

Developer success in the Google Android Market
and Amazon Appstore for Android



The Business of Android Apps Development

Making and Marketing Apps that Succeed

Mark Rollins

Apress®

The Business of Android Apps Development

Making and Marketing Apps That Succeed



Mark Rollins

Apress®

The Business of Android Apps Development: Making and Marketing Apps That Succeed

Copyright © 2011 by Mark Rollins

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-3942-0

ISBN-13 (electronic): 978-1-4302-3943-7

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

President and Publisher: Paul Manning

Lead Editor: Tom Welsh

Technical Reviewer: Bradley D. Brown

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Engel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Corbin Collins

Copy Editor: Damon Larson

Compositor: Bytheway Publishing Services

Indexer: SPI Global

Artist: SPI Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/bulk-sales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at www.apress.com.

Contents at a Glance

■ About the Author	xi
■ About the Technical Reviewer	xii
■ Acknowledgments	xiii
■ Preface	xiv
■ Chapter 1: The Android Application Business	1
■ Chapter 2: Questions to Ask About Your Android Application	23
■ Chapter 3: You’re Not Selling an Application, You’re Selling a Culture	39
■ Chapter 4: Marketing Your Android Application	51
■ Chapter 5: How to Promote Android Applications Within Android Applications ...	63
■ Chapter 6: Marketing the Free Application	73
■ Chapter 7: In-App Billing: Putting a Store in Your Application	91
■ Chapter 8: The Prerelease Guide	105
■ Chapter 9: Publishing to the Android Market	115
■ Chapter 10: Best Launch Day Ever	145
■ Chapter 11: Keep the Application Fires Burning	161
■ Chapter 12: Programming in Android with the Eclipse IDE	181
■ Index	227

Contents

■ About the Author	xi
■ About the Technical Reviewer	xii
■ Acknowledgments	xiii
■ Preface	xiv
■ Chapter 1: The Android Application Business	1
The State of the Android Market.....	1
The Origins of Android	2
Android vs. iOS	5
Porting Difficulties	6
Examples of Cross-Platform Development Tools.....	6
Getting Started As an Android Developer	8
The JDK	8
The Android SDK.....	10
The Eclipse IDE	13
How to Use Your Own Android Device As an Emulator on Eclipse	20
Summary	21
■ Chapter 2: Questions to Ask About Your Android Application	23
How Android Gives Developers the Tools to Succeed	23
Deciding What to Build	24
Creating an Original Application	26
What Do You Want Your App to Do?	26

Is the Application Simple to Use?	27
What Problem Does Your Android Application Solve?	28
Who Is Your Target Audience?	29
What Category Will This Application Go Under?	29
Has Someone Already Done This Before?	30
What Are You Going to Call Your Application?	33
Where Do You Want Your Application to Run.....	34
What You Need to Do, Legally, As an Application Developer	35
Copyrights	35
Trademarks	36
Protecting Your Trade Secrets.....	36
Is a Patent Right for You?	36
Summary	37
■ Chapter 3: You’re Not Selling an Application, You’re Selling a Culture	39
The Definition of Culture	39
Defining Your Culture.....	40
Creating a Style That Is Unique.....	41
The Icon.....	41
Logo.....	44
User Interface	45
Your Application’s Web Site.....	47
Setting Up Your Web Site	49
Summary	50
■ Chapter 4: Marketing Your Android Application	51
How Did You Hear About Your Android Applications?.....	51
Word of Mouth: Your Best and Worst Marketing Strategy	53

Creating a Marketing Plan for Your Android Application	54
Who Are You Going to Tell About Your Android Application?.....	54
Blogs.....	55
Printed Journals	56
Social Networking	57
Other Methods of Spreading the Word for Your Application	60
SEO Tactics.....	61
Public Relations.....	61
Sales Goals	61
Scheduling Your Launch	62
Summary	62
■ Chapter 5: How to Promote Android Applications Within Android Applications...	63
Setting Up Your Application for Reviews	63
Timing Is Everything	67
Sharing an Application on a Social Network.....	68
Cross-Promotion: One Application Sells Another.....	70
Social Gaming Platforms	71
Summary	72
■ Chapter 6: Marketing the Free Application.....	73
Developing a Lite Version of Your Application	74
Free For Now	74
Differences Between Free and Paid Applications.....	75
When to Ask Your Users for an Upgrade.....	79
Advertising on Your Free Application.....	80
AdMob	81
Mobclix.....	87

