

Programmation **iOS 5** **pour iPhone et iPad**

De la conception à la publication

Jean-Marc Lacoste
Thomas Sarlandie



EYROLLES

Programmation iOS 5

La réussite d'une application iOS repose sur sa conception et sa réalisation : elle exige un savoir-faire en ergonomie mobile et la maîtrise de l'ensemble des contraintes spécifiques à la plate-forme.

Cet ouvrage pour iOS 5 aborde le développement d'applications iPhone et iPad dans tous leurs aspects, depuis l'inscription chez Apple et l'achat de licence du SDK à la distribution sur l'AppStore... sans oublier les bonnes pratiques de conception et les design patterns d'interface.

La référence pour le développeur iPhone et iPad professionnel : de la conception à la publication sur l'App Store

De la conception de l'application – encadrée par de strictes règles d'ergonomie – jusqu'à son déploiement, cet ouvrage détaille les bonnes pratiques garantissant la qualité de vos développements iPhone : gestion de projet et architecture MVC, ergonomie mobile et design patterns d'interface. Les fondamentaux du développement iPhone sont détaillés, de l'Objective-C et sa gestion spécifique de la mémoire aux contrôleurs de vue, en passant par la mise en place des vues et des TableView.

Les nouveautés d'iOS 5 telles que les storyboards et l'ARC (Automatic Reference Counting) dans Objective-C sont traitées.

Écrit par le directeur technique de l'une des premières agences spécialisées dans le développement sur plate-forme mobile et iPhone et par un formateur iOS, l'ouvrage traite en profondeur d'aspects avancés tels que l'accès aux services web (JSON, XML), la gestion de flux audio et vidéo, la persistance avec le framework CoreData et l'utilisation du service de notifications Apple. Enfin, il fournit de précieux conseils pour publier sur l'App Store et y gagner en notoriété.

Au sommaire

L'environnement de développement • Développer pour iPhone/iPad • L'essentiel d'Objective-C • Premiers pas avec le SDK iOS • **Conception et ergonomie** • Méthode de développement d'un projet iOS • Principes ergonomiques et designs patterns d'interface • Conception de l'interface graphique • **Développement de l'interface** • Contrôler les écrans de l'application • Assembler les écrans de l'application • Développer et animer les vues • UIPickerView, delegate et data source • Liste d'éléments • TableViews • **Gestion du multi-tâche** • Lire et enregistrer les données • Communiquer avec l'extérieur • Utilisation du MapKit • Persistance d'objets avec Core Data • Manipuler les données multimédia • APNS • Publication • Bandeaux publicitaires iAd • Les inApp Purchase • Publier sur l'AppStore.

À qui s'adresse cet ouvrage ?

- Aux professionnels de la conception web et mobile qui souhaitent être présents sur le marché des services portés sur iPhone et iPad ;
- À tous les particuliers et fans d'iPhone et iPad qui souhaitent concevoir, publier ou vendre une application sur l'App Store.



J.-M. Lacoste

Jean-Marc Lacoste s'est orienté vers les systèmes Unix/Linux et le C après avoir débuté dans les langages d'intelligence artificielle (Lisp, Scheme, Prolog, Ada, etc.). Ce consultant depuis 20 ans chez Ambre Systems partage son temps entre la formation en milieu professionnel et les interventions auprès des constructeurs (Digital, HP, Sun, IBM), éditeurs et leurs partenaires. À l'arrivée de la plate-forme iOS, il a repris le développement sous l'angle de la formation : la prise en main du langage Objective-C et de l'ensemble de l'environnement de développement sous Xcode pour réaliser rapidement des applications cohérentes et efficaces.

T. Sarlandie

Thomas Sarlandie est co-fondateur de Backelite, l'agence mobile leader en ergonomie et développement d'applications mobiles. Il a été l'un des pionniers du développement iPhone en France et a souhaité partager sa passion et cette expérience à travers ce livre.

www.editions-eyrolles.com
Groupe Eyrolles | Diffusion Geodif

Code éditeur : G85424
ISBN : 978-2-212-85424-4

EYROLLES

Programmation **iOS 5** **pour iPhone et iPad**

De la conception à la publication

CHEZ LE MÊME ÉDITEUR



DANS LA COLLECTION DESIGN WEB



DANS LA COLLECTION A BOOK APART



Programmation **iOS 5** **pour iPhone et iPad**

De la conception à la publication

Jean-Marc Lacoste
Thomas Sarlandie

EYROLLES



ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© Groupe Eyrolles, 2012, ISBN : 978-2-212-85424-4

Avant-propos

Après le succès sans précédent de l'iPhone en 2007, Apple a conduit à une mutation de l'utilisation des outils informatique par l'introduction de l'iPad en 2010, révolutionnant le paysage informatique et modifiant la façon d'utiliser les différents médias disponibles sur Internet.

En permettant aux développeurs de réaliser leurs propres applications sur les appareils et en gérant le circuit de ventes, Apple a réussi en quelques années à devenir l'une des sociétés les plus riches de la planète.

Bonne lecture !

Table des matières

PREMIÈRE PARTIE

Découverte de l'environnement de développement .. 1

CHAPITRE 1

Développer pour iPhone et iPad 3

Équipement matériel requis 3

Un Mac Intel pour développer 3

Un iPhone, un iPod touch ou un iPad pour tester l'application 4

Compétences techniques utiles au développeur iOS 5

La programmation orientée objet, au cœur du SDK iOS 5

L'Objective-C : un langage comme un autre 5

Programmation multithread 6

Développement d'un « client lourd » 6

Concevoir des applications universelles 7

L'adhésion au programme développeur d'Apple 7

Développeur iOS enregistré : un accès bon marché à l'environnement
de développement et à la documentation 7

Le programme Developer pour tester et publier vos applications 7

Les deux modes d'adhésion au programme iOS Developer 8

Le processus d'adhésion au programme iOS Developer 8

Le programme Entreprise pour des applications internes 9

Le programme universitaire pour l'enseignement 9

Les sites web Apple pour le développeur iPhone 9

Le centre de développement iOS 10

Le portail du programme iOS 10

iTunes Connect, pour la publication des applications 11

Présentation du SDK iOS 12

La documentation Apple, une aide à ne pas négliger 13

Les guides pour le développeur 13

Le guide consacré aux règles à respecter en matière d'ergonomie 13

La description des API et de la bibliothèque graphique 14

<i>La référence Objective-C</i>	14
Les exemples de code : des projets Apple comme modèles	14
La documentation de référence exhaustive	14
Prérequis pour la distribution d'une application	14
Préparation à la diffusion d'une application en test (mode Ad Hoc)	15
<i>Générer un certificat de développeur</i>	16
<i>Faire signer par Apple la demande de certificat</i>	17
<i>Créer un identifiant d'application</i>	18
<i>Définir les appareils autorisés</i>	19
<i>Générer le profil de provisionnement</i>	20
<i>Installer le profil sur des appareils</i>	21
<i>Installer l'application sur des appareils</i>	22
Mode de distribution via l'App Store pour une diffusion large	23
<i>Publication sur l'App Store</i>	23
Conclusion	24

