

Semaine 01

Introduction au développement iOS

Antoine Moevus

Table des matières

Introduction à la conception d'application mobile	1
Méthodes de développement d'applications mobiles	1
Devenir un développeur iOS	2
Le quotidien d'un développeur iOS	2
Les outils et compétences clés	2
Les enjeux de la collaboration	2
Comment évoluer en tant que développeur iOS	3
Environnement de Développement Apple	4
macOs	4
iOS	4
Xcode	4
Accès à distance	4
Exploration de l'interface d'Xcode	6
Installation et configuration d'Xcode	6
Barre d'outils et panneaux de navigation	6
Éditeur de codes et éditeur d'interfaces	6
Inspecteurs et bibliothèques	6
Création d'un nouveau projet iOS	6
Émulateur iOS et débogage	6
Travailler avec le contrôle de version	8
Questions	9
Questions de réflexion	10

Introduction à la conception d'application mobile

Le développement mobile est devenu un incontournable dans l'industrie du logiciel. Des milliards d'utilisateurs utilisent des téléphones intelligents et des tablettes pour effectuer différentes tâches et se divertir. Le marché est estimé à plus de 200 000 000 USD en 2021 et devrait atteindre 500 000 000 USD en 2030^[1].

Il est devenu évident que les applications mobiles sont essentielles pour les entreprises qui cherchent à atteindre et fidéliser leur clientèle, offrant ainsi des opportunités économiques considérables pour les développeurs.

Méthodes de développement d'applications mobiles

Il existe deux principales approches pour développer des applications mobiles: le développement natif et le développement multiplateforme.

- Le développement natif : Il s'agit de créer une application spécifiquement pour une plateforme, comme iOS ou Android, en utilisant les outils et langages de programmation propres à chaque système d'exploitation (par exemple, Swift et Xcode pour iOS, Java et Android Studio pour Android).
- Le développement multiplateforme : Cette méthode consiste à créer une application qui peut fonctionner sur plusieurs plateformes avec un seul code source. Des outils et frameworks tels que React Native, Xamarin, ou Flutter permettent de réaliser ce type de développement, rendant le processus plus rapide et économique pour les entreprises.

Les applications natives offrent généralement de meilleures performances et une intégration plus étroite avec le système d'exploitation, tandis que les applications multiplateformes permettent de réduire le coût et le temps de développement en partageant une grande partie du code entre différentes plateformes.

[1] Straits Research. (s. d.). Mobile App Development Market. Récupéré sur <https://straitresearch.com/report/mobile-app-development-market>

Devenir un développeur iOS

Un développeur iOS est un spécialiste qui conçoit, construit et maintient des applications pour la plateforme iOS d'Apple, notamment pour iPhone et iPad. Le développement d'applications pour iOS implique l'utilisation d'outils et de technologies spécifiques à l'écosystème Apple.

Le quotidien d'un développeur iOS

Les développeurs iOS travaillent généralement sur des projets qui impliquent la conception et la mise en œuvre d'applications mobiles, ainsi que la maintenance et la mise à jour des applications existantes. Leurs tâches quotidiennes comprennent la programmation en Swift ou en Objective-C, l'utilisation d'Xcode pour la création et le débogage d'applications, la conception d'interfaces utilisateur avec Interface Builder, et la collaboration avec d'autres développeurs ou concepteurs.

Les outils et compétences clés

Pour réussir en tant que développeur iOS, il est essentiel de maîtriser les compétences suivantes :

- Connaissance approfondie du langage de programmation Swift (ou Objective-C pour les projets plus anciens)
- Maîtrise d'Xcode, l'environnement de développement intégré (IDE) d'Apple pour les applications iOS
- Expérience avec les bibliothèques et les frameworks iOS tels que UIKit, Core Data, Core Graphics, etc.
- Connaissance des principes de la programmation orientée objet et des modèles de conception couramment utilisés dans les applications iOS, tels que MVC (Model-View-Controller)
- Expérience avec les systèmes de gestion de version, comme Git