Affiliate Programs	88
Summary	90
■ Chapter 7: In-App Billing: Putting a Store in Your Application	91
When to Use In-App Billing	91
Types of In-App Billing.....	93
In-App Billing for Android	94
Examples of In-App Billing	95
Setting Up Your Application for In-App Billing	97
How to Install the Sample.....	98
Installing In-App Billing Without the Demonstration.....	100
Summary	104
■ Chapter 8: The Prerelease Guide	105
Getting the Word Out on the Release Date on Your Official Web Site.....	105
Web Site Presence.....	105
Increasing Your Blogging About Your Application	107
RSS Feeds and Followers	108
Spreading the Word Through Social Media	109
Twitter	109
Facebook	110
Making a Video	110
Writing a Press Release.....	111
Format for a Press Release	112
Example of a Press Release	112
Employing Other Creative Marketing Strategies to Reach Your Audience.....	114
Summary	114

■ Chapter 9: Publishing to the Android Market	115
Is This Your First Android Application?	115
What to Do Before You Submit to the Android Market.....	116
How to Obtain a Screenshot of Your Android Device.....	117
Selecting the Proper Screenshot.....	120
How to Create a Distributable File	121
Creating an APK File Using Eclipse.....	122
Creating an Account on the Android Market.....	127
Uploading Your Application.....	135
The App Is Published, Now What?	142
Summary	142
■ Chapter 10: Best Launch Day Ever.....	145
Deciding Whether to Have a Beta-Testing Phase	145
You Have a Million Calls to Make.....	147
Sending Out Press Releases.....	148
Giving Your Media People a Complimentary Copy of Your Application.....	149
Setting Up a Press Room on Your Web Site.....	150
Using Third-Party Press Release Distribution Services	152
Get Your Paid Search On, If You Dare!	152
Getting the Word Out on Your Social Networks	154
The Season for Your Application.....	155
Considering a Temporary Giveaway	155
Getting Your Application a QR Code.....	156
Breaking Out of the Box.....	160
Summary	160

■ Chapter 11: Keep the Application Fires Burning	161
How Will You Change?.....	161
Following New Technology.....	161
Figuring Out the Season for Your Application.....	162
How to Update Your Android Application.....	164
Using Google Statistics.....	176
Attending Conferences.....	177
The Price.....	178
When to Do a Price Increase.....	178
When to Do a Price Decrease.....	178
Summary.....	179
■ Chapter 12: Programming in Android with the Eclipse IDE	181
Setting Up the Eclipse IDE to Work with the Android SDK.....	181
Starting a New Android Project.....	183
What You Will Notice About Your Android Project Structure.....	186
Uploading the Icon and Background for Your Android Application.....	188
How to Replace the Default icon.png File.....	188
How to Set a Background.....	191
Understanding Java SE.....	192
Objects.....	193
Methods.....	194
Inheritance.....	195
Interfaces.....	195
The Package Declaration.....	195
Getting Your Program to Do What You Want.....	196
The Android Developers Web Site.....	196
Stack Overflow.....	197

Other Ways to Find Programming Answers.....	198
Handling Errors in Eclipse.....	198
Running Applications on Eclipse.....	199
Creating an Emulator.....	199
Running Your Application on Your Emulator.....	202
Example of an Android Application.....	203
Summary	219
■ Index	227

About the Author

■ **Mark Rollins** was born in Seattle in 1971, and graduated Washington State University in 1994 with a degree in English. After college, he began to write skits for college-age groups. After four years working for Walmart, and another five years working for Schweitzer Engineering Laboratories (SEL), he decided to pursue a full-time career in writing. Since then he has written for many tech and gadget blogs, including screenhead.com, image-acquire.com, cybertheater.com, mobilewhack.com, carbuyersnotebook.com, gearlive.com, zmogo.com, gadgetell.com, gadgets-weblog.com, and coolest-gadgets.com. He has also written for video game blogs such as gamertell.com and digitalbattle.com. He recently began writing for the Android blog androidedge.com. Mark currently resides in Pullman, Washington with his wife and three children.