CHAPITRE 2

L'essentiel d'Objective-C 25

Les origines	25
Principes fondamentaux	26
Tout est objet	26
Envoi de messages	26
La syntaxe Objective-C	27
Envoyer des messages à un objet	27
Le type id et la valeur nil	28
Déclaration d'une classe	29
<i>Les variables d'instance</i>	30
<i>Les différents types de méthodes</i>	30
<i>Héritage</i>	31
<i>Faire référence à l'objet courant et à son père</i>	31
Initialisation d'une instance d'objet	31
Les protocoles	32
<i>Déclaration d'un protocole</i>	32
<i>Implémenter un protocole</i>	33
Les propriétés	33
<i>Déclarer une propriété</i>	33
<i>Implémenter les accesseurs</i>	34
<i>Attributs des propriétés</i>	34
La notation point	35
Les sélecteurs : des pointeurs sur fonction	35
Synchronisation de threads	36

La bibliothèque standard : le framework Foundation	36
Chaînes de caractères	37
Listes	38
<i>Parcourir rapidement une liste</i>	38
Dictionnaire	39
Le mécanisme de comptage de références	39
Vie et mort des objets	40
<i>Création d'un objet</i>	40
<i>Libération de la mémoire d'un objet</i>	40
<i>Déclaration de propriétés avec ARC</i>	40
Conclusion	41
 CHAPITRE 3	
Premiers pas avec le SDK iOS	43
À la découverte de Xcode	44
Les modèles de projet	44
Découverte d'un projet iPhone vide	45
<i>Organisation sous Xcode</i>	46
<i>Fichiers générés lors de la création du projet</i>	48
Comprendre le code généré par Xcode	48
Rôle du fichier Info.plist	49
<i>Édition du fichier Info.plist</i>	49
<i>Nom de l'application et choix de l'icône</i>	50
<i>Personnalisation de l'écran de démarrage</i>	51
Rôle du délégué de l'application	52
<i>applicationDidFinishLaunching: – Votre application a fini de se lancer</i>	53
<i>applicationWillTerminate: – Votre application va se terminer</i>	53
<i>applicationDidReceiveMemoryWarning: – Votre application doit libérer de la mémoire</i>	53
<i>applicationWillResignActive: – Votre application passe en arrière-plan</i>	54
Construction de l'interface avec un storyboard	55
Déclaration du lien entre l'application et son délégué	58
Hommage à MM. Kernighan et Ritchie	58
Utilisation de la console	58
Création d'une nouvelle vue avec les storyboards	59
Ajouter une nouvelle vue au projet	59
Création d'une nouvelle vue en code	63
<i>Créer une vue</i>	66
<i>Créer un label</i>	67
<i>Créer un bouton</i>	67
<i>Assembler la vue</i>	68

Lancer l'application sur votre appareil	68
Définir les paramètres de signature de l'application	68
Compiler et lancer l'application sur iPhone	69
Manipulations des schémas de compilation (schemes)	69
Créer un nouveau schéma de compilation	70
Créer une macro utilisable pour le débogage	71
Tester notre nouveau schéma de compilation	72
Et maintenant ?	72

DEUXIÈME PARTIE

Conception et ergonomie 73

CHAPITRE 4

Méthode de développement d'un projet iOS..... 75

Qu'est-ce qu'un projet d'application réussi ?	75
Satisfaire les utilisateurs	75
Maîtriser le projet	76
Les étapes du projet	76
Identifier les fonctionnalités clés	76
Définition des principes ergonomiques	77
Storyboarding et spécifications	78
Intervention artistique	80
Développement de l'interface	80
Développement de l'application	81
Tests et optimisation	82
Publication	82
Conclusion	83

CHAPITRE 5

Principes ergonomiques et design patterns d'interface 85

L'ergonomie dans l'univers de iOS	86
Une interface différente basée sur des métaphores	86
Les design patterns d'interface	87
Deux bonnes raisons pour s'aligner sur l'ergonomie iOS	87
<i>Faciliter la prise en main de l'application</i>	87
<i>Accélérer les développements</i>	88
Développer des applications originales	88
Des applications pour consulter et manipuler des données	88
Les métaphores de l'interface iOS	89
Les listes d'éléments ou TableViews	89

<i>Comportements communs aux listes</i>	89
<i>Utilisation de sections dans une liste</i>	90
<i>Ajout d'un index sur une liste</i>	90
<i>Accessoires des éléments et indicateurs de détails</i>	90
<i>Le mode édition</i>	92
Les listes groupées	92
<i>Édition d'une liste groupée</i>	93
Les design patterns d'interface iOS	93
Navigation dans une application utilitaire	93
Utilisation d'une barre d'outils	93
Navigation dans des listes hiérarchiques	94
<i>La barre de navigation</i>	95
<i>La vue de contenu</i>	96
<i>Les animations</i>	96
<i>Principes de navigation à respecter</i>	96
Navigation par onglet	97
<i>Utilisation en combinaison avec d'autres design patterns d'interfaces</i>	97
<i>Personnalisation de la barre d'onglets</i>	98
Ergonomie iPad avec le SplitViewController	99
Ergonomie de style livre	100
Passons à la conception	102

CHAPITRE 6

Conception de l'interface graphique 103

Utilisation d'une application mobile	103
Temps et fréquence d'utilisation	103
Concentration et attention disponible	104
Méthode pour concevoir l'interface d'une application	105
Identifier les fonctionnalités	105
Trier les fonctionnalités par ordre d'importance	106
<i>Les trois groupes de fonctionnalités</i>	106
<i>De l'importance des trois groupes</i>	106
Concevoir l'interface pour ses fonctionnalités	107
<i>Les fonctionnalités du premier groupe doivent être accessibles en un temps minimum</i>	107
<i>Mise en avant des fonctionnalités du deuxième groupe</i>	108
<i>Fonctionnalités du troisième groupe</i>	108
Quelques outils pour concevoir l'interface	109
Un accès unique à chaque fonctionnalité	109
Éviter les menus et autres listes de fonctionnalités	110
Distinguer le premier lancement des suivants	110

