Get started building your very own Android apps

Android Apps for Absolute Beginners

Wallace Jackson

Apress®

Android Apps for Absolute Beginners

Wallace Jackson

Apress®

Android Apps For Absolute Beginners

ISBN 978-1-4302-3446-3

ISBN 978-1-4302-3447-0 (eBook)

Printed and bound in the United States of America (POD)

President and Publisher: Paul Manning
Lead Editor: Matthew Moodie
Technical Reviewer: Kunal Mittal
Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jonathan Hassell, Michelle Lowman, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Frank Pohlmann, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Coordinating Editor: Corbin Collins
Copy Editors: Marilyn Smith, Sharon Terdeman, Tracy Brown
Compositor: MacPS, LLC
Indexer: BIM Indexing & Proofreading Services
Artist: April Milne
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/info/bulksales.

The source code for this book is available to readers at www.apress.com. You will need to answer questions pertaining to this book in order to successfully download the code.

In loving memory of all of our wonderful furry companions who graced our lives with unconditional love for decades here on our ranch in the La Purisima State Historic Park on the Point Conception Peninsula in Northern Santa Barbara County.

Contents at a Glance

Contents

About the Author

Wallace Jackson is the CEO of Mind Taffy Design, a new media content design and production company founded in 1991. Mind Taffy specializes in leveraging free for commercial use open source technologies to provide an extremely compact data footprint, royalty-free, digital new media advertising and branding campaigns for the leading international brands and manufacturers worldwide.

Wallace has been pushing the cutting edge of i3D and Rich Media Application Design via viral digital content deliverables, using under 512KB of Total Data Footprint, for over two decades. He has worked for leading international brands to create custom new media digital campaigns for industry-leading companies, including brand marketing, PR, product demonstration, digital signage, e-learning, AdverGaming, logo design, and end-user training for top Fortune 500 companies.

He has produced new media projects in a number of digital media "verticals" or content deliverable areas, including: interactive 3D [i3D], Rich Internet Applications (RIA) content production, virtual world design, user interface (UI) design, user experience (UX) design, multimedia production, 3D modeling, sound design, MIDI synthesis, music composition, image compositing, 3D animation, game programming, mobile application programming, BrandGame creation, website design, CSS programming, data optimization, digital imaging, digital painting, digital video editing, special effects, morphing, vector illustration, IPTV Programming, iTV application design, interactive product demos, and tradeshow multimedia.

Wallace has created new media digital campaigns for leading international branded manufacturers, including Sony, Samsung, Tyco, Dell, Epson, IBM, Mitsubishi, Compaq, TEAC, KDS USA, CTX International, ADI Systems, Nokia, Micron, ViewSonic, OptiQuest, SGI, Western Digital, Sun Microsystems, ProView, Sceptre, KFC, ICM, EIZO, Nanao, Digital Equipment [DEC], TechMedia, Pacific Digital, ArtMedia, Maxcall, Altrasonic, DynaScan, EZC, Smile, Kinoton GMBH, and many others.

Wallace holds an MSBA post-graduate degree in Marketing Strategy from USC, an MBA degree in Management Information Systems Design and Implementation from the USC Marshall School of Business, and a Bachelor's degree in Business Economics from UCLA Anderson School of Management. He is currently the #2 ranked All Time Top Expert on LinkedIn, out of more than 90,000,000 executives that use that social media business web site.

About the Technical Reviewer

Kunal Mittal serves as an Executive Director of Technology at Sony Pictures Entertainment, where he is responsible for the SOA, Identity Management, and Content Management programs. He provides a centralized engineering service to different lines of business and leads efforts to introduce new platforms and technologies into the Sony Pictures Enterprise IT environment.

Kunal is an entrepreneur who helps startups define their technology strategy, product roadmap, and development plans. Having strong relations with several development partners worldwide, he is able to help startups and even large companies build appropriate development partnerships. He generally works in an advisor or consulting CTO capacity, and serves actively in the project management and technical architect functions. He has authored and edited several books and articles on J2EE, cloud computing, and mobile technologies. He holds a Master's degree in Software Engineering and is an instrument-rated private pilot.

Acknowledgments

My sincere thanks go to:

Matthew Moodie, my lead editor, for his patience and thoughtful guidance in shaping this first edition of *Android Apps for Absolute Beginners*. Matthew, thanks for guiding me as a new Apress author, and I look forward to future collaborations with you.

Kunal Mittal, my esteemed technical reviewer, for his hard work and insightful suggestions in shaping this edition of the book.

Steve Anglin, my acquisitions editor, for bringing me into the Apress family to write this book. I wouldn't have done it at all if it were not for you!

Dominic Shakeshaft, editorial director, for overseeing the editorial process while I wrote. I appreciate your help with the higher-level issues involved.

Corbin Collins, my coordinating editor, for listening to all of my miscellaneous and sundry problems during the writing of this book and helping to get them all sorted out.

Marilyn Smith, **Sharon Terdeman**, and **Tracy Brown**, my copy editors, for their excellent editing and book-polishing skills and for all the great suggestions for making this a fantastic Android book.