About the Technical Reviewer

■ **Bradley D. Brown** is a Senior Managing Partner and Chief Technology Officer of Breosla LLC, a business acceleration company he co-founded with Kevin Fallon. Breosla (www.breosla.com) leverages private equity and business process design, strategy, and execution for superior value creation. Breosla invests in software-related business model innovation opportunities where it can change or improve how a business creates, delivers, and retains value. An investing and operating company, it works primarily on cloud/SaaS, collaboration, enterprise, security, and mobile opportunities. Within Breosla are three areas of competency: Breosla Consulting that engages with established and expansion stage companies to accelerate business performance; Breosla Advisors, a licensed investment banking practice that helps portfolio companies with investment strategy and merger/acquisition (M&A) activity; and Breosla Partners, recently formed to launch and manage Breosla Partners I LP, a private equity fund.

Prior to Breosla, Brad founded and is a board advisor of TUSC, an international Oracle software consultancy. Vast experience earned Brad roles as acting CTO and board member for many companies over 23+ years at TUSC. Brad has published many applications in the Android markets. He has been a startup CTO and actively involved in IT since the early 1980s. He worked for Pioneer, Midwest Stock Exchange, and Oracle. Brad has written five technical books on Web Development in the Oracle world. He is the author of several best-selling Oracle Press books, and Oracle awarded him the honorary title of Oracle ACE Director. Brad taught “New Venture Creation” at the University of Denver’s Daniels College of Business. In 2009, his alma mater, Illinois State University put Brad into their first Hall of Fame for Applied Science and Technology. Brad holds a BS in Applied Computer Science from Illinois State University.

Brad may be reached at brad.brown@breosla.com and followed on bradleydbrown.blogspot.com.

Acknowledgments

I would like to give special acknowledgement to the following people:

Brian Dorgan, who gave me terrific advice for programming on Android. I wouldn't have been able to write half the things in this book if not for his advice.

Wallace Jackson, who wrote the Apress book *Android Apps for Absolute Beginners* and inspired me to write Android applications of my own.

Steve Anglin, whom I first met at CES 2011 and who helped grant me the opportunity to write for Apress.

Corbin Collins and his editorial team, as their good advice helped me to create this work. It has been a busy few months, but I am proud to say it is complete.

Geoff Webb, who helped me while I took my first programming course at Washington State University.

Kristina Rollins, my beautiful wife and the mother of my three wonderful children.

Al Carlton, who gave me a terrific opportunity for writing for coolest-gadgets.com for many years.

Preface

It's probably easier to explain what this book isn't, rather than go into detail about what it is. If you are looking for a book that will teach you exactly how to write Android programs, this is not that book.

This is not to say that I won't be discussing how to construct an Android app, but I just won't go into great detail. My purpose is to make certain that you, the successful Android developer, are pointed in the right direction. If help in programming Android apps is the type of help you need, then you might want to check out some other books from Apress, such as the following:

- *Android Apps for Absolute Beginners*, by Wallace Jackson (Apress 2011)
- *Beginning Android 3*, by Mark Murphy (Apress 2011)
- *Pro Android 3*, by Satya Komatineni, Sayed Hashimi, and Dave MacLean (Apress 2011)
- *Pro Android Web Apps*, by Damon Oehlman and Sébastien Blanc (Apress 2011)
- *Android Essentials*, by Chris Haseman (Apress, 2008)
- *Learn Java for Android Development*, by Jeff Friesen (Apress 2010) (note that this book is only about the subset of Java you need to program on Android; it says nothing about Android itself)

It is of course essential that the Android developer learn as much about programming as possible to pursue a career in original application development. However, what is almost as important is what to do once that incredible mobile program has been created. That's where this book comes in. Proper marketing, promotion, and advertising could be the difference between major and minor profits for your application. If you are in the business to make money from your Android mobile software, then the information contained within these pages is going to make a significant difference.

Much of the focus of this book is geared toward the marketing of an Android application, but I touch on specific aspects of programming and app creation.

Chapters 1 through 3 discuss how to conceive the idea of a well-marketed application, and what tools you will need to build this application. Chapters 4 through 9 are all about what marketing steps you need to take as you are building the application, in anticipation for the launch date. Chapters 10 and 11 discuss what to do after the launch date and how to keep momentum going on your application so it will continue to make you money for years to come. Chapter 12 talks about how to program with Android and the Eclipse IDE. The following lists details what's covered in each chapter.