Adapter l'interface aux habitudes de l'utilisateur	110
<i>Mémoriser le dernier écran utilisé</i>	110
<i>Proposer à l'utilisateur de personnaliser la barre d'onglets</i>	111
Paramètres de l'application	111
Conclusion	111

TROISIÈME PARTIE

Le développement de l'interface 113

CHAPITRE 7

Contrôler les écrans de l'application 115

Le modèle MVC dans iOS	116
Le modèle pour charger et stocker en mémoire les données de l'application ...	116
La vue pour représenter graphiquement le modèle et fournir l'interface graphique	117
Le contrôleur pour lier le modèle et la vue	117
Le contrôleur de vue standard d'iOS	117
Cycle de vie d'un contrôleur de vue	118
<i>Contrôleur initialisé sans vue</i>	118
<i>Vue chargée et non affichée</i>	119
<i>Vue chargée et affichée</i>	119
<i>Avertissement de mémoire</i>	120
Utilisation des contrôleurs de vue	120
Création d'un nouveau contrôleur de vue	120
Instanciation d'un contrôleur de vue	121
<i>Créer un contrôleur de vue sans storyboard</i>	121
<i>Créer un contrôleur de vue dans un storyboard</i>	121
Établir des transitions	122
Réagir au chargement et au déchargement de la vue	124
<i>Utilisation de la méthode viewDidLoad</i>	124
<i>Comment savoir si la vue est chargée ?</i>	125
Réagir lorsque la vue est affichée puis masquée	125
<i>Affichage de la vue</i>	126
<i>Masquage de la vue</i>	126
Gérer les événements	126
<i>Créer une méthode pour traiter l'événement</i>	126
<i>Lier un événement à une action</i>	127
Gérer les rotations d'écran	128
<i>Événements associés aux rotations d'écran</i>	129
Conclusion	129

CHAPITRE 8

Assembler les écrans de l'application 131

Généralités sur les contrôleurs-conteneurs	131
Le contrôleur de navigation	132
Création d'un contrôleur de navigation	133
Spécifier le contenu de la barre de navigation	133
<i>Titre du contrôleur</i>	134
<i>Boutons supplémentaires</i>	135
<i>Définir la façon dont est représenté le contrôleur quand il n'est plus affiché</i> ..	135
<i>Masquer la barre d'outils ou la barre d'onglets</i>	136
Pousser des écrans dans le contrôleur de navigation	136
Personnaliser la barre de navigation	137
Exemple complet de navigation avec un storyboard	137
Exemple complet de navigation entièrement codé	140
Contrôleur d'onglets	144
Création d'un contrôleur d'onglets	145
Personnalisation du titre et de l'icône des onglets	145
Réagir aux événements de la barre d'onglets	145
<i>Suivre les changements de sélection</i>	146
<i>Réagir à la personnalisation de la barre</i>	146
Limiter la personnalisation de la barre d'onglets	147
Exemple de projet avec contrôleurs d'onglets et storyboard	147
Combiner les contrôleurs d'onglets avec des contrôleurs de navigation	149
Construction d'une combinaison onglets et navigation avec un storyboard ...	150
Affichage d'un contrôleur en mode modal	150
Pousser une nouvelle vue modale	151
Faire disparaître une vue modale	151
Définir le mode de transition	151
Construction d'une transition modale avec les storyboards	152
Construction d'un SplitViewController avec les storyboards	152
Conclusion	156

CHAPITRE 9

Développer et animer les vues..... 157

Comprendre les vues	157
Coordonnées des vues	157
<i>Centre et limites d'une vue</i>	158
<i>Frame d'une vue</i>	159
Hiérarchie des vues	160
Positionnement des sous-vues et redimensionnement automatique des vues ..	160

Les vues élémentaires de UIKit	161
Les labels pour afficher du texte	161
Les vues d'images	162
Les boutons pour déclencher des actions	163
Les zones de texte	164
Les boutons segmentés	165
Les commutateurs	166
Les scrollviews	167
Alertes et feuilles d'action	168
Utilisation d'un compteur	171
Affichage de contenus web dans l'application	172
Animation des vues	173
Internationalisation des vues	175
Localisation des chaînes de caractères	176
Localisation du storyboard	178
Conclusion	178

CHAPITRE 10

Listes déroulantes : les UIPickerView 179

Le design pattern délégation de contrôle	179
Les sources de données	181
Utilisation d'une liste UIPickerView	181
Création d'un simple UIPickerView	182
Connexion	183
Implémentation des méthodes	183
Test	185
Méthodes delegate et data source pour plusieurs objets	185
<i>La méthode pour le nombre de composants.</i>	186
<i>La méthode pour le nombre d'éléments par composant.</i>	186
<i>La méthode pour remplir les éléments pour chaque ligne</i>	187
<i>La méthode optionnelle pour la largeur des composants.</i>	188
<i>Méthode appelée lors de la sélection d'un élément.</i>	188
Conclusion	189

CHAPITRE 11

Listes d'éléments..... 191

Les deux types de listes	191
<i>Les listes simples</i>	191
<i>Les listes groupées</i>	191
Créer une TableView	192
Fournir des données à une TableView	193

Décider du nombre de sections	194
Indiquer le nombre de lignes	194
Afficher des données	194
Définir les en-têtes et pieds de section	196
Réagir aux actions sur la liste	196
Sélection d'un élément	196
<i>Sélection à l'aide des storyboards</i>	196
<i>Édition dans une TableView</i>	198
Techniques pour afficher des cellules personnalisées	198
Composition de la cellule	199
Dessiner manuellement le contenu de la cellule	200
Un exemple complet	200
Création du projet	201
Création des ViewController	202
Création du storyboard et connexion avec les ViewController	202
Le fichier RootViewController.h	202
Connecter les propriétés au storyboard	204
Le fichier RootViewController.m	205
<i>Les propriétés</i>	205
<i>Méthodes associées aux boutons</i>	205
<i>Déplacer les cellules</i>	207
<i>Insérer ou supprimer une cellule</i>	207
<i>Les méthodes de base d'un TableView</i>	209
<i>Autres méthodes du RootViewController.m</i>	210
Le fichier DetailViewController.h	211
<i>Le fichier DetailViewController.m</i>	211
Conclusion	212