My Editorial Board, including Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jonathan Hassell, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Frank Pohlmann, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, and Tom Welsh, for making sure this is the best book for beginners about the esteemed open source Android operating system.

The many loved ones and clients who patiently awaited my return to i3D content production from the "professional sidetracker" commonly known as writing a programming book.

Introduction

Over the last two years, Google's Android operating system (OS) has gone from a virtually unknown open source solution to the current mobile OS market leader among all mobile handsets, with over one-third of the market share, and it's still climbing rapidly. Android has even started to dominate the tablet OS marketplace, and is also the foundation for the popular iTV OS known as GoogleTV. There seems to be no end in sight for Android's rocketing success, which is great news for owners of this book.

I've heard a great many people say, "I have a really phenomenal idea for a smartphone application! Can you program it for me!?" Rather than sit back and code all of these applications for everyone, I thought it might be a smarter idea to write a book about how an absolute beginner could code an Android application using open source tools that cost nothing to download and that are free for commercial use, and then leverage that new found knowledge to reach their dream of making their application idea a revenue-generating reality.

Thanks to open source and Google's Android development environment, Oracle's Java programming Language, Linus Torvald's Linux operating system, the Eclipse code editing software, and this book, vaporizing a software product out of thin air, and at no production cost other than your PC and "sweat equity," is now a complete reality.

The Target: The Programming Neophyte

As you may have inferred from the title, this book assumes that you have never programmed before in any programming language. It is written for someone who has never written a single line of code before, and who is thus unfamiliar with object-oriented programming (OOP) languages such as Oracle's Java and mark-up languages such as XML. Both of these open source languages are used extensively in creating Android applications.

There are lots of Java and Android books out there, but all of these books assume you have programmed before, and know all about OOP. I wanted to write a book that takes readers from knowing absolutely nothing about programming or knowing how to install a Software Development Kit (SDK) and Integrated Development Environment (IDE) all the way to being able to program Android applications using Java and XML.

The Weapon: Android, the Innovative Mobile Code Environment

Android is my Internet 2.0 development weapon of choice, because it allows me to develop highly advanced applications for the primary Internet 2.0 devices, including the main three where revenue potential is by far the greatest:

- Smartphones
- Tablets
- iTV or Interactive Television

The other reason I place my bets on Android is because it is open source, and thus free from royalties and politics. I do not have to submit my Android application to any company and ask permission to publish it, as long as it is not harmful in any way to others. For this reason, and due to the free for commercial use nature of open source software, there is little external risk involved in developing an application for the Android Platform.

How This Book Is Organized

Because this is a book for absolute beginners, we start at the very beginning, showing where to download and how to install the various Android, Java, and Eclipse environments, as well as how to configure these environments and how to set them up for application development and testing. This in itself is no easy task, and must be done correctly, as these tools provide the foundation for all of our Android development, debugging, and testing for the remainder of the book.

Next I will provide you with an overview of where Android came from, why, how, and when Google acquired it, and how it is uniquely structured among software development platforms. I will introduce XML, Java, OOP, and Android concepts soon after that, as well as cover how Android manages its screen layout. We will then move these concepts into use in later chapters in the second half of the book; these chapters explain the most important concepts in Android in their most logical order as they pertain to applications development.

In that second half of the book, we'll start getting into developing a user interface (UI), as that is the front-end or interface for your user to your Android application. Soon after we'll cover how your UI talks to your application via events processing. To spice up your application's visual appearance, we'll get into graphics, animation, and video, and then get into even more advanced topics after that, such as databases and communications.

Finally we will look at some of the advanced features of Android that you will want to visit after finishing the book; these are topics that are too advanced for a first book on Android but which provide some of the coolest features in smartphone development today.

We'll walk you through all of these topics and concepts with screenshots of the IDE and visual examples and then take you though step-by-step examples reinforcing these concepts. Sometimes we will repeat previous topics to reinforce what you have learned and apply these skills in new ways. This enables new programmers to re-apply development skills and feel a sense of accomplishment as they progress.

The Formula for Success

Learning to develop an Android application is an interactive process between you and the tools and technologies (Eclipse, XML, Java, Android, and so on) that I cover in this book. Just like learning to play a sport, you have to develop skills and practice them daily. You need to work through the examples and exercises in this book, more than once if necessary to become comfortable with each concept.

Just because you understand a concept that doesn't necessarily mean you will know how to apply it creatively and use it effectively; that takes practice, and ultimately will happen when the "ah-ha" moment occurs, when you understand the concept in context with the other concepts that interconnect with it.

You will learn quite a bit about how Android works from this introductory book. You will glean a lot of insight into the inner working of Android by working through all of the exercises in this book. But you will also learn new things not specifically mentioned in this book when you compile, run and debug your programs. Spending time experimenting with your code and trying

to find out why it is not working the way you want, or trying to add new features to it, is a learning process that is very valuable.

The downside of debugging is it can sometimes be quite frustrating to the new developer. If you have never wanted to put a bullet in your computer monitor, you will soon. You will question why you are doing this, and whether you are savvy enough to solve the problem. Programming can be very humbling, even for the most experienced of developers.