- *Chapter 1:* This chapter introduces Android in general and talks about its origins and development as a platform.
- *Chapter 2:* As you well know, the Android Market is flooded with applications, so you need to find out what sets yours apart. Chapter 2 discusses what you can do to produce a work that will be in demand, how to analyze the competition, where to find your target audience, and even how to name your application. Chapter 2 also discusses what you need to do to protect the intellectual property of your application.
- *Chapter 3:* Chapter 3 discusses how to narrow your target audience by introducing culture into everything your application does. This includes making the user interface, the icon, the logo, and the web site follow a specific theme.
- *Chapter 4:* Once you have decided on a look and theme for your application, it's time to create a marketing plan for when it is released. You should determine your marketing strategy based on what media outlets you want to report on your Android application. This includes blogs, printed media, social networks, and other methods of spreading the word about your application, including word-of-mouth (the best *and* worst marketing method).
- *Chapter 5:* As a developer, you might want to create several applications. Chapter 5 shows how you can use one app to promote other apps you have developed. It also shows how to promote applications on social networks like Twitter and Facebook, as well as how to prompt users to review your applications, which will lead to more downloads.
- *Chapter 6:* Just because you are giving away an app for free on the Android Market doesn't mean that you can't get something out of it. This chapter will discuss creating a lite version of your app along with a paid version. Chapter 6 also discusses ad revenue, as well as affiliate programs to maximize profits.
- *Chapter 7:* The initial price of your app isn't the only way to make money from an application; there are several other methods, including in-app billing. This is when the application sets up a store for selling virtual goods, such as rewards within a game. This chapter shows how to put it in your program.
- *Chapter 8:* At this point, you should be just about ready to publish the application to the Android Market. In order to avoid work after your prerelease date, you should prepare a press release and get the word out on social networks and your application's website.
- *Chapter 9:* The Android Market is a lot simpler to get into than Apple's App Store, because there is no approval process. However, there *are* a lot of hoops to jump through before you can get your application on the Android Market. This chapter discusses the process, including digital signing, keystores, and creating an Android Market account.

- *Chapter 10:* Once your application is out on the market, it is time to let everyone know. Chapter 10 shows the easiest way to send out multiple e-mails, as well as how to inform potential users via social networks and other methods of promotion. The more people that know about your application, the better chance it has of selling.
- *Chapter 11:* Now that the app is out there, you can take other steps to ensure its success. Chapter 11 shows you steps that need to be taken in order to ensure that profits increase and the application lives on.
- *Chapter 12:* This chapter is for those who are using Eclipse IDE for the first time, and teaches basic Java SE programming commands. Also included is a sample application so you can see these tools in action.

The Android Application Business

The mobile application business model is essentially the new get-rich-quick scheme. There are many who put out their app on smartphone platforms, and begin to start receiving profits from users, advertisers, and shops within their applications. There is, of course, a lot of work involved before the developer can simply sit back and watch the money roll in.

This was certainly the case for iOS during the big app boom of the iPhone launch of 2007, and there is no question that the Android Market has been influenced by Apple's App Store. I will talk about their specific similarities and differences later, but for now, I want to discuss the Android Market in general.

The State of the Android Market

Android, with its services on phones and tablets, is a pretty good market to be in right now. In May 2011, Google announced that there had been 3 billion applications downloaded from the Android Market. Just a few months prior to that, there were approximately 30,000 free and paid applications available to Android users. This number grew to 30,000 from 16,000 in exactly three months. By the time that this book is published, I am sure that the Android Market will have over 100,000 applications, and I wouldn't be surprised if that growth is even faster than anticipated.

Dividing the March figure of 3 billion applications downloaded by the figure of 30,000 available applications, it means that every Android application, on average, has had 100,000 downloads.

Sure, that last number looks pretty good, but it is just a statistic. There is no guarantee that your brand-new Android application will have that number. Also keep in mind that a lot of these applications are free, which is helpful in achieving the greater-than-average number of downloads. I will detail in this book later how to have a free application and still make profits, but you should know that there is a paid download for every 100 downloads, and some say that figure is as high as 1,000 to 1.

As an example, *Angry Birds*, by Rovio, is free for Android users, and it is a mobile game giant that has produced massive funds for the company. The mobile game had over 2 million Android downloads in the three days after its Android release, and 7 million Android downloads one month after that. These stats are over a year old, and Rovio, the game's developers, are still finding ways to make money on the *Angry Birds* franchise with spin-off applications, merchandise, and more.

