

ANDROID

**Apprenez à développer efficacement
pour le leader des OS mobiles**

Florent Garin

*Gérant fondateur de la société DocDoku
Certifié Sun Java 2 Programmer
et Sun Enterprise Architect for J2EE*

3^e édition

DUNOD

Toutes les marques citées dans cet ouvrage sont des marques déposées par leurs propriétaires respectifs.

Illustration et maquette de couverture : WIP

Le logo du robot utilisé sur la couverture est reproduit à partir d'une œuvre créée et partagée par l'Android Open Source Project. Il est utilisé conformément aux termes de la *Creative Commons 2.5 Attribution License*.

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du

Centre français d'exploitation du droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, Paris, 2009, 2011, 2012
ISBN 978-2-10-058510-6

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Table des matières

Avant-propos	XI
Chapitre 1 – Présentation	1
1.1 Qu'y a-t-il dans la boîte ?	1
1.2 L'Open Handset Alliance	1
1.2.1 Les constructeurs	2
1.2.2 La licence	3
1.3 Les applications de base	3
1.4 Positionnement par rapport à son environnement	5
1.4.1 L'iPhone	6
1.4.2 Nokia	7
1.4.3 Windows Mobile	7
1.4.4 BlackBerry	7
1.4.5 HP WebOS	8
1.4.6 JavaFx	10
1.4.7 Flash/Flex	10
1.4.8 JavaME	11
1.4.9 Les autres	11
Chapitre 2 – Architecture logicielle	13
2.1 Un Linux sous le capot	13

2.2	Des bibliothèques C/C++	13
2.3	Un middleware Java	15
2.4	Dalvik	15
2.5	Le JDK	16
Chapitre 3 – Le modèle de programmation		19
3.1	Un développement presque classique	19
3.1.1	<i>Règles de codage</i>	20
3.2	Le SDK Android	22
3.2.1	<i>Plug-in Eclipse</i>	23
3.2.2	<i>Scripts Ant</i>	26
3.3	Développer, exécuter et déboguer	26
3.3.1	<i>Installation</i>	26
3.3.2	<i>Dalvik Debug Monitor Service (DDMS)</i>	27
3.4	Structure d'un projet	28
3.4.1	<i>Le manifest</i>	30
3.4.2	<i>Les ressources</i>	38
3.5	Le NDK (Native Development Kit)	43
3.5.1	<i>Présentation</i>	43
3.5.2	<i>Installation</i>	43
3.5.3	<i>Développement</i>	44
3.6	Et les WebApp ?	45
Chapitre 4 – Construire l'interface graphique		49
4.1	Le rendu visuel	49
4.2	Approche programmatique ou déclarative	50
4.3	Les composants graphiques	55
4.3.1	<i>TextView</i>	55
4.3.2	<i>EditText</i>	56
4.3.3	<i>CheckBox</i>	57
4.3.4	<i>ToggleButton</i>	58
4.3.5	<i>RadioGroup</i>	59
4.3.6	<i>Button</i>	61

4.3.7	<i>Spinner</i>	62
4.3.8	<i>AutoCompleteTextView</i>	63
4.3.9	<i>DatePicker</i>	65
4.3.10	<i>TimePicker</i>	65
4.3.11	<i>Gallery</i>	66
4.3.12	<i>ImageView</i>	69
4.3.13	<i>ImageButton</i>	70
4.3.14	<i>ProgressBar</i>	71
4.3.15	<i>AnalogClock</i>	73
4.3.16	<i>DigitalClock</i>	73
4.3.17	<i>RatingBar</i>	74
4.4	Les layouts	75
4.4.1	<i>ListView</i>	75
4.4.2	<i>FrameLayout</i>	76
4.4.3	<i>LinearLayout</i>	76
4.4.4	<i>TableLayout</i>	76
4.4.5	<i>RelativeLayout</i>	77
Chapitre 5	Le modèle de composants	79
5.1	Une forte modularité	79
5.2	Quatre familles de composants	80
5.3	La navigation entre activités	80
5.3.1	<i>L'objet Intent</i>	81
5.3.2	<i>La résolution</i>	82
5.3.3	<i>La communication entre activités</i>	85
5.3.4	<i>Gestion du cycle de vie des activités</i>	87
5.3.5	<i>Les fragments</i>	89
5.4	Les services	91
5.4.1	<i>Le langage AIDL</i>	92
5.4.2	<i>Implémentation du service</i>	96
5.4.3	<i>Publication du service</i>	97
5.4.4	<i>Côté client</i>	98
5.4.5	<i>Started service</i>	100