Like an athlete, the more you practice, the better you will become at your skill. You can do some truly amazing things as an Android programmer. The world is your oyster. It is one of the most satisfying accomplishments you can have, seeing your app in the Android App Store. However, there is a price, and that price is time spent practicing your coding.

Here is our formula for success:

- Trust that you can pull it off. You may be the only one who says you can't do this. Don't tell yourself that.
- Work through all the examples and exercises in this book, twice if necessary, until you understand them.
- Code, code some more, and keep coding – don't stop. The more you code, the better you'll get.
- Be patient with yourself. If you were fortunate enough to have been a star pupil who can memorize material simply by reading it, this will not happen with Java and XML coding. You are going to have to spend lots of time coding in order to understand what is happening inside the OS.
- Whatever you do: DON'T GIVE UP!

Required Software, Materials, and Equipment

One of the great things about Java, Android and Eclipse is they are available in both 32-bit and 64-bit versions on the three primary operating systems in use today:

- Windows
- Mac
- Linux

The other great thing about Java, Android and Eclipse is that they are free. You can download Android at `http://developer.android.com/SDK/`. For equipment, any modern computer will do. Fortunately they are only $250 to $500 brand new on `www.PriceWatch.com` and an OS such as SUSE Linux is free and an amazing development operating system. SUSE Linux V11 can be downloaded at `www.OpenSUSE.com` and is currently at version 11.4 and very stable.

Operating System and IDE

Although you can use Android on many platforms, the Eclipse integrated development environment (IDE) that developers use to develop Android apps is most commonly used on an Intel-based Windows or Linux PC. The Eclipse IDE is free and is available on the Internet at `www.eclipse.org`. The operating system should be Windows XP or later or SUSE Linux 11.4 or later to run Eclipse most effectively.

Software Development Kits

You will need to download the Eclipse IDE from Eclipse and the Android SDK from Google. This is available at `http://developer.android.com/SDK/`.

Dual Monitors

It is highly recommended that developers have a second monitor connected to their computer. It is great to step through your code and watch your output window and Android emulator at the same time on dual, independent monitors. Today's PC hardware makes this easy. Just plug your second monitor in to the second display port of any Intel-based PC or laptop, with the correct display port adapter, of course, and you're able to have two monitors working independently from one another. Note it is not required to have dual monitors. You will just have to organize your open windows to fit on your screen if you don't.

Chapter 1

Preliminary Information: Before We Get Started

This chapter introduces the Android operating system, giving you a little background information to put things into perspective. We'll visit just how expansive this platform has become in today's Internet 2.0 environment of portable consumer electronic devices. *Internet 2.0* here refers to the consumption of the Internet over a wide variety of different types of data networks using highly portable consumer electronic devices, including smartphones, tablets, e-book readers, and even new emerging consumer electronic products such as interactive television (iTV).

As this is an introductory book on the subject, not all of the advanced new media-related areas, such as 3D and video streaming, will be covered. Some specifics of what the book will and will not cover are outlined in this chapter.

At the end of the chapter, you'll learn which tools you need to obtain in order to develop for the Google Android platform, with instructions on how to download them.

Those of you who already recognize the significance of the Android revolution and know which tools are needed to develop Android applications development may want to skip this chapter. However, may be some tidbits in here that could spawn development ideas —so skip along at your own risk!

Just a bit of fair warning: developing reliable applications for Android is not in any way a trivial task. It takes a fair amount of knowledge of both high-level programming languages such as Java and markup languages like XML. Building useful and engaging new media applications also requires a deep knowledge of related new media technologies such as 2D imaging, 3D rendering, audio processing, video streaming, GPS localization, and database design.

Don't expect to learn all of this at one sitting. Becoming a top-notch Android programmer will take years of dedication and practice, as well as diligent research and trial and error. In this book, you will gain the foundation that you need to build future expertise, as well as learn the work process for eventually building your Android masterpeice.

Some History: What Is Android?

Android was originally created by Andy Rubin as an operating system for mobile phones, around the dawn of this twenty-first century. In 2005, Google acquired Android Inc., and made Andy Rubin the Director of Mobile Platforms for Google. Many think the acquisition was largely in response to the emergence of the Apple iPhone around that time; however, there were enough other large players, such as Nokia Symbian and Microsoft Windows Mobile, that it seemed like a salient business decision for Google to purchase the talent and intellectual property necessary to assert the company into this emerging space, which has become known as Internet 2.0.

Internet 2.0 allows users of consumer electronics to access content via widely varied data networks through highly portable consumer electronic devices, such as smartphones, touchscreen tablets, and e-books, and even through not so portable devices, such as iTVs, home media centers, and set-top boxes. This puts new media content such as games, 3D animation, digital video, digital audio, and high-definition imagery into our lives at every turn. Android is one of the vehicles that digital artists will leverage to develop media creations that users have never before experienced.

Over the past decade, Android has matured and evolved into an extremely reliable, bulletproof, embedded operating system platform, having gone from version 1.0 to stable versions at 1.5, 1.6, 2.0, 2.1, 2.2, 2.3, and, recently, 3.0. An embedded operating system is like having an entire computer on a chip small enough to fit into handheld consumer electronics, but powerful enough to run applications (commonly known as *apps*).