You will soon discover that the massive amount of Android applications on the market can work against the developer, as the Android Market is flooded with applications of all types. This means that one application, as great as it might be, can get lost in the crowd and become very difficult to be noticed by its intended audience. If a developer markets their application so it gets a lot of publicity, Android users might pay to download one type of application, even though a similar version with more features is readily available for free.

Go to the Android Market on a web browser. You will see a slideshow that you can browse through on your Android's device touchscreen or your browser (see Figure 1-1). You will also see a list of featured apps. On the web version, you'll see the following lists of apps as well: Top Paid, Top Free, Top Grossing,

Top New Paid, and Top New Free. Yes, you can probably make more from your app if you are on these lists, as your app will be easier to find by random Android users. However, there are ways of making money from your app even if you are not on anyone's Featured list.



Figure 1-1. A quick peek at the Android Market, from a web browser. Note the Featured section, the Top Paid list, and the slideshow of application ads.

Before we can start talking about how to make money with Android, let's talk about some Android basics.

The Origins of Android

Android Inc. was a small startup company that developed its own open source operating system for mobile phones. It was bought by Google in August, 2005.

In November 2007, shortly after Apple introduced the iPhone, Google began working on its contribution to the mobile phone business. It announced the start of the Open Handset Alliance (OHA), which is a consortium of companies such as HTC, Motorola, Samsung, Sprint, T-Mobile, and other big names in the telecommunications industry. On the very same day, OHA unveiled the mobile operating system that we know today as Android.

Android got off to a slow start with the HTC Dream (also known as the T-Mobile G1) in October 2008. Since then, Android has been gaining popularity with the release of every new version. It is important that you know about them when you begin developing Android applications, as newer versions contain more features than their predecessors. In terms of programming, these versions have a definite numerical designation. I will discuss that when we get into downloading Android development tools like the Android SDK (Software Development Kit) and Eclipse.

For now, you should know that in addition to a specific number, Android versions also have an informal name, which is always named after some sweet treat. This is partly due to tradition and Google's sense of humor. This naming convention is also alphabetical, having begun with Cupcake, followed by Donut, Éclair, Froyo, Gingerbread, Honeycomb, and most recently Ice Cream Sandwich. Here is a very basic summary of the features of the latest versions of Android:

- Version 1.5 (Cupcake):
 - Ability for videos to be recorded through camcorder
 - Bluetooth enabling
 - Widgets on home screen
 - Animated screen capabilities
 - Uploading of videos to YouTube and pictures to Picasa on the fly
- Version 1.6 (Donut):
 - Camcorder, camera, and integrated gallery
 - Voice search
 - Voice dial
 - Bookmark
 - History
 - Contacts search
 - WVGA screen resolution
 - Google turn-by-turn navigation
- Version 2.0/2.1 (Éclair):
 - HTML5 and Exchange Active Sync 2.5 support
 - Improved speed
 - Google Maps 3.1.2
 - MS Exchange Server integration
 - Flash for the camera
 - Bluetooth 2.1 integration
 - Option of virtual keyboard
- Version 2.2 (Froyo):
 - 320dpi screen with 720p
 - JIT compiler
 - Chrome with JS engine version 8

- Wi-Fi hotspot tethering
- Bluetooth contacts sharing
- Adobe Flash support for version 10.1
- Ability for apps to be installed on expandable memory, such as SD cards
- Version 2.3 (Gingerbread):
 - Improved gaming graphics and audio effects
 - SIP VoIP support
 - WXGA (extra-large screen size and resolution)
 - Near field communication
 - Copy/paste feature
 - Download manager for large downloads
 - Better control of applications
 - Support for multiple cameras
- Version 3.0 (Honeycomb):
 - The first tablet-only release
 - 3D desktop with newer widgets
 - Tabbed web browsing
 - Gtalk video chat
 - Hardware acceleration
 - Multicore processor
 - Multipane navigation
- Version X (Ice Cream Sandwich):
 - Streamlined user interface fit for both tablets and smartphones
 - Advanced app framework

You will discover that specific Android devices start out as one particular version of Android, and upgrades tend to come out slowly. It depends on the carriers, and often subscribers don't hear when the new updates come out. For example, I updated my Droid X from Froyo to Gingerbread, but Verizon didn't inform that an update was readily available. I only heard about the update because of my work in tech blogging, and someone else figured out how to give me the update. If you are curious about discovering whether you have the most current version on your Android device, click its Settings icon, select About Phone, and then select System Updates.