5.5	Le bus de messages	100
5.5.1	Deux modes d'émission	101
5.5.2	Deux façons de s'abonner aux événements	102
5.5.3	Implémentation du receiver	103
5.5.4	Une durée de vie très courte	105
Chapitre 6 – La persistance des données		109
6.1	Sauvegarder l'état des applications	109
6.2	Système de fichiers	110
6.2.1	Lecture d'un fichier	110
6.2.2	Écrire dans un fichier	111
6.3	Les préférences utilisateur	112
6.3.1	Lecture des préférences	113
6.3.2	Écriture des préférences	114
6.3.3	IHM de configuration	114
6.4	SQLite	116
6.4.1	Optimisation des performances	118
6.5	Exposer ses données	119
6.5.1	Utiliser les providers	119
6.5.2	Modifier les données du content provider	122
6.5.3	Créer son propre ContentProvider	123
Chapitre 7 – Conception générale		129
7.1	Architecture logicielle	129
7.1.1	Développer des applications réactives	129
7.1.2	La conception générale	132
7.1.3	Véhiculer des données entre composants	134
7.1.4	Le maître mot : modularité	136
7.2	Infrastructure back office	136
7.2.1	Serveurs d'applications	136
7.2.2	Google App Engine	136
7.2.3	CouchDB	138

Chapitre 8 – Fonctions IHM poussées	141
8.1 Les menus	141
8.1.1 Les « Option Menus »	142
8.1.2 Les « Context Menus »	144
8.1.3 Les « Popup Menus »	146
8.2 Étendre les composants existants	147
8.3 Les animations	148
8.4 Personnalisation en fonction de la configuration	151
8.4.1 Les ressources alternatives	151
8.5 Notifier l'utilisateur	154
8.5.1 Le Toast	155
8.5.2 Barre de statut	156
8.5.3 Les boîtes de dialogue	158
8.6 2D et 3D	160
8.6.1 Graphisme en deux dimensions	161
8.6.2 Graphisme en trois dimensions	164
8.7 App Widgets	165
8.7.1 Définition de l'IHM	166
8.7.2 AppWidgetProvider	167
8.7.3 Écran de configuration	170
8.8 La notion de Task	171
8.9 Styles et thèmes	171
8.10 La recherche Android	173
Chapitre 9 – Interaction avec le matériel	177
9.1 Les fonctions de téléphonie	177
9.1.1 Les appels vocaux	177
9.1.2 Réception de SMS	179
9.1.3 Envoi de SMS	180
9.2 Géolocalisation	182
9.2.1 Service de localisation	182
9.2.2 API de cartographie	184
9.2.3 La classe MapView	188

9.3	API réseau bas niveau	192
9.3.1	EDGE et 3G	192
9.3.2	WiFi	193
9.3.3	Bluetooth	194
9.3.4	NFC (<i>Near Field Communication</i>)	194
9.4	L'appareil photo	196
9.5	API Media	198
9.5.1	MediaPlayer	199
9.5.2	MediaRecorder	201
9.5.3	JET Engine	202
9.6	Le vibreur	202
9.7	L'écran tactile	203
9.8	L'accéléromètre	204
Chapitre 10 – Le réseau		207
10.1	Connexion à des serveurs distants	207
10.1.1	Les bases de données	207
10.1.2	Quels protocoles réseaux utiliser ?	209
10.1.3	Les web services	209
10.1.4	SOAP	210
10.1.5	POX (<i>Plain Old XML</i>)	212
10.1.6	JSON (<i>JavaScript Object Notation</i>)	213
10.1.7	XMPP	218
Chapitre 11 – Sécurité et déploiement		223
11.1	Signer les applications	223
11.1.1	La phase de développement	223
11.1.2	La phase de packaging	224
11.2	Publier son application	228
11.2.1	Gestion des versions	228
11.2.2	Google Play Android	229