Les enjeux de la collaboration

Les développeurs iOS doivent être prêts à s'adapter rapidement aux nouvelles technologies et aux changements dans l'écosystème Apple. Chaque année, Apple met à jour ses systèmes d'exploitation, introduisant de nouvelles fonctionnalités et des améliorations qui peuvent affecter la manière dont les applications sont développées. Les développeurs doivent donc rester informés des dernières tendances et des meilleures pratiques pour s'assurer que leurs applications continuent de fonctionner correctement et offrent une expérience utilisateur optimale. Les développeurs iOS doivent tenir compte de la concurrence croissante sur l'App Store, en se concentrant sur la qualité et l'innovation pour se démarquer. Ils doivent également s'adapter aux attentes changeantes des utilisateurs et aux normes de l'industrie, en améliorant constamment leurs compétences et en apprenant de nouvelles technologies pour rester pertinents.

De plus, les développeurs iOS travaillent généralement en étroite collaboration avec les concepteurs d'interface utilisateur (UI) et d'expérience utilisateur (UX) pour créer des applications attrayantes et intuitives. Ainsi, il est essentiel de comprendre les principes de conception, de travailler en équipe et de communiquer efficacement avec les concepteurs pour assurer une expérience utilisateur

réussie et des livrables de qualité.

Autre part de l'équation: les gestionnaires de projets qui supervisent et coordonnent les différentes étapes du développement d'une application. Ainsi en tant que programmeur, il est important de comprendre les besoins du projet, de respecter les délais et les budgets, et de communiquer clairement et régulièrement avec les gestionnaires pour garantir un déroulement fluide du projet.

En renforçant vos compétences en matière de collaboration et en apprenant à travailler efficacement avec les concepteurs UI/UX et les gestionnaires de projets, vous pourrez améliorer encore plus votre carrière en tant que développeur iOS et contribuer à la réussite des projets auxquels vous participez.

Comment évoluer en tant que développeur iOS

Pour progresser et réussir en tant que développeur iOS, il est important d'adopter une approche proactive en matière d'apprentissage et de développement professionnel. Voici quelques conseils pour évoluer en tant que développeur iOS :

1. Améliorez vos compétences en programmation : approfondissez votre maîtrise de Swift et Objective-C et apprenez d'autres langages de programmation pertinents, tels que Python ou JavaScript, pour étendre votre polyvalence en tant que développeur.
2. Restez informé des tendances et des mises à jour d'iOS : suivez les blogs, les forums, et les conférences consacrés au développement iOS pour vous tenir au courant des nouveautés et des meilleures pratiques.
3. Étudiez les applications populaires : analysez les applications à succès sur l'App Store pour comprendre ce qui fonctionne bien et ce qui peut être amélioré.
4. Travaillez sur des projets personnels : développez vos propres applications pour expérimenter de nouvelles fonctionnalités, résoudre des problèmes complexes et vous familiariser avec différentes approches de développement.
5. Collaborez avec d'autres développeurs : participez à des projets open source, rejoignez des groupes de développeurs locaux et échangez avec d'autres professionnels pour apprendre de nouvelles méthodes et élargir votre réseau.
6. Poursuivez votre éducation : participez à des formations, des ateliers ou des cours en ligne pour développer vos compétences et obtenir des certifications pertinentes.

En investissant dans votre apprentissage et votre croissance personnelle, vous pouvez évoluer en tant que programmeur iOS et continuer à être un acteur clé dans le domaine du développement mobile.

Environnement de Développement Apple

macOS

macOS est le système d'exploitation développé par Apple pour les ordinateurs Mac. Il fournit un environnement de développement riche en fonctionnalités et en outils pour créer des applications pour les plateformes Apple, notamment iOS, iPadOS, watchOS et tvOS.

macOS, anciennement connu sous le nom de Mac OS X puis OS X, a été introduit en 2001 comme le successeur de Mac OS 9. Il s'agit d'un système d'exploitation basé sur UNIX et a été construit sur la base du système NeXTSTEP, qui a été développé par NeXT, la société fondée par Steve Jobs après son départ d'Apple en 1985. Depuis lors, macOS a connu plusieurs mises à jour majeures, avec des noms de code basés sur des grands félins, puis sur des lieux emblématiques de Californie.

