

TP1 - Première interface graphique

Objectif du TP

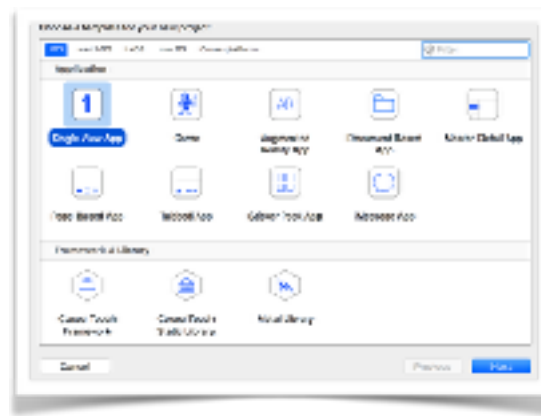
L'objectif du TP consiste à se familiariser avec l'environnement de développement Xcode et à manipuler quelques éléments d'interface graphique.

Partie I : Création de l'application

Vous allez tout d'abord créer un nouveau projet que vous appellerez PremiereInterface



(1)



(2)



(3)

- (1) Lancer Xcode et sélectionner « Create a new Xcode project »
- (2) Sous le menu iOS, choisissez le template « Single View App »
- (3) Saisissez le nom de votre projet (PremiereInterface) dans « Product Name », indiquez votre nom sous « Organisation Name » puis indiquez « com.IUTOrsay » sous le champ « Organization Identifier ». Vérifier que le champ « Language » correspond à « Swift »

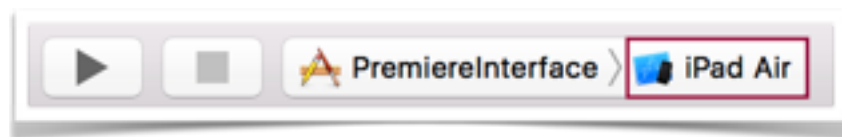


Pour pouvoir tester votre application sur iPad, le Bundle Identifier de votre application doit obligatoirement commencer par « com.IUTOrsay » afin d'être rattaché au compte développeur de l'IUT.

L'écran principal de l'IDE Xcode s'ouvre. Sur la partie gauche de l'écran, vous voyez apparaître la liste des fichiers associés à votre application. Les fichiers `AppDelegate.swift`, `ViewController.swift`, `Main.storyboard`, `Assets.xcassets` et `LaunchScreen.storyboard` et `Info.plist`.

Au centre de l'écran, vous voyez apparaître différentes informations concernant votre application avec notamment le « Bundle Identifier » de votre application ainsi que votre équipe de développement (champ « Team » pour l'instant à None).

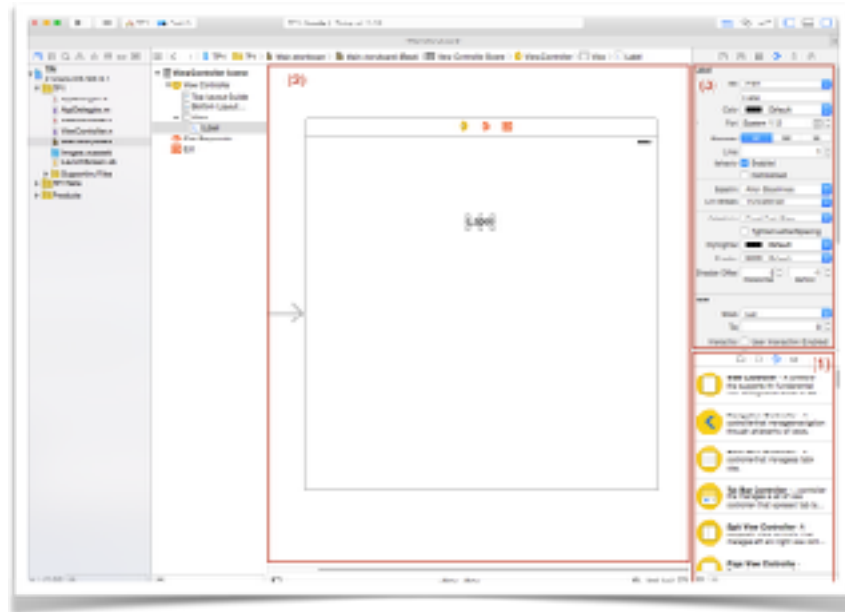
Pour le moment, votre application est vide. Vous pouvez la tester en lançant l'iOS Simulator en cliquant sur la flèche en haut à gauche de l'écran Xcode. Vous pouvez choisir l'appareil utilisé pour la simulation en cliquant sur l'appareil situé à droite du nom de votre application en haut de la fenêtre Xcode. Sélectionnez iPad Air.