Chapitre 12 – Tablettes et autres appareils	231
12.1 Les tablettes tactiles	231
12.1.1 Différences avec le mobile	232
12.1.2 Android 3.x et 4.x	233
12.2 La fragmentation	233
12.2.1 De nombreux frameworks	234
12.2.2 Les applications hybrides	234
12.3 Le futur : HTML5 et CSS3	237
12.3.1 HTML5=HTML+JavaScript+CSS	238
Index	245

Avant-propos

Cela fait maintenant un peu plus de 5 ans que l'iPhone, premier du nom est sorti. Je me souviens encore de cette fameuse *keynote*, l'une des plus mémorables, où Steve Jobs a présenté ce nouveau téléphone révolutionnaire. « *Apple réinvente le téléphone* », jamais avare de superlatifs, je crois toutefois que ce coup-ci il n'avait pas tort.

Un an plus tard, fin d'année 2008, le premier téléphone Android, l'HTC G1 était mis sur le marché. À l'époque, le système Android, issu du rachat en 2005 par Google de la société du même nom, n'était pas en toute honnêteté au même niveau que l'OS d'Apple. Mais on sentait le potentiel : la technologie et la motivation de Google bien conscient de l'importance vitale de ce secteur naissant qu'était le mobile. Par ailleurs l'approche était totalement différente, la stratégie de Google a toujours été de s'assurer que ses services soient parfaitement accessibles sur les terminaux mobiles, qui, on le rappelle, devraient devenir rapidement le premier mode d'accès à internet.

Cinq ans plus tard, on peut dresser un premier bilan de la féroce bataille qui s'est véritablement engagée en 2007 pour le leadership des dispositifs nomades. La première chose qui frappe est à quel point le paysage industriel a été profondément remodelé. Les champions d'hier sont aujourd'hui en grande difficulté, voire luttent purement et simplement pour leur survie.

Nokia en 2007 dominait largement le marché des téléphones mobiles avec près de 40 % des ventes. RIM (*Research In Motion*) positionné dès ses origines sur le haut de gamme et le monde professionnel était avec ses BlackBerry le roi des smartphones. Seulement voilà, dans un marché aussi dynamique et agressif que celui qui nous intéresse ici, il est crucial d'innover.

Apple a montré la voie en imposant le tactile comme le mode d'interaction standard et en popularisant le concept de boutique en ligne, facilitant au maximum la distribution d'applications et de contenu multimédia. Google lui a emboîté le pas, de version en version son OS ne fait que s'améliorer, Android a également su innover et propose des fonctionnalités inexistantes sur l'OS de son grand concurrent comme les applications tournant en tâche de fond comme de vrais services et les gadgets que l'on peut placer sur le bureau. Google a comme Apple parfaitement compris que le design de son système d'exploitation et des applications amenées à tourner dessus devait être

irréprochable, un site entier a récemment été mis en place (<http://developer.android.com/design/index.html>) pour accompagner les développeurs dans cette tâche.

Sa stratégie ouverte et multi-constructeurs s'est également révélée redoutable, aujourd'hui Android est l'OS mobile le plus répandu de la planète. Avec l'informatique de demain qui se joue sur les mobiles et les tablettes, et l'incapacité de Microsoft à percer sur ce créneau, Android serait-il devenu le nouveau Windows ?

Sur le plan technique, Android est un sujet d'étude très vaste qui mérite un livre à part entière. En effet, son modèle de programmation, son interface graphique, ses composants logiciels, ses fonctionnalités de sauvegarde de données ou encore ses API réseau et de géolocalisation sont uniques.

Ce livre s'adresse à tous ceux désireux de développer des applications professionnelles ou de loisirs fonctionnant sous Android. Il a pour ambition d'accompagner le lecteur du téléchargement du SDK (*Software Development Kit*) au déploiement du programme sur le téléphone. L'ouvrage a pour objectif d'être très pratique, les explications didactiques sont le plus possible étayées d'exemples concrets. Même s'il est préférable de lire l'ouvrage séquentiellement, il est toutefois possible de le parcourir chapitre par chapitre, une fois que les composants essentiels seront maîtrisés.

Chaque chapitre traite d'un aspect spécifique d'Android accompagné d'exemples de code concrets dont les projets Eclipse qui sont téléchargeables sur la page dédiée à l'ouvrage sous la rubrique « compléments en ligne » du site Internet de Dunod, www.dunod.com.