CHAPITRE 12

Contact et gestuelle 213

Détection de doigt(s) au contact de l'écran	213
La classe UITouch	214
Méthodes déclenchées par les événements liés aux mouvements des doigts ..	214
<i>Cycle de vie d'un contact à l'écran</i>	215
Prise en charge des gestes courants (discrets et continus)	216
<i>Taper à l'écran</i>	217
<i>Faire glisser le doigt (swipe)</i>	219
<i>Faire pivoter un objet</i>	221
<i>Déplacer (pan)</i>	222
<i>Pression longue</i>	224
<i>Pincer/zoomer (pinch)</i>	226
Conclusion	229

QUATRIÈME PARTIE

La manipulation des données 231

CHAPITRE 13

Lire et enregistrer des données 233

Les préférences utilisateur	233
Obtenir une instance des préférences utilisateur	234
Enregistrer une valeur dans les préférences	234
Lire les valeurs des préférences	234
Permettre à l'utilisateur de modifier directement les préférences	235
Les fichiers de propriétés - plist	236
Le format plist	236
Lire un fichier de données plist	238
Écrire un fichier de données plist	238
Le format de données JSON	239
Intégrer JSON dans vos applications	240
Manipuler des données XML	242
Création d'un parseur XML	243
Gérer les événements XML	243
<i>Début du document.</i>	244
<i>Début d'un élément.</i>	244
<i>Récupérer le contenu texte des éléments</i>	245
<i>Repérer la fermeture d'un élément et enregistrer son contenu.</i>	246
Conclusion	246

CHAPITRE 14

Géolocalisation avec le MapKit..... 247

Zone affichée à l'écran (région)	248
Création d'un projet basé sur MapKit	248
Ajout du framework	248
Ajout d'une vue de type MKMapView	249
Géolocalisation de la vue	250
Positionnement de la région	251
Modification du style de carte	252
Modification de la valeur du zoom	253
Positionner une annotation	253
Suivre le mouvement de l'utilisateur	254
Conclusion	255

CHAPITRE 15

Communiquer avec l'extérieur..... 257

Premiers appels réseau synchrones	258
Modifier le comportement d'une requête synchrone	259
<i>Authentification</i>	259
<i>Gestion des redirections</i>	259
<i>Définir le délai d'attente d'une requête</i>	259
Réaliser des traitements en arrière-plan	259
Comprendre le thread principal	260
Lancer un traitement en arrière-plan	260
Particularités des traitements en arrière-plan	261
<i>Interactions avec l'interface</i>	261
Connexions réseau asynchrones	262
Préparation d'une requête asynchrone	263
Établissement de la connexion	264
Réception de données	264
Fin de connexion	264
Conclusion	265

CHAPITRE 16

Persistance d'objets avec Core Data..... 267

Introduction à l'ORM	268
Du monde objet au monde relationnel	268
Gestion des relations	268
Performances	268
Notion de contexte ou de session	269
Mise en place de l'environnement Core Data	269
Chargement de la description du modèle	270
Mise en place de l'entrepôt de stockage des données	271
Création du contexte	272
Description du modèle	272
Création d'un nouveau modèle	273
Édition du modèle	274
Création des classes du modèle	274
Manipulation d'objets gérés par le contexte	276
Création d'une nouvelle instance	276
Enregistrement des objets du contexte	276
Exécution d'une requête pour obtenir des objets	276
<i>Recherche d'une entité dans la base</i>	277
<i>Recherche basée sur un prédicat</i>	277

<i>Définir l'ordre des objets renvoyés</i>	277
<i>Aller plus loin avec les requêtes</i>	278
Supprimer un objet	278
Conclusion	278

CHAPITRE 17

Manipuler des données multimédias 279

Intégrer le son au cœur de vos applications	279
Les formats audio pris en charge par iOS	279
Convertir les fichiers audio pour l'iOS	280
Lancer la lecture de sons dans votre application	281
Lecture de vidéos	283
Formats de vidéos pris en charge	283
Intégrer le lecteur vidéo dans une application	283
S'abonner aux notifications pour suivre le déroulement de la lecture	285
Personnaliser le lecteur vidéo	286
Aller plus loin avec les vidéos sous iOS	286
<i>Proposer des vidéos live</i>	286
<i>Ajouter des éléments par-dessus la vidéo</i>	286
Accéder à la bibliothèque musicale de l'appareil	287
Parcourir la bibliothèque musicale de l'appareil	287
<i>Demander à l'utilisateur de choisir de la musique</i>	287
<i>Interroger directement la bibliothèque iPod</i>	289
Contrôler l'iPod depuis l'application	289
Tirer parti des photos et vidéos de l'utilisateur	290
Vérifier ce que permet le matériel	291
Paramétrer l'interface de prise de vue	291
Récupérer le média de l'utilisateur	292
Conclusion	293

CHAPITRE 18

Utiliser les API de notifications 295

Principe de fonctionnement d'APNS	295
Qu'est-ce qu'une notification ?	296
Prérequis pour l'utilisation du service de notification	296
Les notifications en quatre étapes	296
Étape 1 : inscription au service de notification	297
Étape 2 : transmettre le jeton APNS à votre serveur	298
Étape 3 : envoyer les notifications à votre application	300
Obtenir un certificat SSL pour le service APNS	300

Envoyer une notification depuis le serveur	302
<i>Préparer le message de notification</i>	302
<i>Envoi du message</i>	303
Étape 4 : recevoir les notifications	304
Réception des notifications quand l'application est fermée	304
Réception des notifications lorsque l'application est ouverte	304
Détecter les désinscriptions et les erreurs	305
Conclusion	305

CHAPITRE 19

Créer une bannière publicitaire iAd 307

Principe de fonctionnement d'iAd	308
Dimensions des publicités iAd	309
Exemple de mise en place d'iAd	309
Ajouter le framework iAd	309
Modifier le fichier d'en-tête	309
Implémentation dans le fichier .m	310
<i>Création de la bannière</i>	310
<i>Contact établi avec le serveur iAd</i>	310
<i>Actions lorsque la bannière est sélectionnée</i>	311
Gestion de la rotation avec iAd	312
Conclusion	313

CINQUIÈME PARTIE

La publication des applications315

CHAPITRE 20

Publier sur l'App Store 317

Préparer les éléments marketing en vue de la publication	317
Nom de société et langue principale	320
Le nom de l'application	321
Description de l'application	321
SKU : référence de l'application	321
Catégorie de l'application	322
Numéro de version de l'application	322
Dé détenteur des copyrights	322
Mots-clés	322
Informations de contact	322
Informations de démonstration (Demo account)	323
Contrat de licence	323