iOS

iOS est le système d'exploitation mobile développé par Apple pour les iPhone et iPod touch. Il propose une interface utilisateur intuitive et un ensemble complet de frameworks et de bibliothèques pour faciliter le développement d'applications mobiles.

iOS, initialement appelé iPhone OS, a été introduit en 2007 avec la sortie du premier iPhone. Basé sur macOS, il a été conçu pour fournir une expérience utilisateur optimisée pour les appareils mobiles avec écran tactile. Depuis lors, iOS a évolué pour prendre en charge d'autres appareils tels que l'iPod touch, l'iPad et l'Apple TV (tvOS étant un dérivé d'iOS). Les versions successives d'iOS ont apporté de nouvelles fonctionnalités et améliorations, ainsi que des mises à jour des frameworks de développement pour permettre aux développeurs de créer des applications toujours plus puissantes et innovantes.

Xcode

Xcode est l'environnement de développement intégré (EDI) officiel d'Apple pour la création d'applications pour les plateformes Apple, y compris macOS, iOS, iPadOS, watchOS et tvOS. Xcode offre un éditeur de code source, un éditeur d'interface graphique, un gestionnaire de projet, un émulateur pour tester les applications, ainsi que des outils de débogage et d'optimisation des performances.

Accès à distance

Si vous n'avez pas de Mac à disposition pour le développement iOS, il est possible d'accéder à un Mac à distance via des services en ligne qui fournissent des machines virtuelles macOS. Ces services vous permettent de louer un Mac sur le cloud et d'accéder à l'ordinateur via une connexion à distance. Voici trois options populaires :

- MacinCloud (<https://www.macincloud.com/>): MacinCloud propose des plans mensuels pour accéder à des machines virtuelles macOS. Les plans incluent différentes configurations matérielles et logicielles pour répondre à vos besoins.

- VirtualMacOSX (<https://virtualmacosx.com/>): VirtualMacOSX offre également des plans mensuels pour l'accès à distance à des ordinateurs Mac. Ils proposent des options d'accès via VNC ou RDP et disposent de serveurs dans plusieurs régions pour minimiser la latence.
- HostMyApple (<https://www.hostmyapple.com/>): HostMyApple est un autre fournisseur d'accès à distance aux machines macOS. Ils proposent des plans mensuels flexibles avec différentes configurations matérielles et logicielles, ainsi que des options de serveurs dans plusieurs pays.

Ces services sont particulièrement utiles pour les développeurs qui souhaitent travailler sur des projets iOS sans investir dans du matériel Apple coûteux. Avec ces services, vous pouvez accéder à un Mac à distance et utiliser Xcode et d'autres outils de développement macOS pour créer des applications iOS.

Exploration de l'interface d'Xcode

Installation et configuration d'Xcode

1. Rendez-vous sur le Mac App Store et recherchez "Xcode".
2. Téléchargez et installez Xcode.
3. Lancez Xcode et acceptez les conditions d'utilisation.
4. Laissez Xcode installer les outils de développement supplémentaires si nécessaire.

Barre d'outils et panneaux de navigation

1. Familiarisez-vous avec la barre d'outils en haut de l'écran.
2. Explorez le panneau de navigation à gauche, qui contient la hiérarchie des fichiers, des symboles, des problèmes, etc.
3. Découvrez comment ouvrir et fermer les différents panneaux à l'aide des boutons de la barre d'outils.

Éditeur de codes et éditeur d'interfaces

1. Ouvrez un fichier source Swift (.swift) pour explorer l'éditeur de code.
2. Ouvrez un fichier de storyboard (.storyboard) pour explorer l'éditeur d'interface.

Inspecteurs et bibliothèques

1. Apprenez à utiliser les inspecteurs de l'éditeur de code pour examiner les propriétés et les attributs des éléments.
2. Apprenez à utiliser les bibliothèques de l'éditeur d'interface pour ajouter des composants et des ressources à votre projet.