1

Présentation

Objectifs

Ce chapitre permet de faire connaissance avec Android. Il décrit l'organisme chargé de son développement, sa licence, son environnement technologique, pour terminer avec les applications proposées par défaut sur les terminaux Android.

1.1 QU'Y A-T-IL DANS LA BOÎTE ?

Définir Android n'est pas chose aisée tant les concepts derrière cet intitulé unique sont nombreux. Pour commencer et en simplifiant à l'extrême, on peut dire qu'Android est un système d'exploitation pour mobile open source développé par la société Google. Tout au long de ce livre, les nombreuses facettes de ce système seront mises à nu élargissant ainsi la définition d'Android qui est étudié ici jusqu'à sa version 4.1.

1.2 L'OPEN HANDSET ALLIANCE

Dès son origine, la démarche de Google a été d'ouvrir le développement d'Android en rassemblant autour de lui et au travers de l'Open Handset Alliance (OHA) un maximum de sociétés. Les membres de ce consortium sont très variés : on y trouve des fabricants de téléphones connus tels que Sony Ericsson, Samsung ou Motorola, des opérateurs de téléphonie comme Sprint, T-Mobile ou NTT DoCoMo, des sociétés Internet, Google évidemment mais aussi eBay, des constructeurs de puces électroniques Intel, nVidia, ou encore des acteurs du marché du GPS comme Garmin.

Toutes ces entités se retrouvent donc au sein de cette alliance, pour des raisons qui leur sont propres, pour œuvrer au développement d'Android.

1.2.1 Les constructeurs

Parmi les membres de l'Open Handset Alliance, il y a une catégorie qui intéressera plus particulièrement les consommateurs, c'est celle des constructeurs. C'est en effet dans cette liste qu'il faudra chercher son futur combiné téléphonique. Il est donc crucial pour le succès de la plateforme que cette liste soit bien fournie avec de préférence des poids lourds du secteur. Aujourd'hui, on peut recenser : HTC, Motorola, Samsung, Sony Ericsson, Toshiba, LG, Huawei, Asus.

En 2009, le choix de terminaux était encore très limité. Seul le Taïwanais HTC avait véritablement dégainé en proposant le Dream (G1) dès fin 2008 puis son évolution le Magic (G2).



Figure 1.1 – G1 HTC

Au premier trimestre 2009, les choses se sont accélérées fortement. Les constructeurs ont fait de nombreuses annonces, finalement l'année 2010 s'est terminée avec plusieurs dizaines de terminaux sous Android. Aujourd'hui, en 2012, Android a explosé pour devenir l'OS numéro un dans le monde avec plus de 50 % de parts de marché. Grâce lui d'ailleurs, Samsung est dorénavant le premier fabricant de téléphones de la planète, devant un Nokia déclinant et devant aussi Apple, le seul vrai rival.

En outre, après les téléphones et les tablettes, il est intéressant de noter l'initiative d'Archos qui intègre Android dans des radios-réveils ou des téléphones fixes. Asus, avec ses Transformer, propose des appareils hybrides tablette/netbook. Manifestement la plateforme se sentait à l'étroit sur les téléphones portables, on peut probablement anticiper que les prérogatives d'Android pourraient s'étendre à tout type d'appareil remplissant des fonctions multimédias et ayant des capacités de communication importante.



Figure 1.2 – Galaxy S3 (Quad Core)

1.2.2 La licence

Comme il est mentionné sur le site officiel, la licence préférée d'Android est l'« Apache open source licence v2 ». Le terme « préférée » signifie que le projet Android pourra dans une certaine mesure accepter et intégrer du code source n'ayant pas été publié sous cette licence. Néanmoins, de façon prévisible, seules les licences « Open Source » approuvées par l'Open Source Initiative – organisation dont l'objet est de labéliser les licences « Open Source » – sont acceptées. Les contributeurs directs au projet, individus ou sociétés, devront, par contre, signer (il est possible de le faire en ligne) un agrément de licence.

La licence Apache a été choisie pour son côté « *business friendly* » comme on dit. Plus clairement, la licence Apache, contrairement aux licences GPL, autorise la reprise de tout ou partie du code dans un produit sous une autre licence y compris propriétaire.

1.3 LES APPLICATIONS DE BASE

Les terminaux Android de base sont livrés avec un ensemble d'applications dénommées « *Core Applications* ». En fonction des constructeurs, d'autres logiciels pourront être pré-installés sur le téléphone mais les Core Applications constituent un socle commun minimum obligatoire.