Niveau de contrôle parental de l'application (Ratings)	323
Pays de distribution	323
Éléments graphiques	324
<i>Icône de l'application</i>	324
<i>Captures d'écran de l'application</i>	324
Date de disponibilité de l'application	325
Prix de l'application	325
Localisation de votre application	325
Éléments techniques et dernières vérifications	326
Fournir l'application à Apple	326
Dernières vérifications techniques	326
<i>Mode de compilation et niveau de log</i>	326
<i>Vérifier le contenu du package applicatif</i>	327
<i>Respect de la charte graphique Apple</i>	327
<i>Messages d'erreurs réseau</i>	327
<i>Conserver le fichier de symbole</i>	328
Après la soumission	328
Modification des éléments marketing	328
Modification de l'application	328
En cas de rejet	329
Votre application est publiée	329
Suivre les progrès de votre application	329
<i>Statistiques de téléchargement</i>	329
<i>Les commentaires</i>	329
<i>Les rapports de crash</i>	330
Quelques conseils de lancement	330
<i>Utilisez vos canaux existants pour communiquer sur l'application</i>	330
<i>Communiquez auprès des blogs et des sites spécialisés</i>	331
<i>Utilisez le bouche-à-oreille</i>	331
<i>Utilisez les réseaux sociaux</i>	331
<i>Préparez une vidéo de démonstration</i>	331
<i>N'oubliez pas l'autopromotion</i>	332
Conclusion	332

PREMIÈRE PARTIE

Découverte de l'environnement de développement

Cette première partie constitue une introduction indispensable au développement iOS. Après un rappel des bases de l'Objective-C, elle donne un aperçu de l'environnement de développement pour créer et tester un premier exemple d'application simple.

Le **chapitre 1** présente les différents programmes développeurs, l'inscription et le téléchargement des outils et documentations. Le développeur y trouvera également des explications pour créer un certificat de développement, indispensable pour tester son application sur un iPhone.

Le **chapitre 2** est une introduction à l'Objective-C. Destiné à des développeurs familiers de la programmation orientée objet, il présente le langage d'une façon très pragmatique visant à vous rendre opérationnel rapidement.

Enfin, le **chapitre 3** permet au développeur de faire sa première application et de la tester dans le simulateur et sur son iPhone. Les outils indispensables comme Xcode et Interface Builder sont introduits en suivant quelques exemples très simples.

1

Développer pour iPhone et iPad

Le développement d'applications iPhone, iPod touch et/ou iPad sous iOS 5 est à la portée de tous les développeurs. Seuls un Mac, un iPhone et/ou un iPad et l'inscription au programme développeur Apple sont nécessaires pour développer son application, la tester et la publier.

Ce premier chapitre couvre les prérequis matériels et les connaissances qui seront utiles au développeur, avant d'accompagner le lecteur dans l'inscription à un des programmes développeurs iOS et à la création d'un certificat pour signer et distribuer des applications. Il est destiné aux développeurs, mais aussi au reste de l'équipe qui y trouvera comment s'inscrire pour accéder à la documentation, comment ajouter un appareil de test, etc.

Équipement matériel requis

Pour développer une application iPhone/iPad sous iOS, il faut disposer d'un Mac et d'un appareil de test.

Un Mac Intel pour développer

Officiellement, le développement d'applications iPhone avec le SDK Apple ne peut se faire que sur des Mac équipés d'un processeur Intel. En pratique, c'est la seule solution pour le développeur qui souhaite publier ses applications sur l'App Store.

Un iPhone, un iPod touch ou un iPad pour tester l'application

Avoir un iPhone, un iPod touch et/ou un iPad à disposition est indispensable. Les règles ergonomiques de la plate-forme et les contraintes liées à la taille de l'écran ne peuvent être comprises sans avoir l'appareil entre les mains. De plus, le simulateur possède quelques contraintes qui ne permettent pas de tout pouvoir programmer ; le système de notification, par exemple, n'est pas pris en charge par le simulateur.

Bien qu'un iPod touch puisse servir pour tester la plupart des applications, il ne permettra pas de tester votre application dans un contexte EDGE ou 3G (c'est-à-dire avec un débit très différent du Wi-Fi) et vous privera de certaines des applications auxquelles la plupart des utilisateurs sont très habitués, comme le téléphone.

	iPod touch 4ed	iPhone	iPhone 3	iPhone 3GS	iPhone 4	iPhone 4S	iPad	iPad 2	Nouvel iPad
CPU (MHz)	1000	412	412	600	1000	Dual 1000	1000	Dual 1000	Dual 1000 (4xGPU)
RAM (Mo)	256	128	128	256	512	512	256	512	1000
Résolution	640 x 960	320 x 480	320 x 480	320 x 480	640 x 960	640 x 960	768 x 1 024	768 x 1 024	1 536 x 2 048
Diagonale	3,5"	3,5"	3,5"	3,5"	3,5"	3,5"	9,7"	9,7"	9,7"
Capacité (Go)	8/32/64	4/8	8/16	16/32	16/32	16/32/64	16/32/64	16/32/64	16/32/64
MP	5/3(avant)	2	2	3	5/3 (avant)	8/3 (avant)	N/A	1/0,3 (avant)	5/0,3 (avant)
Vidéo	Oui	N/A	N/A	Oui	Oui	Oui	N/A	Oui	Oui
Durée Tél.	N/A	10 h	5 h	5 h	6 h	6 h	N/A	N/A	N/A
E/3G/W	W	E/W	E/3G/W	E/3G/W	E/3G/W	E/3G/W	E/3G/W	E/3G/W	E/3G/W/LTE
Réseau Max	N/A	384 Kbp/s	3,6 Mbp/s	7,2 Mbp/s	7,2 Mbp/s	7,2 Mbp/s	7,2 Mbp/s	7,2 Mbp/s	7,2Mbp/s
Bluetooth	2.1	2.0	2.0	2.1	2.1	4.0	2.1	2.1	4.0
OpenGL ES	2.0	1.1	1.1	2.0	2.0	2.0	2.0	2.0	2.0
Batterie	900 mAh	1400 mAh	1150 mAh	1219 mAh	1420 mAh	1420 mAh	6750 mAh	6930 mAh	11560mAh
Coque	Métallique	Métallique	N/B Plastique	N/B Plastique	N/B Métallique	N/B Métallique	Métallique	N/B Métallique	N/B Métallique
Dimensions	111 x 59 x 7	115 x 62 x 12	115 x 62 x 12	115 x 62 x 12	115 x 58 x 9	115 x 58 x 9	243 x 190 x 13	241 x 186 x 8,8	241 x 186 x 9,4
Poids (g)	101 g	135 g	133 g	135 g	137 g	140 g	680 g / 730 g	601 g / 613 g	652 g/662 g
Gyroscope	Oui				Oui	Oui		Oui	Oui