You will also discover that knowing about these versions of Android will be helpful when programming new applications. Later versions of Android devices are reverse (backward) compatible, which means that an Android 2.2 device will play an Android 1.6 program. However, Android apps

designed for latter versions may not play on devices on former versions. For example, a version 1.6 Android device will not be able to use an app programmed for Android 2.2. In fact, if you have a device that runs 1.6 and do a search on the Android Market for an application designed for versions 2.2 or above, it will not even show up in the results.

By the way, a lot of low-priced Android phones still run very early versions of Android; for this reason, it might be wise to create a program that can run on Android 1.5 or 1.6 rather than only version 2.0 or above. Unless you have specific tablet-based application in mind, do not program with Android 3.0. Version 3.0 is optimized for tablets, not for smartphones. Right now, the Honeycomb market is limited to a few models, and until more models are made (which could be soon) it could take a while before a developer can make money off of it.

Android vs. iOS

When the iPhone was first unveiled, a new sort of business model was established for consumer electronics. While Steve Jobs and his friends at Apple were not the first to invent the touchscreen, they were able to create a new type of software enterprise that was personable and utilitarian. Apple's "there's an app for that" slogan has proven to users that the mobile software that they need should be readily available where and when they need it. It works for the smartest engineer and the dumbest consumer, and it has created a new type of software market.

As of this writing, Apple is a monster when it comes to apps. In fact, there is a little bit of a legal suit on who has the rights to the word *app*, and I'm guessing that it won't be settled by the time this has been published. Android may not currently have the high numbers that Apple has, but its market is ever-growing.

In February 2011, Google CEO Eric Schmidt stated that Android activations reached about 350,000 per day. At a recent I/O developer conference last May, it was revealed that there were 400,000 new activations per day, not to mention 450,000 developers and 215 carriers. That is about 100 million total activations overall as of May 2011. In comparison, Steve Jobs stated in March 2011, during his presentation of the iPad 2, that over 100 million iPhones have been activated. It would appear that Android's 100 million devices in May 2011 means that Android is catching up quickly. In short, the Android market is presently booming, and if any software has a chance to eclipse Apple's huge dominance over the app industry, it's Android.

When Steve Jobs announced the iPad 2, he ribbed that Honeycomb only had 100 applications while the iPad had hundreds of thousands. That figure was slightly biased, as Honeycomb had been around for only one month. Also, many Android smartphone applications work on Honeycomb.

As a developer, you should know how Android compares to iOS at least at some level. Apple iOS is a proprietary operating system, while Android is open source, which means it gives users the right to use, study, change, and improve design through ready availability of the source code. It uses the Linux kernel software for all of its versions. This is different from iOS, which works only on Mac products. This is one of the reasons why the Android operating system is so popular on smartphones and tablets, and will probably have a significant presence in televisions in the near future: it is not required to be tied to any particular device manufacturer.

Another way that Android differs from Apple is that Android has no approval process when it comes to apps. Once the user has signed up, uploading and publishing becomes a relatively simple process. (This is different for when you submit to Amazon's Android market, and we will discuss Amazon's approval process in later chapters.)

Also, you should know that Google reserves the right to yank your application off Android Market, even after you are already on it. A lot of applications are removed due to NSFW (not-suitable-for-work) content, but they can easily be yanked for other reasons as well.

Porting Difficulties

For those who want to turn their iOS app into a full-fledged Android application (or vice versa), I want to let you know the process and pitfalls.

If you haven't written an app for iOS, you might want to skip this section if you are thinking that you just want to market your application for the Android Market. However, I'm going under the assumption that you are trying to monetize as much as possible, so you might consider turning your Android application into an iOS application. I will cover how to do that by the end of this chapter.

Let's say you've written an iOS app that is out right now, or perhaps waiting in the app approval process. To turn it into an Android application, you will have to adapt your software so that an executable program can be created for a computing environment that is different from the one that it was originally designed for. This is known as porting.

Most of the iOS apps on the iTunes Store are written in Objective-C, while Android is specifically Java and XML. Although the logic of these programming languages is quite similar, as they are both linear, procedural, and use notions of object orientation (OO), they are very different with respect to OS support, GUI objects, and application life cycle. Sadly, Objective-C is not supported on Android.