Partie II : Création des éléments d'interface graphique

Vous allez maintenant commencer à créer l'interface de votre application. Dans cette application vous allez utiliser des objets de type « Label », « Button », « SegmentedControl », « ActivityIndicator » et « Slider ». La création de ces objets va se faire par le biais de votre storyboard.

Ouvrez le fichier `Main.storyboard`. En bas à droite de la fenêtre, vous trouvez la bibliothèque d'objets (1) qui va vous permettre de sélectionner et glisser-déposer différents types d'objets sur votre canevas (2). Lorsqu'un objet a été ajouté sur le canevas et qu'il est sélectionné, le volet Utilities (3) permet de modifier ses caractéristiques.




Si la bibliothèque d'objets n'est pas ouverte par défaut, elle peut être ouverte via le menu « View > Utilities > Show object Library ».

Dans un premier temps, vous allez manipuler un objet de type « Slider » qui va permettre à l'utilisateur de choisir une valeur comprise entre 0 et 20, valeur qui sera affichée dans un objet de type « Label ».

Question 1 : Ajoutez un objet de type « Slider » ainsi qu'un objet de type « Label » à votre interface graphique. Connectez ces deux objets à votre code en créant les IBOutlet correspondant. Pour cela, ouvrez l'assistant d'édition pour mettre côte à côte les fichiers `Main.storyboard` et `ViewController.swift`. Cliquez sur le label dans le storyboard et glissez votre souris vers le fichier `ViewController.swift` en maintenant la touche « ctrl » et le bouton gauche de la souris enfoncés. Lorsque vous relâchez le bouton de la souris, un menu pop-up s'ouvre. Sélectionnez une « Connection » de type « Outlet » et entrez le nom de la variable que vous voulez utiliser.

Paramétrez les attributs du « Slider » dans le menu « Attributes Inspector » pour que les valeurs minimale et maximale soient comprises entre 0 et 20.

Question 2 : Créez l'IBAction qui va être appelée lorsque l'utilisateur manipulera le slider dans votre interface.

Question 3 : Cliquez sur l'objet de type « Slider » dans le fichier Main.Storyboard et étudier l'inspecteur de connection (onglet ). Que constatez-vous ?

Question 4 : Finalement, modifiez le fichier `ViewController.swift` afin que la valeur obtenue grâce à l'objet « Slider » s'affiche dans votre objet de type « Label ». **Testez votre application.**

Conseil : Rendez vous sur les pages suivantes pour connaître les différents attributs et méthodes associés aux objets de type UILabel et UISlider :

❖ <https://developer.apple.com/documentation/uikit/UILabel>

❖ <https://developer.apple.com/documentation/uikit/UISlider>

Vous allez maintenant manipuler un objet de type « ActivityIndicator » dont l'animation sera commandée par un bouton.

Question 5 : Ajoutez les deux éléments d'interface graphique à votre fichier Main.storyboard. Connectez l'interface graphique à votre code en créant l'IBOutlet et l'IBAction correspondants.

Question 6 : Modifier le fichier `ViewController.swift` afin d'ajouter les lignes de code permettant de démarrer ou d'arrêter l'animation de l'objet « ActivityIndicator » lorsque le bouton est cliqué par l'utilisateur. Faites en sorte que le titre de votre bouton change en fonction de l'animation de l'objet activityIndicator. Si l'objet « ActivityIndicator » est déjà en mouvement, le titre du bouton doit être égal à « Stop ». Au contraire, si l'objet « ActivityIndicator » est arrêté, le titre du bouton sera égal à « Tournez ».

