViewControllerMiroir »)
12. Implémentez les lignes de code dans ce fichier « ViewControllerMiroir » qui permettent de renvoyer le miroir de la chaîne de caractère saisie (par exemple : Bonjour => ruojnoB)
13. (Optionnel) Le code suivant permet de faire disparaître le clavier lorsque vous cliquez n'importe où sur l'écran :

```
override func viewDidLoad() { // Dans la fonction viewDidLoad()
    super.viewDidLoad()

    //Looks for single or multiple taps.
    let tap = UITapGestureRecognizer(target: self, action:
#selector(UIInputViewController.dismissKeyboard))

    //Uncomment the line below if you want the tap not to interfere and cancel other
interactions.
    //tap.cancelsTouchesInView = false

    view.addGestureRecognizer(tap)
}

//Calls this function when the tap is recognized.
@objc func dismissKeyboard() {
    //Causes the view (or one of its embedded text fields) to resign the first responder
status.
    view.endEditing(true)
}
```

Testez votre application

Exercice sur les tableaux

14. Sur la vue « Jours », ajoutez deux boutons (« Endroit » et « Envers ») et un label (prévoyez 7 lignes au minimum).
15. Créez un nouveau fichier « ViewControllerJours » que vous associerez à la vue jour.
16. Puis à partir d'une liste composée des jours de la semaine, affichez les jours de la semaine (un jour par ligne) lorsque le bouton « Endroit » a été cliqué et affichez les jours de la semaine dans l'ordre inverse (du dimanche au lundi) lorsque le bouton « Envers » a été cliqué.

Testez votre application

17. Déposez une archive compressée sur moodle « TP4_exercice.zip »