CONSEIL Utilisez quotidiennement votre appareil

Pour développer des applications iOS, il est fortement recommandé de posséder un iPhone ou un iPad (ou les deux), de l'utiliser comme téléphone principal et de télécharger fréquemment des applications. Gardez en permanence un regard curieux et critique sur les nouveautés de l'App Store, c'est votre première source d'inspiration. Il n'est pas envisageable de développer des applications pour iPhone/iPad sans être un utilisateur averti. Chaque jour, de nouvelles offres promotionnelles vous permettent de télécharger gratuitement certaines applications ou de les acheter à des tarifs vraiment très bas.

Compétences techniques utiles au développeur iOS

Le développeur d'applications iPhone/iPad doit maîtriser plusieurs connaissances. La plupart ne sont pas spécifiques au développement d'applications pour mobiles ou à l'environnement Mac et vous les avez peut-être apprises au préalable.

La programmation orientée objet, au cœur du SDK iOS

Une bonne maîtrise de la programmation orientée objet est un prérequis essentiel. Ce sujet ne sera pas repris dans ce livre.

L'héritage, la composition et les design patterns classiques doivent être maîtrisés, car ils sont utilisés de manière intensive dans tout le SDK iOS.

 H. Bersini, *La programmation orientée objet*, Eyrolles, 5^e édition, 2011.

L'Objective-C : un langage comme un autre

L'Objective-C est le langage imposé pour le développement d'applications iOS. Ce langage, bien qu'il provienne d'une évolution du langage C, est une nouveauté pour la plupart des développeurs arrivant sur la plate-forme iPhone/iPad et sa syntaxe peut sembler peu naturelle au premier contact.

VOUS VENEZ D'AUTRES LANGAGES Pour les développeurs Java, PHP et C#

Le développeur Java, C# ou PHP objet devrait retrouver rapidement ses marques. Le chapitre suivant, « L'essentiel d'Objective-C », présente, en partant de Java, les éléments essentiels d'Objective-C et de l'API standard : manipulation de chaînes, dates, dictionnaires, etc.

Programmation multithread

Toutes les applications iOS utiliseront plusieurs threads d'exécution. C'est grâce à eux, par exemple, que des contenus pourront être chargés en arrière-plan alors que l'interface reste réactive.

Le fait que plusieurs morceaux de code puissent accéder simultanément à la mémoire peut entraîner de subtils bogues, difficiles à reproduire. Le développeur doit donc bien visualiser l'exécution de l'application et comprendre par quels threads chaque morceau de code pourra être exécuté.

VOUS VENEZ D'AUTRES LANGAGES Synchronisation de threads

Les techniques de synchronisation entre threads en Objective-C ne sont pas différentes de celles des autres langages et le développeur ayant déjà une expérience de cette problématique ne sera pas surpris. Pour les autres, le chapitre 2 en présente les notions élémentaires, la documentation Apple reprenant également ce sujet.

Développement d'un « client lourd »

On parle de client lourd par opposition au client léger, qui n'embarque pas la logique métier de l'application. Dans une application web, le navigateur est un client léger qui ne prend en charge que l'interface, et la logique métier est exécutée dans un environnement totalement distinct : sur le serveur.

Une application iPhone/iPad est un client lourd qui embarque à la fois la logique d'affichage et la logique métier. La réunion des deux offre au développeur une maîtrise beaucoup plus grande de l'ergonomie, mais qui se paie par une augmentation de la complexité de l'application.

Il est néanmoins possible de développer des applications iPhone/iPad qui se comporteront comme des clients légers, n'embarquant que l'équivalent d'un navigateur de type Safari et s'appuyant sur un serveur web pour toute la partie métier. Ce type d'application web, même s'il est simple à réaliser, nécessite une connexion au réseau lors de son utilisation et une réactivité plus faible, due notamment à l'utilisation du réseau téléphonique ou Wi-Fi pour véhiculer les images.

Un développeur qui a déjà rencontré ce type de problématique, en développant des clients lourds avec Java/Swing ou C# par exemple, retrouvera facilement ses marques. Les autres doivent se préparer à un changement important dans la façon de concevoir l'application et les échanges avec l'utilisateur.

Concevoir des applications universelles

Le développeur iOS peut concevoir des applications dites « universelles » qui pourront s'exécuter aussi bien sur iPhone et iPod touch que sur iPad, tout en adaptant leur contenu en fonction de l'appareil. Le développeur ne programme qu'un seul projet pour l'ensemble des appareils.

L'adhésion au programme développeur d'Apple

L'adhésion au programme développeur d'Apple est nécessaire pour télécharger le SDK iPhone et l'installer. L'adhésion permet également d'accéder à toute la documentation, aux exemples de code, et aux vidéos de présentation Apple.

Il existe plusieurs modes d'adhésion en fonction du besoin :

- iOS Registered Developer ;
- iOS Developer Program à titre individuel ;
- iOS Developer Program au titre d'une entreprise ;
- iOS Developer Enterprise Program ;
- iOS Developer University Program.

Développeur iOS enregistré : un accès bon marché à l'environnement de développement et à la documentation

C'est le mode d'adhésion le plus simple et la première étape des autres programmes.

Devenir un développeur enregistré Apple est gratuit et vous permettra déjà de télécharger gratuitement l'environnement de développement afin de développer vos applications et de les tester, mais uniquement dans le simulateur.

Pour vous enregistrer, il suffit de vous rendre sur le site <http://developer.apple.com/> et de suivre le lien pour s'enregistrer. On vous demandera alors d'indiquer votre identifiant Apple (votre compte iCloud ou le compte utilisé pour acheter sur l'iTunes Store par exemple) ou d'en créer un et de répondre à quelques questions sur vos expériences précédentes de développement.

Le programme Developer pour tester et publier vos applications

L'adhésion au programme standard (*iPhone Developer Program*) vous permettra de tester vos applications sur iPhone et de les publier sur l'App Store. Elle est payante (99 \$ ou 79 €).