Conseil : Rendez vous sur les pages suivantes pour connaître les différents attributs et méthodes associés aux objets de type UIButton et UIActivityIndicatorView :

❖ <https://developer.apple.com/documentation/uikit/uibutton>

❖ <https://developer.apple.com/documentation/uikit/uiactivityindicatorview>

Question 7 : Vous allez maintenant ajouter un objet de type UISwitch à votre interface. Cet objet devra rendre le bouton précédent « non-cliquable » lorsqu'il est en position OFF et « cliquable » lorsqu'il est en position ON.

Conseil : Rendez vous sur la page suivante pour connaître les différents attributs et méthodes associés aux objets de type UISwitch :

❖ <https://developer.apple.com/documentation/uikit/uiswitch>

Vous allez à présent travailler avec un objet de type « SegmentedControl ». Cet objet, composé de trois segments, doit permettre de changer la couleur de fond de la vue principale (rouge, bleu ou jaune) de votre application qui est un objet de type « UIView ».

Question 8 : Ajoutez un objet de type « SegmentedControl » à votre interface graphique. Connectez le à votre code grâce aux IBOutlet et IBAction correspondants et paramétrez le grâce au menu « Attributes Inspector » afin d'augmenter le nombre de segments qu'il contient et de personnaliser le nom de chacun de ces segments.

Question 9 : Modifiez le fichier ViewController.swift afin d'ajouter les instructions permettant de modifier la couleur d'arrière plan de la vue en fonction du segment sélectionné par l'utilisateur dans l'objet de type « SegmentedControl ».

Vous pouvez définir une nouvelle couleur en utilisant des méthodes pré-définies par Apple en utilisant, par exemple, la commande : `UIColor.blueColor()`

Conseils : Rendez vous sur les pages suivantes pour connaître les différents attributs et méthodes associés aux objets de type UIColor, UISegmentedControl et UIView :

❖ <https://developer.apple.com/documentation/uikit/uicolor>

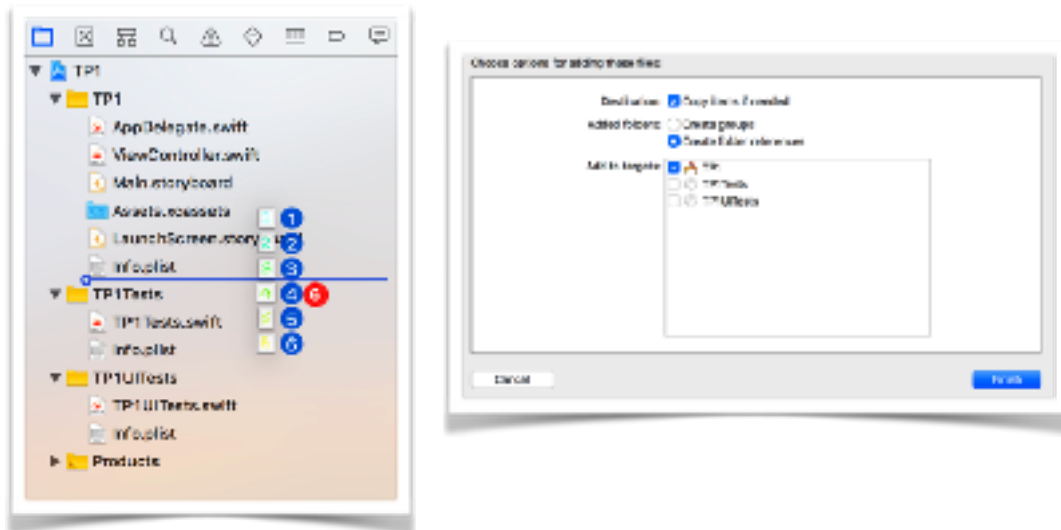
❖ <https://developer.apple.com/documentation/uikit/uisegmentedcontrol>

❖ <https://developer.apple.com/documentation/uikit/uiview>

Finalement, les derniers éléments d'interface graphique que vous allez manipuler sont les objets de type « Stepper » et de type « Image View ». L'objectif consiste à afficher dans un objet de type UIImageView, des images (dont le nom de fichier est stocké dans un tableau), que vous sélectionnerez grâce au UIStepper.