Android has the power of a full-blown computer operating system. It is based on the Linux open source platform and Oracle's (formerly Sun Microsystems's) Java, one of the world's most popular programming languages.

> **NOTE:** The term *open source* refers to software that has often been developed collaboratively by an open community of individuals, is freely available for commercial use, and comes with all of the source code so that it can be further modified if necessary. Android is open source, though Google develops it internally before releasing the source code; from that point on, it is freely available for commercial use.

It is not uncommon for an Android product to have a 1GHz processor and 1GB of fast, computer-grade DDR2 memory. This rivals desktop computers of just a few years ago and netbooks that are still currently available. You will see a further convergence of handheld operating systems and desktop operating systems as time goes on. Some examples are the Windows Mobile 7 and iPhone 4 mobile platforms.

Once it became evident that Android and open source were forces to be reckoned with, a number of major companies—including HTC, Samsung, LG Electronics, and T-Mobile—formed and joined the Open Handset Alliance (OHA). This was done in order to put some momentum behind Google's open source Android platform, and it worked.

Today, more brand manufacturers use Android as an operating system on their consumer electronic devices than any other operating system.

This development of the OHA is a major benefit to Android developers. Android allows developers to create their applications in a single environment, and support by the OHA lets developers deliver their content across dozens of major branded manufacturer's products, as well as across several different types of consumer electronic devices: smartphones, iTV sets, e-book readers, home media centers, set-top boxes, and touchscreen tablets. Exciting possibilities—to say the least.

So, Android is a seasoned operating system that has become one of the biggest players in computing today, and with Google behind it. Android uses freely available open source technologies such as Linux and Java, and standards such as XML, to provide a content and application delivery platform to developers as well as the world's largest consumer electronics manufacturers. Can you spell O-P-P-O-R-T-U-N-I-T-Y? I sure can ... it's spelled *ANDROID*.

Advantage Android: How Can Android Benefit Me?

There are simply too many benefits of the Android platform to ignore Android development.

First of all, Android is based on open source technology, which was at its inception not as refined as paid technologies from Apple and Microsoft. However, over the past two decades, open source software technology has become equally as sophisticated as conventional development technologies. This is evident in Internet 2.0, as the majority of the consumer electronics manufacturers have chosen Linux and Java over the Windows and Macintosh operating systems. Therefore, Android developers can develop not only for smartphones, but also for new and emerging consumer electronic devices that are network-compatible and thus available to connect to the Android Market. This translates into more sales onto more devices in more areas of the customer's life, and thus more incentive to develop for Android over closed and PC operating systems.

In addition to being free for commercial use, Android has one of the largest, wealthiest, and most innovative companies in modern-day computing behind it: Google. Add in the OHA, and you have more than a trillion dollars of megabrand companies behind you supporting your development efforts. It seems too good to be true, but it's a fact, if you are an Android developer (which you are about to be, in about a dozen chapters).

Finally, and most important, it's much easier to get your Android applications published than those for other platforms that are similar to Android (I won't mention any names here to protect the not so innocent). We've all heard the horror stories regarding major development companies waiting months, and sometimes years, for their apps to be approved for the app marketplace. These problems are nearly nonexistent on the open source Android platform. Publishing your app on Android Market is as easy as paying $25, uploading your *.apk* file, and specifying free or paid download.

The Scope of This Book

This book is an introduction to developing applications on Android. It's intended for absolute beginners—that is, people who have never created an application on the Android platform for a consumer electronic device. I do not assume that you know what Java is or how XML works.

What's Covered

This book covers the basic and essential elements of Android development, including the following:

- The open source tools required to develop for this platform
  - Where to get these free tools
  - How to properly install and configure the necessary tools for applications development
  - Which third-party tools are useful to use in conjunction with the Android development tools
  - Which operating systems and platforms currently support development for the Android using these tools
- The concepts and programming constructs for Java and XML, and their practical applications in creating Android applications
- How Android goes about setting up an Android application
  - How it defines the user interfaces
  - How it writes to the display screen
  - How it communicates with other Android applications
  - How it interfaces with data, resources, networks, and the Internet
  - How it alerts users to events that are taking place inside and outside the application
- How Android applications are published
- How Android applications are ultimately sold, downloaded, and updated automatically through the Android Market

Realize that Android has more than 44 Java packages that contain over 7,000 pieces of programming code functionality to allow you to do just about anything imaginable—from putting a button on the screen to synthesizing speech and accessing advanced smartphone features like the high-resolution camera, GPS, and accelerometer.

NOTE: A *package* in Java is a collection of programming utilities that all have related and interconnected functionality. For example, the `java.io` package contains utilities to deal with input and output to your program, such as reading the contents of a file or saving data to a file. Later chapters describe how to organize your own code into packages.

What does this mean? It means that even the most advanced Android books cannot cover the plethora of things that the Android platform can do. In fact, most books specialize in a specific area in the Android APIs. There is plenty of complexity in each API, which ultimately, from the developer's viewpoint, translates into incredible creative power.

What's Not Covered

So, what *isn't* covered in this book? What cool, powerful capabilities do you have to look forward to in that next level book on Android programming?