CONSEIL Développer pour un tiers

Si vous souhaitez développer des applications pour le compte d'une autre société, vous devez demander à votre client d'ouvrir son propre compte sur le programme développeur iOS et de vous ajouter comme développeur. C'est le seul moyen pour que l'application apparaisse avec le nom de votre client comme éditeur.

Les deux modes d'adhésion au programme iOS Developer

L'adhésion à ce programme peut se faire à titre individuel ou au nom d'une société.

Dans le premier cas, un seul développeur pourra utiliser ce compte pour créer des applications, les signer et les installer sur des iPhone.

Dans le second cas, vous pourrez enregistrer plusieurs développeurs associés à ce compte (on ne paie qu'une fois pour toute l'équipe) et distribuer les droits aux membres de l'équipe. C'est le mode recommandé pour toute équipe de développement.

Le processus d'adhésion au programme iOS Developer

Pour adhérer, il faut se rendre sur le site du programme développeur Apple et suivre le lien *Continue* :

► <http://developer.apple.com/programs/start/standard/>

ATTENTION Ne pas confondre le programme Developer et Entreprise

L'inscription au programme Developer au nom de votre entreprise (deuxième mode d'adhésion décrit ci-dessus) se fait en suivant le lien *Developer Program*. Le choix entre l'inscription à titre individuel ou au nom d'une entreprise se fait plus tard dans le processus d'inscription.

Ne confondez pas avec le programme *Enterprise Program* (299 \$) qui permet lui de diffuser des applications en interne au sein d'un grand groupe, sans passer par l'App Store.

Les étapes de l'adhésion pour un développeur individuel sont les suivantes.

- 1** Devenir un développeur iOS enregistré (voir paragraphe précédent).
- 2** Demander l'adhésion au programme Developer et répondre aux questions sur le site d'Apple.
- 3** Attendre la confirmation par courriel d'Apple (quelques jours).
- 4** Payer en ligne les frais d'adhésion.

Pour une adhésion au nom d'une entreprise, le processus est un peu plus compliqué :

- 1** Devenir un développeur iOS enregistré (voir paragraphe précédent).

- 2 Demander l'adhésion au programme Developer et répondre aux questions sur le site d'Apple – il faut indiquer le contact juridique de la société.
- 3 Attendre le courriel d'Apple au service juridique : Apple envoie après quelques jours un courriel au contact juridique en lui demandant d'envoyer l'extrait Kbis de l'entreprise par fax.
- 4 Attendre la confirmation par courriel d'Apple (quelques jours).
- 5 Payer en ligne les frais d'adhésion.

CONSEIL N'hésitez pas à contacter le support Apple aux développeurs

Dans certains cas, des demandes d'adhésion au nom d'une entreprise peuvent mettre longtemps avant d'être traitées, voire même rester sans réponse.

Le support Apple Developer Connection est très efficace et peut aider à connaître l'état d'une demande en cours. Leurs numéros de téléphone sont disponibles sur le site Apple :

► <http://developer.apple.com/contact/phone.html>

Pour la France, le numéro est : +33 (0) 800 90 7226.

Le programme Entreprise pour des applications internes

Le programme Entreprise (*iOS Enterprise Program*) permet à l'équipe de développement d'une entreprise de développer des applications pour une distribution et un usage interne (*In-House Distribution*).

Ce programme ne permet pas de distribuer des applications sur l'App Store.

Le programme universitaire pour l'enseignement

Ce programme gratuit permet à un enseignant de s'inscrire afin que ses étudiants puissent développer, tester sur leurs appareils et publier sur l'App Store. Il permet également aux étudiants d'échanger leurs applications entre eux.

Il est disponible aux États-Unis depuis 2008 et en France depuis le début de l'année 2009 pour quelques écoles et universités.

Les sites web Apple pour le développeur iPhone

L'adhésion au programme développeur iPhone vous donne accès à plusieurs sites web d'Apple.

Le centre de développement iOS

Le centre de développement iOS (*iOS Dev Center*) regroupe toute la documentation destinée aux développeurs.

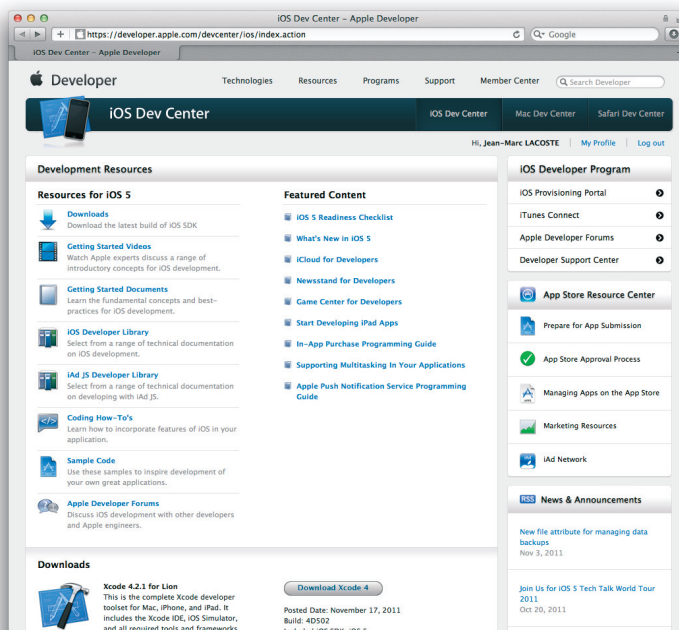
► <http://developer.apple.com/devcenter/ios/index.action/>

Vous y trouverez également des contenus vidéo, des exemples de code, et des liens pour télécharger la dernière version du SDK.

C'est enfin le point d'accès au portail du programme iOS.

Figure 1–1

Le centre de développement pour les développeurs d'applications iOS



Le portail du programme iOS

Ce site est accessible depuis le centre de développement (lien *iOS Developer Program Portal* en haut à droite). C'est un outil web qui régit tous vos échanges avec Apple avant la soumission de l'application.