Question 10 : Créez une nouvelle variable d'instance `tableauImage` à votre classe ViewController, de type tableau de chaîne de caractère, contenant le nom des toutes les images à afficher (de « 1.png » à « 6.png »).

Question 11 : Ajoutez les images « 1.png » à « 6.png » à votre projet Xcode. Pour cela, téléchargez les depuis Moodle, et glissez les dans votre projet. Vérifiez que la case « Add to targets » est cochée.



Question 12 : Ajoutez un objet de type « Stepper » à votre interface graphique. Connectez le à votre code grâce aux IBOutlet et IBAction correspondants et paramétrez le grâce au menu « Attributes Inspector » afin de préciser sa valeur minimale et sa valeur maximale.

Question 13 : Ajoutez un objet de type « UIImageView » à votre interface graphique. Connectez le à votre code grâce à un IBOutlet. Initialisez cet objet dans la méthode viewDidLoad afin que l'image affichée soit la première image du tableau.

Conseils : Rendez vous sur les pages suivantes pour connaître les différents attributs et méthodes associés aux objets de type UIImageView et UIImage :

- ❖ <https://developer.apple.com/documentation/uikit/UIImageView>
- ❖ <https://developer.apple.com/documentation/uikit/uiimage>

Question 14 : Modifiez la méthode associée à votre stepper afin que l'image affichée dans votre UIImageView change à chaque fois que le stepper est sélectionné par l'utilisateur. La valeur du stepper correspond à l'indice dans le tableau `tableauImage`.

Conseils : Rendez vous sur les pages suivantes pour connaître les différents attributs et méthodes associés aux objets de type UIStepper :

- ❖ <https://developer.apple.com/documentation/uikit/ui stepper>

Question 15* : Ajoutez d'autres jeux d'images à votre application et permettez à l'utilisateur de choisir le jeu d'images de son choix en utilisant l'élément d'interface graphique qui vous semble le plus adapté.

Partie III : Test de l'application sur l'iPad

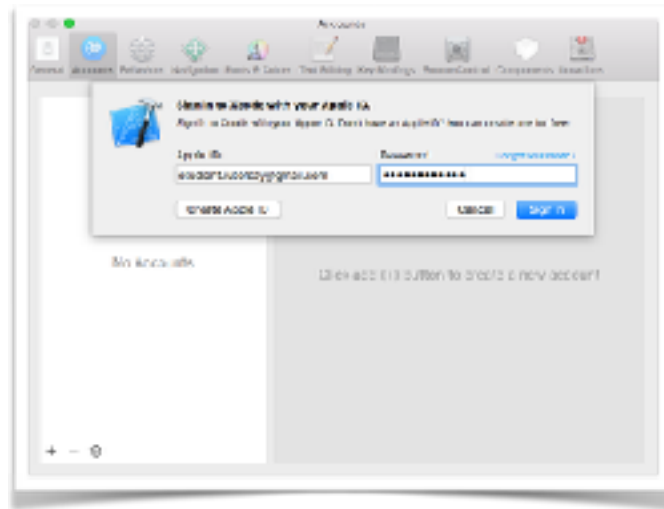
Afin de tester les applications que vous développez, il est nécessaire de posséder un compte Apple Developer et d'adhérer au programme Developer proposé par Apple. Il existe plusieurs mode d'adhésion à ce programme. Dans le cadre de ce module, l'IUT participe au programme iOS Developer University Program qui vous permet de tester vos applications sur iPad en mode Ad Hoc (mode de distribution qui permet de diffuser une application à un ensemble fini d'appareils qui sont explicitement listés).

Connectez votre projet Xcode au compte Apple Developer de l'IUT afin de rejoindre l'équipe de développement. Pour cela, cliquez sur le bouton « Add Account » du menu « Signing ».