Création d'un nouveau projet iOS

1. Cliquez sur "File" > "New" > "Project" dans la barre de menu.
2. Choisissez un modèle de projet (par exemple, "App").
3. Nommez votre projet et choisissez un emplacement pour le sauvegarder.
4. Sélectionnez les options de configuration du projet, comme le langage de programmation et l'interface utilisateur.

Émulateur iOS et débogage

1. Sélectionnez un appareil dans la liste déroulante en haut à gauche de la barre d'outils.
2. Cliquez sur le bouton "Run" (flèche) pour lancer votre application sur l'émulateur.

3. Utilisez les outils de débogage pour résoudre les problèmes de code et de performance.

Travailler avec le contrôle de version

0. (si ce n'est pas fait) Créez un compte sur GitHub.com.
1. Créez un nouveau projet dans Xcode et assurez-vous d'activer le repo Git local.
2. Allez dans la barre de menu. Choisissez Xcode > Settings > Accounts.
3. Cliquez sur le "+" en bas et ajoutez un compte GitHub.
4. Entrez votre nom d'utilisateur et votre token Github.
 - Si vous n'avez pas de token Github, suivez simplement les instructions sur GitHub pour en créer un. Assurez-vous d'avoir les droits pour **commit**, **push**, et créer des repos.
5. Confirmez en cliquant sur "Sign In".
6. Dans le menu de gauche, à côté de l'icône "Project Navigator", cliquez sur "Show source control navigator". Ensuite, cliquez sur l'onglet "Repositories".
7. Faites un clic droit sur l'élément "Remote" et choisissez parmi "New [*nomDeVotreProjet*] Remote" et "Add existing remote".
8. Vous pouvez maintenant **push** et **pull** sur votre dépôt GitHub.

Questions

1. Quelle est l'estimation du marché des applications mobiles en 2030 ?
 - a. 100 000 000 USD
 - b. 300 000 000 USD
 - c. 500 000 000 USD
 - d. 700 000 000 USD
2. Quelle sont les différences principales entre le développement natif et le développement multiplateforme ?
 - a. Le langage de programmation utilisé
 - b. Le temps de développement
 - c. Le type d'applications créées
 - d. Le coût de développement
3. Quel est l'environnement de développement intégré (IDE) d'Apple pour les applications iOS ?
 - a. Android Studio
 - b. Visual Studio
 - c. Xcode
 - d. IntelliJ IDEA
4. Quelle est la méthode recommandée pour accéder à un Mac à distance pour le développement iOS ?
 - a. Utiliser un service de machines virtuelles macOS
 - b. Installer macOS sur un ordinateur non-Apple
 - c. Utiliser un émulateur macOS sur un ordinateur Windows
 - d. Se connecter à un Mac via FTP
5. Quel est le langage de programmation principal utilisé pour le développement d'applications iOS ?
 - a. Java
 - b. Python
 - c. Swift
 - d. C#
6. Quel système de contrôle de version est le plus couramment utilisé avec Xcode ?
 - a. SVN
 - b. Mercurial
 - c. Git
 - d. Perforce
7. Quel est le but principal d'un token (jeton) dans GitHub?

- a. Améliorer la performance des requêtes API
- b. Offrir une alternative plus sécurisée aux mots de passe pour l'authentification
- c. Faciliter le partage de code entre les membres de l'équipe

Questions de réflexion

- 1) Discutez des avantages et des inconvénients du développement natif par rapport au développement multiplateforme. Pourquoi une entreprise choisirait-elle l'une ou l'autre méthode ?
- 2) Décrivez les étapes à suivre pour installer et configurer Xcode sur un Mac. Quelles sont les principales fonctionnalités offertes par Xcode pour les développeurs iOS ?
- 3) Expliquez l'importance de la collaboration entre les développeurs iOS, les concepteurs d'interface utilisateur (UI/UX), ainsi que les gestionnaires de projets. Comment ces collaborations contribuent-elles au succès des projets de développement d'applications iOS ?