As far as I can tell, there doesn't seem to be some magical program that will allow you to insert iPhone apps in and get Android applications out (unless you use development tools that have this in mind from the beginning). I will take a moment to talk about cross-platform development tools later, but let me discuss what you can do for both Android and iOS.

Your iOS coding cannot be reused for Android coding, but it does not mean that you have to rewrite all your Android code from scratch. For example, you can completely reuse the icons and images, as well as any SQLite database code.

Generally, it usually takes the same amount of effort to port an iOS app to Android as it does to create it. Hopefully it will take less time, but I can't guarantee that. It really depends on how big the application is, as well as the complexity of the code, reliance on GUI tools, and the ability of the developer.

By the way, there are people who make it their business to program apps, and that could lighten your workload. The same case exists for those who are into porting iOS apps to Android applications.

If you have written your iOS app in ANSI C or C++, such as in the many gaming engines that come out for such a purpose, then you are in luck. Android has the Native Development Kit (NDK), which allows for ANSI C or C++ code to be used with a Java layer for interface. If you go with the NDK, you will not have to use the Android SDK, which I will describe later.

Examples of Cross-Platform Development Tools

Chances are you want your application to be downloaded as much as possible, which means that you probably want it to be on many devices as possible. If you want to have your application on both iOS and Android, as well as other mobile platforms, you are going to have to do this from the beginning with certain cross-platform development kits.

Allow me to explain that I cannot testify to the success of such programs as these. I believe that the application market is headed in the direction where one developer's toolkit will work for all platforms, but, as explained before, we are not quite there yet. For the sake of the subject matter of this book, I only discuss how to use the Java JDK (Java Development Kit), the Android SDK, and the Eclipse IDE (integrated development environment) to construct Android applications. I will explain these programs later, but I wanted to discuss a few cross-platforms development kits, just so you know that there are alternatives.

Livecode

Livecode is the work of RunRev, a company that creates development tools. In the words of Ben Beaumont, product manager for RunRev, Livecode is “a multi-platform element environment that [has] now been moved to the mobile space.” Livecode was originally made for Mac, Windows, and Linux, and it boasts “compile-free coding.” Compile-free coding means that when you make a change to your program, you will see it as you are programming. This is different than the usual method of editing, compiling, running, and debugging.

Livecode also has a visual development environment, where the user can drag and drop the objects and images that will make up the final interface. The user can then attach scripts to these objects to really bring them to life, as well as lend them speed. Livecode uses a very high-level language, which allows the user to write in code that is very close to English. This allows you to write in this code easily, and the code will be easier to read. All this allows for the creation of live prototypes that actually run on the device, and promises to make it easy to work with clients, as you can more directly show them what you want, rather than showing them stagnant code.

Appcelerator

Titanium has created a free and open source application development platform that allows the user to create native mobile, tablet, and desktop application experiences. Their Appcelerator program allows the user to build applications full of features, as if they were written in Objective-C or Java. The end results are native apps that are customizable with a lot of features, all built with the web technology of JavaScript.

They allow developers to concentrate on building the application and provide a toolset for many platforms.

appMobi XDK

appMobi’s mobile app development XDK is made for web developers. appMobi’s claim is that if you can build an application for the web using HTML, CSS, and JavaScript, then you can build it as an application on the iPhone, the iPad, and Android smartphones and pads. According to their appMobi, developers can develop robust, 100 percent native API-compliant mobile applications in hours using preferred editors, and write once to deploy for all target platforms.

XDK includes an onscreen emulator with simple, approachable tool palettes to simulate user interaction with a testing device. It also allows you to send your application project over a local Wi-Fi connection or upload to the cloud to test it from anywhere. As a warning, you will need Java 6 and Google Chrome 6.0 to even begin running this program.

appMobi also offers a service known as MobiUs, which allows any app publisher to offer its app from anywhere on the Web, which could mean the end of frustrating and complicated processes of submission and approval to traditional app distributors (and also the end of developers needing to share their profits with these distributors). It is also cloud-based, which means it is possible to create iPhone apps on a Windows PC and Android applications on a Mac.

PhoneGap

According to its web site, PhoneGap allows users to build apps with web standards based on HTML 5. PhoneGap users can also access native APIs in order to create applications for multiple platforms, including iOS, Android, Windows, BlackBerry, webOS, and more. PhoneGap is currently in version 1.0.0.