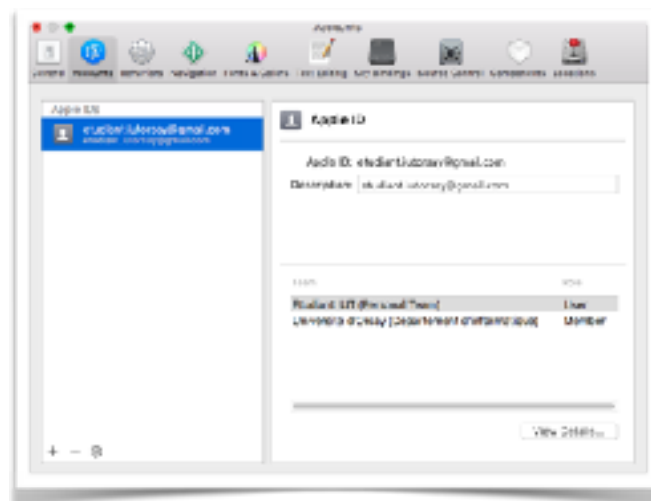


Une fenêtre de dialogue s'ouvre, vous permettant de rentrer les informations de connexion de votre compte Apple Developer. Si vous n'avez pas encore votre compte, suivez les instructions du document « Créer un compte Apple Developer » disponible sur Module.

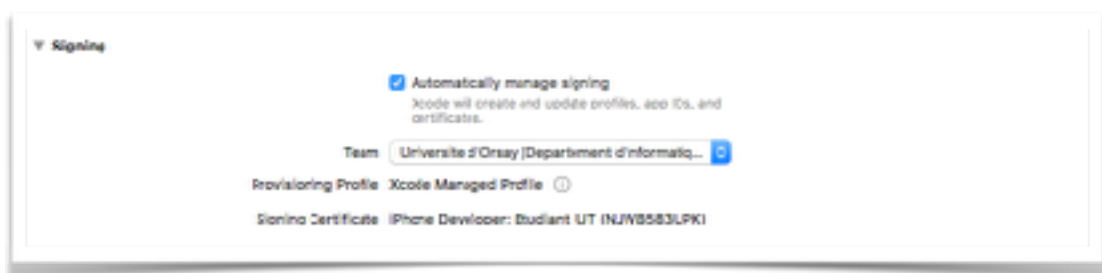
Remplissez les champs **Apple ID** et **Password** puis cliquez sur le bouton « Sign In ».



Vous verrez apparaître les informations concernant votre compte Apple Developer précisant que vous êtes membre de l'équipe de développement « Université d'Orsay (Département Informatique) ». Vous pouvez fermer la fenêtre en cliquant sur la croix rouge en haut à gauche.



Finalement, dans le menu « Signing », mettez à jour le champ **team** en sélectionnant « Université d'Orsay (Département Informatique) ».



En cliquant sur l'appareil situé à droite du nom de votre application en haut de la fenêtre Xcode vous pouvez désormais sélectionner un appareil de test. Sélectionnez « Device > iPad ».

Il est possible que vous voyez apparaître le message d'erreur suivant : **OS version lower than deployment target**.



Il vous faut, dans ce cas, mettre à jour la version du système d'exploitation de votre application dans le menu **Deployment Info**.



Modifiez la valeur du champ **Deployment target** afin qu'elle soit égale à 9.3.

Partie IV : Fin du TP

Question 10 : Supprimez l'application de l'iPad de test (appuie long sur l'icône de l'application, puis appuie sur la croix en haut à gauche de l'icône) et éteignez l'iPad de test.



Déposez votre projet sur Moodle. Pour cela, faites un clic droit sur le dossier parent de votre projet (c'est à dire, celui qui contient le fichier `PremiereInterface.xcodeproj` et le dossier `PremiereInterface`) et sélectionnez « Compresser « PremiereInterface » » dans le menu contextuel. Déposez le fichier `.zip` sur Moodle (rendu TP1).