On the hardware side, we will not be looking at how to control the camera, access GPS data from the smartphone, and access the accelerometer and gyroscope that allow the user to turn the phone around and have the application react to phone positioning. We will not be delving into advanced touchscreen concepts such as gestures, or accessing other hardware such as the microphone, Bluetooth, and wireless connections.

On the software side, we will not be diving into creating your own Android MySqLite Database Structure, or its new media codecs for digital video and digital audio, and its real-time 3D rendering system (called OpenGL ES). We will not be exploring speech synthesis and recognition, or the universal language support that allows developers to create applications that display characters correctly in dozens of international languages and foreign character sets. We will not be getting into advanced programming such as game development, artificial intelligence, and physics simulations. All of these topics are better suited to books that focus on these complex and detailed topical areas.

Preparing for Liftoff: SDK Tools to Download

In Chapter 3, you'll learn how to set up a complete Android development environment. We'll focus on Windows, because that's what I use to develop for Android, but the process on Mac or Linux systems is similar, and I'll make sure you can follow along if you prefer either of those systems.

Here, we'll look at where to go to download the tools you'll need, so that you are ready for action when the time comes to install and configure them. This is because each of these development tools is hundreds of megabytes in file size, and depending on your connection speed, may take anywhere from ten minutes to ten hours to download.

There are three major components of an Android development environment:

- Java
- Eclipse
- Android

In Chapter 3, when you install and configure the packages you are downloading now, you will see that Eclipse requires the Java package to be installed in order to install and run. Therefore, we will walk through downloading them in the order of installation, from Java to Eclipse to Android.

Java

Let's start with the foundation for everything we are doing, the Java Platform, Standard Edition (Java SE). Java SE contains the core Java programming language.

To download Java SE, simply go to the Java SE Downloads section of Oracle's web site, which is in the Technology Network section under the Java directory, at this URL:

`http://www.oracle.com/technetwork/java/javase/downloads/index.html`

Figure 1–1 shows the Java SE Downloads site.

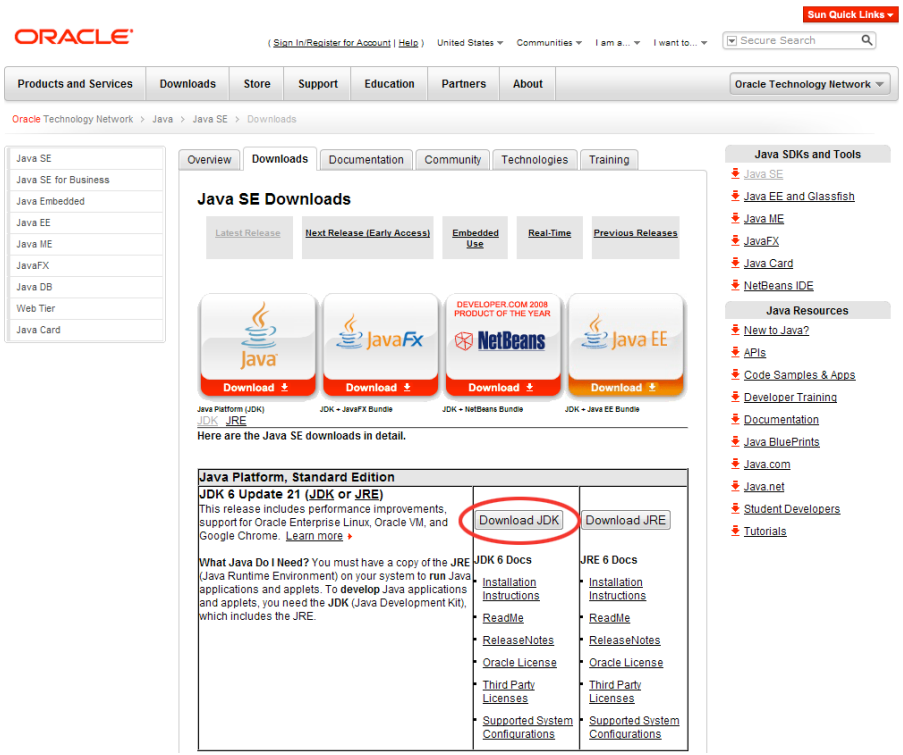

Figure 1–1. *Download the Java SE JDK.*

Click the Download JDK button to start downloading the Java SE Java Development Kit (JDK). Then choose your platform from the drop-down menu that appears, accept the license, and click the Continue button. You will be shown a link to the download that you selected. Click that link to start the download.

> **NOTE:** Make sure *not* to download Java Platform, Enterprise Edition (Java EE), JavaFX, or Java with NetBeans.

Eclipse

Eclipse is an *integrated development environment* (IDE), which is a piece of software dedicated to allowing you to more easily write programming code, and run and test that code in an integrated environment. In other words, you write all your code into its text editor, before running and testing that code using commands in Eclipse, without ever needing to switch to another program.