Comme Android est destiné à motoriser en premier lieu des téléphones, parmi ces applications se trouvent tout logiquement le composeur de numéro et la gestion des contacts :

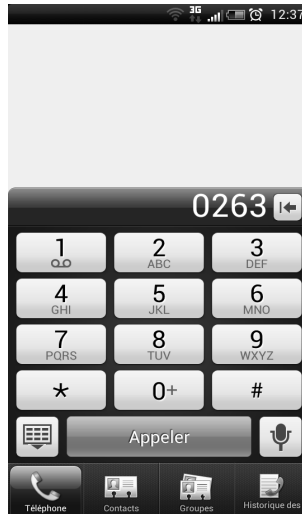


Figure 1.3 – Composeur

Ensuite viennent des applications plus sophistiquées comme Google Maps.



Figure 1.4 – Application de cartographie

Couplée au GPS si l'appareil en est doté (attention, les spécifications d'Android n'imposent pas que tous les téléphones soient équipés d'un système de géolocalisation),

l'application Maps prend tout son sens et montre bien de quoi la plateforme est capable.

Autre application phare, le navigateur bien sûr.



Figure 1.5 – Un navigateur très performant

Il s'agit d'un vrai navigateur qui n'a rien à voir avec les butineurs wap d'il y a quelques années. Motorisé par WebKit (le cœur de Safari, le navigateur du Mac), ce navigateur est en mesure d'interpréter l'HTML, le CSS, le JavaScript dans leur dernière version, au même titre que ceux pour les ordinateurs de bureau.

Grâce à un système de zoom, il est possible de consulter tous les sites. Néanmoins pour une utilisation agréable, il vaut mieux qu'une version mobile du site, à l'ergonomie adaptée, soit disponible.

1.4 POSITIONNEMENT PAR RAPPORT À SON ENVIRONNEMENT

Android est arrivé sur un marché déjà bien encombré. La concurrence y est féroce, Nokia dominait le secteur avec un peu plus d'un tiers de parts de marché. Toutefois Apple, en faisant une entrée fracassante, a prouvé que les positions étaient loin d'être figées et qu'un nouveau venu avait sa chance à condition bien sûr d'innover. D'autre part, l'approche ouverte et multiconstructeurs d'Android conjuguée à la touche Google laissait espérer un avenir radieux à la plateforme, l'histoire n'a fait que confirmer ces bons pressentiments.

1.4.1 L'iPhone

L'iPhone, sans aucune contestation possible, est un téléphone qui fera date dans l'histoire de cette industrie. Apple a fixé en partie certains standards pour les téléphones de demain : le large écran tactile, de préférence multipoints, paraît incontournable, l'accéléromètre aussi. L'esthétique de l'interface graphique, la fluidité des animations, les zooms sur les images ou les cartes faits avec deux doigts posés sur l'écran ont donné brutalement un sacré coup de vieux à la concurrence.

Il semble aussi certain aujourd'hui qu'un smartphone se doit de disposer de sa boutique en ligne ou « App Store ». Pour Apple, cette dernière est un beau succès : début 2012 la barre des 25 milliards de téléchargements a été franchie. Un écosystème s'est ainsi créé et des milliers de développeurs amateurs ou professionnels tentent leur chance en proposant leurs créations.

Malgré toutes les qualités de l'iPhone, celui-ci est loin d'être exempt de défauts. Au-delà des classiques reproches qui lui sont faits sur l'impossibilité de disposer des widgets sur l'écran d'accueil¹ ou sur le peu de profils Bluetooth supportés (pas de transfert de fichiers), au bout du compte, le problème numéro un de l'iPhone est son aspect fermé !

Pas de doute là-dessus, l'iPhone est bien un produit de la firme à la pomme. Au moment de la rédaction de ce livre, le SDK (*Software Development Kit*), certes gratuit, n'était disponible que pour les seuls possesseurs de Mac, pas de version Windows ni Linux. Pour publier ses applications, le développeur a le choix entre le « Standard Program » à 99 \$ et l'« Enterprise Program » à 299 \$. Dans le premier cas, la distribution se fera au travers de l'App Store accessible publiquement par iTunes depuis un ordinateur (PC ou Mac) ou directement depuis le téléphone. Dans le second cas, le programme sera déployé depuis les serveurs internes de l'entreprise. Ce qui permet de distribuer les logiciels propriétaires métiers.