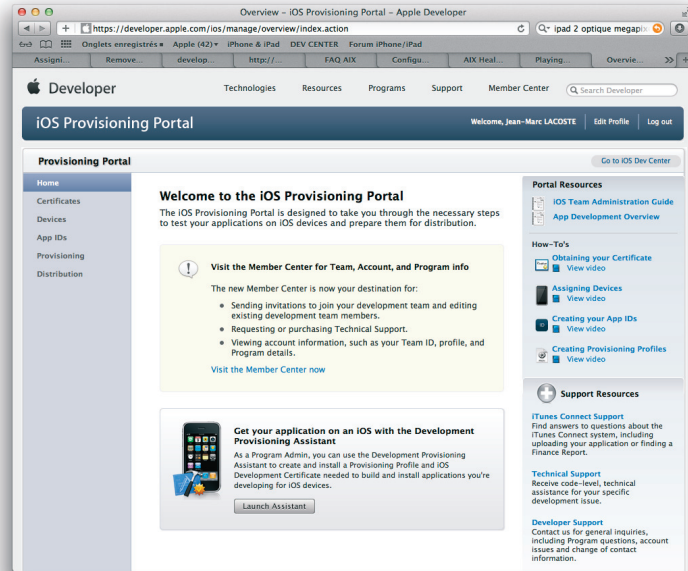
Il permet ainsi de :

- déclarer les membres de l'équipe de développement ;
- créer des certificats électroniques pour les développeurs ;
- déclarer les appareils que vous utiliserez pour tester les applications.

Nous reviendrons un peu plus loin sur cet outil indispensable aux développeurs.

Figure 1-2

Le portail du programme développeur iOS



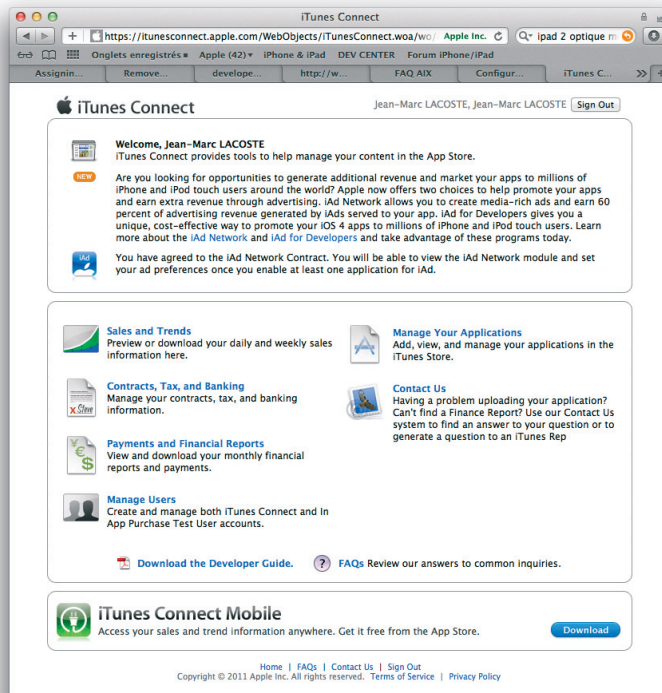
iTunes Connect, pour la publication des applications

iTunes Connect est l'outil utilisé pour publier des contenus vers Apple. Il est utilisé par l'industrie musicale pour publier de la musique sur iTunes, et vous l'utiliserez pour publier vos applications une fois satisfait de votre travail.

C'est également grâce à cet outil que vous suivrez les téléchargements de l'application, et dans le cas d'une application payante, vos revenus.

Figure 1-3

Le portail iTunes Connect pour distribuer vos applications sur l'App Store



Présentation du SDK iOS

Le SDK (*Software Development Kit*, kit de développement logiciel) est un package d'installation qui s'installe depuis le Mac App Store. Il s'obtient en lançant l'application intitulée *App Store* depuis le menu *Application* de votre Mac.

Une fois le téléchargement terminé, on peut installer Xcode en double-cliquant sur l'application *Install Xcode* qui se trouve dans le dossier */Applications*.

Tableau 1-1 Principaux composants installés avec le SDK

Nom du composant	Description
Xcode	L'outil de développement Apple permet la création de projets iOS, l'édition du code source Objective-C, la création graphique de storyboards, la compilation et le débogage des applications. Cet outil permet également de gérer les appareils et leurs certificats et d'accéder à la documentation.

Tableau 1–1 Principaux composants installés avec le SDK (suite)

Nom du composant	Description
SDK Mac OS X 10.7	L'ensemble du SDK standard Mac fait partie des pré-requis de l'installation.
iOS Simulator	Ce simulateur permet de tester les applications directement sur l'ordinateur.
Instruments	Cet outil permet d'analyser un programme pour surveiller l'état de la mémoire, l'utilisation du réseau, du CPU, etc.

Figure 1–4
Installation du SDK iOS



La documentation Apple, une aide à ne pas négliger

La documentation fournie par Apple est très riche, et nous vous recommandons de vous y référer. En voici un sommaire rapide permettant de retrouver l'information pertinente.

Les guides pour le développeur

Les guides font un tour d'horizon complet sur un sujet. Ils sont tous accessibles depuis le portail des développeurs (*iOS Dev Center*), en suivant le lien [iOS Reference Library](#).

Le guide consacré aux règles à respecter en matière d'ergonomie

Le guide *iOS Human Interface Guidelines* décrit les principes ergonomiques qui font de l'ensemble des appareils sous iOS une plate-forme uniforme dans laquelle les utilisateurs retrouvent facilement leurs marques.

La description des API et de la bibliothèque graphique

L'*iOS Application Programming Guide* présente les API les plus importantes de l'iPhone, les limitations imposées aux applications, et le fonctionnement de la bibliothèque graphique UIKit.

La référence Objective-C

L'*Objective-C 2.0 Programming Language* décrit le langage Objective-C et les nouveautés de sa version 2.0.

Les exemples de code : des projets Apple comme modèles

De nombreux exemples de projets sont fournis par Apple, chacun montrant comment utiliser une des API ou répondre à un problème classique.

Les exemples peuvent être téléchargés un par un en suivant le lien [Sample Code](#) depuis l'*iOS Dev Center*.

La documentation de référence exhaustive

La documentation la plus complète et la plus exhaustive couvre l'ensemble des API publiques de l'iOS. Elle peut être consultée en ligne ou bien téléchargée pour être lue directement depuis Xcode.

- 1 Lancer Xcode.
- 2 En haut à gauche, sélectionner [Organizer](#), puis [Documentation](#).
- 3 Sélectionner à gauche les éléments de recherche désirés.

Prérequis pour la distribution d'une application

Une des révolutions apportées par les appareils sous iOS et le SDK est le modèle de distribution intégré au terminal qui contribue fortement au succès des appareils sous iOS et de leurs applications.

Il existe deux modes de distribution des applications : le mode Ad Hoc et le mode de publication par l'App Store. Quel que soit celui retenu, la sécurité des applications est assurée par une signature électronique.