Currently, Android requires the Galileo version of Eclipse (*not* Helios). You should download the version of Eclipse that supports Java—Eclipse IDE for Java Developers. Go to the Eclipse web site's Downloads section at this URL:

`http://www.eclipse.org/downloads/packages/release/galileo/sr2`

Figure 1–2 shows the Galileo package you want to download.

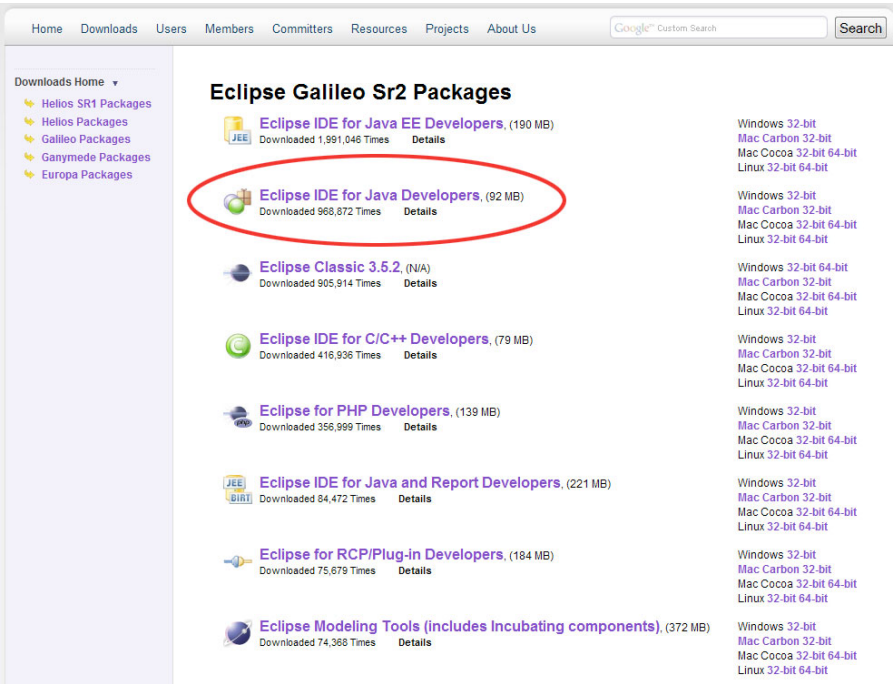

Figure 1–2. *Choose to download the Eclipse IDE for Java Developers.*

Click the link in the right-hand column that matches your system, and then choose the site from which to download.

Android SDK

The Android Software Development Kit (SDK) is a collection of files and utilities that work hand in hand with the Eclipse IDE to create an Android-specific development tool.

To dowload the Android SDK, go to the Android Developers web site, located at this URL:

`http://developer.android.com/sdk/index.html`

Figure 1–3 shows the Android SDK packages available. Download the latest SDK for the platform you are using.

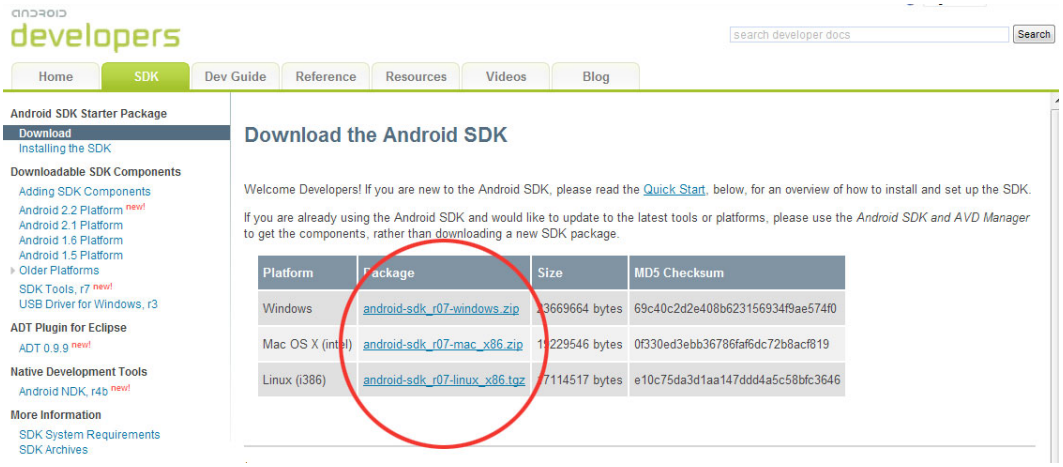

Figure 1–3. *Download the Android SDK.*

> **NOTE:** We will walk through installing the other minor packages (shown on the left side of Figure 1–3) using Eclipse in Chapter 3. For now, you don't need to worry about anything except downloading the main SDK.

Once the Eclipse and Android SDKs are installed and configured, you can further enhance them by installing phone emulators and other add-ins, which are covered in Chapter 3. In that chapter, we will go through the detailed setup of the Eclipse IDE for Android development.

Summary

Andy Rubin's creation called Android was purchased by Google in 2005 and made freely available to developers to create mobile device applications using Java and XML. Since

then, the Android phenomenon has grown to encompass an open industry alliance of the leading manufacturers and become the fastest growing mobile platform today. It is the horse to bet on for the future of not only mobile devices, but also other types of consumer electronic devices, including tablets and iTV.