Ce qui est gênant dans le système Apple, ce n'est pas tant le prix de 99 \$, après tout les coûts liés à l'hébergement et l'exploitation de la boutique en ligne sont bien réels. Le problème vient de la position centrale et incontournable d'Apple. Contrairement à Android (voir chapitre 11, « Sécurité et déploiement »), la distribution d'applications pour l'iPhone ne peut se faire que par iTunes, chasse gardée d'Apple. Ce dernier se réserve donc le droit d'accepter ou de refuser, sans fournir aucune explication, les logiciels dans son App Store. Le logiciel NetShare en a fait les frais, tout d'abord autorisé à intégrer l'App Store puis retiré. Ce logiciel est un proxy SOCKS grâce auquel on peut se servir de son iPhone comme d'un modem concurrençant au passage les clés 3G des opérateurs ! C'est probablement la raison de son retrait.

1. L'écran d'accueil de l'iPhone, dénommé *Springboard* reste obstinément statique, la différence avec Android est flagrante. Celui-ci accepte des fonds d'écran animés et peut afficher sur ses « *home screens* » des widgets, sorte de mini-applications se mettant à jour de manière autonome. Sur ce plan-là, Apple a toujours été très conservateur et n'a pour l'instant jamais voulu complexifier son IHM ni sacrifier la charge de la batterie pour un écran d'accueil plus sophistiqué.

Cette politique unilatérale d'Apple fait couler beaucoup d'encre. Les plus gentils disent que c'est pour garantir une certaine qualité et s'assurer que les programmes en question ne sont pas des malwares, d'autres crient purement et simplement à la censure en arguant qu'Apple ne fait que défendre ses intérêts commerciaux et ceux de ses partenaires en se réservant les applications les plus lucratives.

1.4.2 Nokia

Nokia était il y a encore peu d'années le plus grand constructeur de téléphonie du monde devant Motorola. Sa gamme est très large et couvre tous les segments : du premier prix vendu 1 € avec un abonnement auprès d'un opérateur jusqu'au mobile PDA avec GPS et enregistreur vidéo. Les Nokia sont généralement réputés pour leur grande fiabilité. Leur système d'exploitation était le Symbian OS développé initialement par la société éponyme passée propriété exclusive de Nokia. Mais l'arrivée de l'iPhone a marqué le début de la descente vertigineuse de la marque finlandaise. Nokia n'a pas vu venir l'ère du tactile ni celui des smartphones tournés vers les applications et les nouveaux usages qui vont avec.

Un temps associé à Intel pour la conception du système MeeGo qui devait être le remplacement de Symbian, Nokia s'est finalement tourné vers Microsoft et joue aujourd'hui sa survie en misant tout sur Windows Mobile. La future déclinaison de son Lumia embarquant la version 8 de Windows Phone se doit d'être un succès.

1.4.3 Windows Mobile

Windows Mobile, WiMo pour les intimes, est l'OS mobile de Microsoft. C'est une évolution de Windows Pocket PC, ancêtre de Windows CE. À l'époque cet OS, sans avoir jamais déchaîné les passions, avait réussi au fil des années à s'octroyer une part de marché honorable, il faut dire qu'il n'y avait pas la même concurrence qu'aujourd'hui.

Son intérêt avait sans doute été suscité par son affiliation à la famille d'OS Windows, ultra-dominante sur le bureau. Un autre avantage souvent cité est la facilité de développement apportée grâce à l'environnement cliquodrome de Visual Studio qui a su faire venir au développement mobile les développeurs VB.

Aujourd'hui, Microsoft met en avant, entre autres, son intégration avec Exchange qui, par le biais du protocole Exchange ActiveSync, offre la fameuse fonctionnalité de push mail. Les Windows phones s'interfaçent aussi nativement avec le portail d'entreprise SharePoint et la suite bureautique Office.