What you will learn about in this book spans from how and where to get the Android development environment to how to set it up properly, how to configure it optimally, and how to use it to create applications that employ the powerful features of Android.

The three basic components you'll need for Android development are Java, Eclipse, and of course, Android. You can download these various components for free, as described in this chapter. Once the Android SDK is installed in Eclipse, that IDE becomes a comprehensive Android application development environment.

The next chapter provides an overview of what you will learn in this book, and then we'll get started with setup in Chapter 3.

Chapter 2

What's Next? Our Road Ahead

Before getting into the details of Android development, we'll take a look at our "road ahead." This chapter provides an overview of what is covered in this book, and why it's covered in the order we will cover it.

You will see the logical progression throughout the book of how each chapter builds upon the previous ones. We'll move from setting up the IDE in Chapter 3, to learning how Android works in Chapters 4 and 5, to adding exciting visuals and user interfaces (UIs) in Chapters 6 through 8, to adding interactivity and complexity in Chapters 9 through 11. The final chapter inspires you to keep learning about the more advanced features of the Android platform.

Your Android Development IDE

In Chapter 1, you downloaded the Java SE, Eclipse, and Android SDK packages you need to build an environment for creating Android applications. In Chapter 3, you'll learn how to set up the tools you'll use throughout the rest of the book. You'll do this by creating, step by step, from scratch, the very latest Android IDE out there—right on your very own development workstation.

Note that part of this process must be done while online, so be sure to have your Internet connection active and firing on all cylinders. We'll be connecting in real time, via Google's Android Developers web site, to the latest Android application development tools, plug-ins, drivers, and documentation.

Although it might seem that the setup of Java SE, Eclipse IDE, Android's SDK, and an Android Virtual Device (an emulator that mimics the behavior of a real Android smartphone) is a topic too trivial for an entire chapter, that task is actually one of the most critical in this book. If your IDE does not work 100% perfectly, your code will not work 100% perfectly. In fact, without a robust and properly configured IDE, you may not be able to develop any code at all!

The Eclipse IDE is a sophisticated programming environment that features code highlighting, device emulation, logic tracing, debugging, and a plethora of other features. Figure 2–1 shows an example of working in Eclipse, and Figure 2–2 shows an Android Virtual Device in action.

> **NOTE:** An Android Virtual Device is an emulator that mimics the behavior of a real Android smartphone, as shown in Figure 2–2.

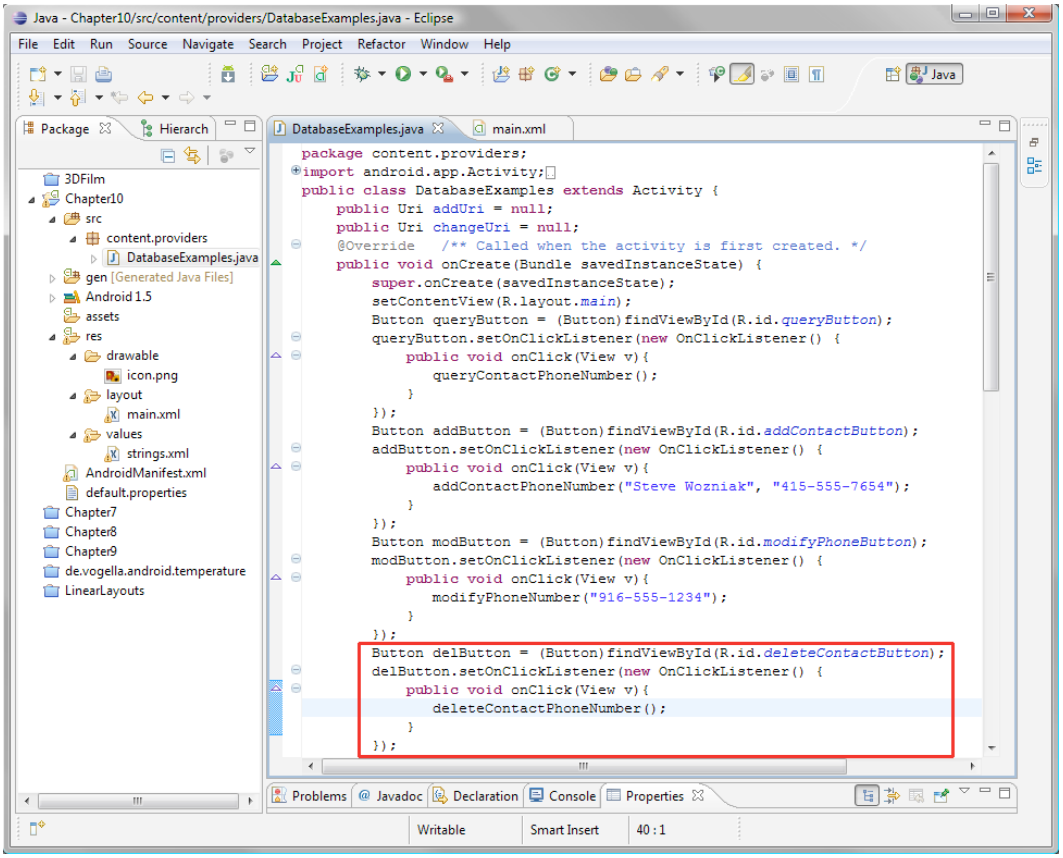

Figure 2–1. *The Eclipse IDE*

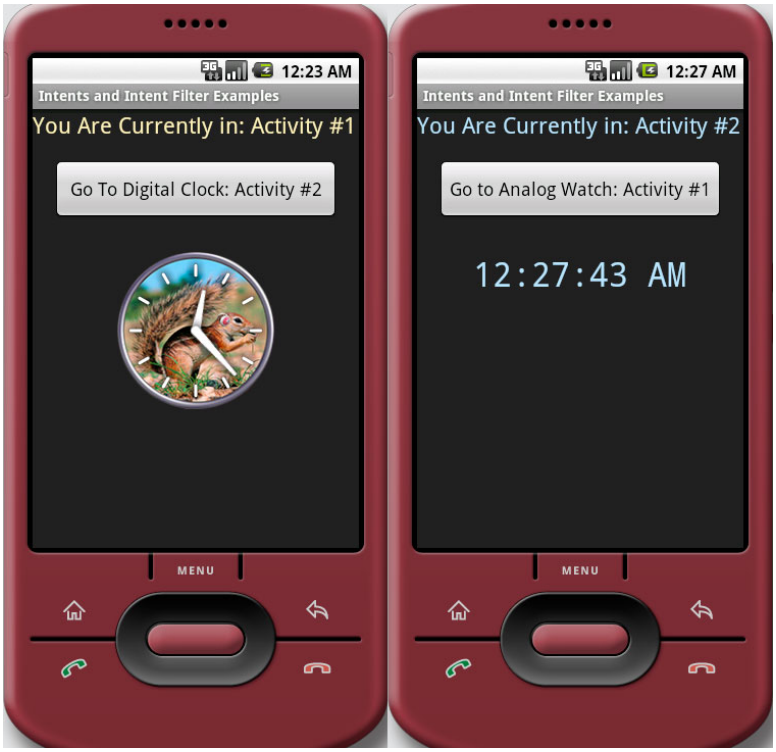

Figure 2–2. *An Android Virtual Device (AVD) in action*

In Chapter 3, you will learn how to customize the Eclipse IDE with Android plug-ins, which will morph the tool into one tailored to the particular needs of an Android developer like you. As you will see, setting up this IDE for your specific development goals is not a trivial undertaking.

Java, XML, and How Android Works

As you'll learn in Chapter 4, an Android application is "stratified." Its functionality is spelled out in Java code, XML markup, and the Android manifest in a way that is truly unique. This adds a great deal of extensibility, or development flexibility, to applications. Android makes heavy use of an XML-based markup language to define the basic components of an application, especially its visual components. Markup is not technically code, but rather consists of tags, similar to the HTML tags web developers use to format their online documents. XML is used in Android to define everything from UIs to data access, and even programmatic constructs like Java object definitions and configurations.

XML markup tags are easier for beginners to comprehend than a complex programming language like Java. For this reason, you'll use XML throughout this book whenever possible, as Google recommends. Here, you'll get a basic beginning knowledge of

Android application development, yet this will still give you the ability to make your apps look very elegant and professional. I call it getting the maximum return on your investment, and XML makes this possible.

The Android Application Framework

By the time you reach Chapter 5, you'll have built a rock-solid integrated Android software development environment and acquired a basic understanding of the components that make up an application development project (images, text, layout, buttons, code, audio, video, animation, XML, and so on).

In Chapter 5, you'll learn the unique lingo of Android application design—that is, what the various components of an Android application are called.

I'll outline how Java programming code and XML, along with any new media resources, are compiled, compressed, and bundled into Android's signature *.apk* file type (APK stands for *A*ndroid *P*ac*K*age), and how logical Android components talk to each other in an application.

The chapter also provides an overview of Android *activities*, which define the user experience on the screen, and explains how they operate. You'll learn about Android *services as well*, which run in the background, separate from the application's activities, and provide the user with advanced functions through the UI.

You'll also take an initial look at *broadcast receivers,* which alert an Android application to events of interest, such as the activation of a camera on an Android device or an incoming phone call. In fact, your app can even send out its own broadcasts, if there is some reason to let other applications know of a change in state in one of your application's data constructs.

The chapter finishes up with a look at *content providers*, which are often databases filled with information, such as a contact list, that applications may want to access to provide functionality of their own. Android ships with a number of preconfigured content providers, and you can also write your own.

Screen Layout Design

By Chapter 6, you will have a better idea of how the Android operating system works internally, and how it wants to see applications put together. You'll be ready to design graphics, UIs, and even user experiences for your applications.

You'll do all of this using screen constructs called *views* and *view groups* (grouped views) and flexible *layout* containers, which can all be nested within each other to create the UI your application needs.

Chapter 6 explains how the display screen—the way most users interact with an Android application—is handled in Android with a mixture of Java code and XML markup that controls the hierarchy of `View` and `ViewGroup` objects and `Layout` containers. You can