1.4.4 BlackBerry

Tout comme les Nokias, le BlackBerry a aussi été un téléphone très en vue. Historiquement positionné sur le marché des entreprises, la fonction majeure qui a fait décoller le BlackBerry au début des années 2000 était le *push mail*. Grâce à elle, l'utilisateur n'a plus besoin de consulter périodiquement sa boîte pour vérifier s'il n'a pas de nouveaux messages, ceux-ci lui parviennent directement comme de banals SMS. Cette fonctionnalité est assurée par les serveurs d'infrastructure du fabricant

RIM (*Research In Motion*) avec un protocole propriétaire. Le mail fut donc le point fort des BlackBerry qui faisaient rêver de nombreux cadres et dirigeants, même Barack Obama en était paraît-il accroc !



Figure 1.6 – BlackBerry Torch 9810

Aujourd'hui, les lignes ont fortement bougé, l'iPhone s'est rapproché du monde de l'entreprise en proposant aussi une fonctionnalité de push mail avec le protocole d'Exchange ActiveSync et les Androids synchronisent en tâche de fond mails, calendriers, contacts, twitter, informations de météo...

BlackBerry, de son côté, a fait un peu le chemin inverse en se dotant de capacité multimédia et d'une ergonomie tactile avec le BlackBerry Storm. RIM se rapproche ainsi du grand public ou plutôt veille à ce que sa clientèle d'entreprise ne soit pas trop attirée par l'élégance de l'iPhone.

Android, quant à lui, n'étant pas porté par un unique constructeur, ne cible pas spécifiquement les particuliers ou le monde professionnel. La plateforme est généraliste, on peut y développer aussi bien des jeux que des applications métiers.

La nouvelle version de BlackBerry OS (v10) sera une évolution majeure, elle incorporera les apports du système QNX (racheté en 2010) et proposera plusieurs SDK dont un en HTML5 (WebWorks). À l'instar de Nokia, BlackBerry joue gros avec la prochaine édition de son système.

1.4.5 HP WebOS

Après avoir connu son heure de gloire en étant l'une des premières sociétés à commercialiser des assistants numériques, Palm était depuis sur le déclin. Il y a

quelques années, Palm a même cédé aux sirènes de Windows Mobile en proposant certains de ses appareils sous l'OS de Microsoft. Palm avait cessé d'innover et se devait de réagir face aux assauts d'Apple et de Google.

La réponse s'est appelée Palm Pré. Ce téléphone a été le premier à tourner sous le nouvel OS de Palm, WebOS. WebOS a été dévoilé à l'occasion du CES 2009 de Las Vegas. Et il faut reconnaître que l'accueil de l'assistance fut plus que chaleureux. On a même pu assister à un certain buzz dans les jours qui ont suivi. L'intérêt qu'a suscité à nouveau Palm était mérité. Le Palm Pré avait un look et une ergonomie très attractifs. Visuellement WebOS était plutôt réussi, il y a même eu une polémique sur sa ressemblance, il y est vrai assez poussée, avec l'iPhone. Le fait que certains hauts dirigeants de Palm soient sortis tout droit de l'entreprise de Cupertino n'a pas contribué à calmer les esprits.



Figure 1.7 — Le Palm Pré et son look arrondi
(Photo de whatleydude (Flickr) sous licence Creative Commons 2.0)

Quoi qu'il en soit, si Palm s'est plus ou moins inspiré de la concurrence (ce qui est en fin de compte dans l'ordre des choses), il a eu la bonne idée d'y adjoindre quelques améliorations : une batterie rechargeable, un chargeur sans fil, un clavier amovible ou encore le Bluetooth. L'OS lui-même parut aussi engageant : il supportait le multitâche, un lecteur flash, une intégration web omniprésente et surtout Palm insistait particulièrement sur le framework de développement (Mojo SDK) qui s'appuyait sur JavaScript, JSON, HTML, XML et CSS, en somme que des technologies « web ».

Cette approche est séduisante car elle permet de recruter des développeurs plus facilement en piochant dans le vivier des aficionados du web. Toutefois penser que l'on pourrait coder pour WebOS sans effort est un raccourci un peu rapide. Il faut évidemment se familiariser avec l'API de la plateforme.

Aujourd'hui Palm a rejoint le giron d'HP car commercialement la sauce n'a jamais pris. HP avait de grands plans pour cet OS, il comptait en faire le système phare pour équiper non seulement ses smartphones et ses tablettes mais aussi ses imprimantes. Malheureusement, là non plus cela n'a pas fonctionné, la multinationale prévoit aujourd'hui de placer le code de WebOS en open source d'ici la fin de l'année 2012.