



Android Apps for Absolute Beginners

Covering Android 7

—

Fourth Edition

—

Wallace Jackson

Apress®

Android Apps for Absolute Beginners

Covering Android 7

Fourth Edition



Wallace Jackson

Apress®

Android Apps for Absolute Beginners: Covering Android 7

Wallace Jackson
Lompoc, California, USA

ISBN-13 (pbk): 978-1-4842-2267-6
DOI 10.1007/978-1-4842-2268-3

ISBN-13 (electronic): 978-1-4842-2268-3

Library of Congress Control Number: 2017934892

Copyright © 2017 by Wallace Jackson

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Steve Anglin
Development Editor: Matthew Moodie
Technical Reviewer: Chaim Krause
Coordinating Editor: Mark Powers
Copy Editor: Karen Jameson
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 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. Apress Media, LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles 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 Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484222676. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

This Android Apps for Absolute Beginners book is dedicated to everyone in the open source community who is working so diligently to make professional application development software and media content development tools freely available to multimedia application developers to utilize to achieve creative dreams and financial goals.

Last, but not least, I dedicate this book to my father, Parker Jackson, my family, my life-long friends, and content production ranch neighbors, for their constant help, assistance, and those stimulating, late night BBQs.

Contents at a Glance

About the Author	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii
■ Chapter 1: An Introduction to Android 7.0 Nougat	1
■ Chapter 2: Setting Up an Android Studio Development System	17
■ Chapter 3: An Introduction to the Android Studio Integrated Development Environment	33
■ Chapter 4: Introduction to XML: Defining Android Apps, UI Design, and Constants.....	59
■ Chapter 5: Introduction to Java: Objects, Methods, Classes, and Interfaces	91
■ Chapter 6: Android User Interface Design: Using Activity, View, and ViewGroup Classes.....	121
■ Chapter 7: Making Apps Interactive: Events and Intents	147
■ Chapter 8: Android Design Patterns: UI Design Paradigms	177
■ Chapter 9: Android Graphic Design: Making UI Designs Visual.....	209
■ Chapter 10: Android Animation: Image and Procedural Animation.....	251

- **Chapter 11: Digital Video: Streaming Video, MediaPlayer, and MediaController Classes..... 291**
- **Chapter 12: Digital Audio: Sequencing Audio Using SoundPool 339**
- **Chapter 13: Android Services and Threads: Background Processing 387**
- **Chapter 14: Android Content Providers: Datastore Concepts..... 415**
- Index..... 473**

Contents

About the Author	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Introduction	xxiii
■ Chapter 1: An Introduction to Android 7.0 Nougat	1
The History of the Android OS: An Impressive Growth.....	1
Advantage Android 7.0: How Can Android Benefit Me?.....	4
The Scope of This Book.....	5
What Is Covered in This Book	6
What Is Not Covered in This Book.....	7
What's New in Android Nougat: Powerful New Features.....	8
Android Apps for the Google Chrome OS: Custom Pointer API.....	8
Power and CPU Optimization: Sustained Performance Mode API.....	9
Seamless Updates: Background Installation to Secondary Partition	9
Multiple Concurrent Windows: Run Two Android Apps at Once	10
Picture in a Picture: Watch Video or TV in an Android TV HD iTV Set	10
Change Display Density: Adjusting Pixel Per Inch (PPI) via Slider	11
Keyboard Themes: Customize Onscreen Keyboard Using Skins.....	11
Enhanced Doze Mode: Control Android 7.0 Device Resting States.....	12

OpenJDK: Moving Android Java from Oracle Java to OpenJDK	12
The Data Plan Saver: Sync Only When Connected to a Wi-Fi Portal	13
The Future of Android: 3D, VR, AR, OpenGL, and Vulkan	13
Khronos Vulkan: i3D Rendering Engine That Replaces OpenGL ES.....	13
Vulkan for Android: Leading-Edge i3D Performance for Android 7.0.....	14
OpenGL ES for Android: Desktop i3D Performance for Android 7.0	14
Summary.....	15
■ Chapter 2: Setting Up an Android Studio Development System	17
Assembling Your Android 7 Development Workstation.....	17
Android Development Workstation: Hardware Foundation	19
Android Development Workstation: Software Foundation	20
Java 8: Download and Install a Foundation for Android	20
Android Studio: Download and Install Android Studio 2	23
Open Source New Media Content Software: UI and UX.....	28
New Media Software: Download and Installation Work Process	29
Other Affordable New Media Software Readers Should Know About.....	29
Summary.....	31
■ Chapter 3: An Introduction to the Android Studio Integrated Development Environment	33
Android Application Structure: Java, XML, and Assets	34
Android 7 Platform Structure: A Collection of Open Technologies	35
Android 7 Executable Structure: Compiled Runtime Java Bytecode	36
Creating Android 7 Apps: Android Studio's New Project.....	37
The Android Studio Welcome Menu: Creating a New Android 7 App	38
Exploring Your Android Studio Project: The Android App Structure.....	45
Android Resource: Project Folder Hierarchy for Assets.....	48
Android Drawables: Images or Illustration That Draws on the Screen	49
Android User Interface Design Layout: Asset to Design UI Layout.....	50
Android Menu Design: Asset to Define Menu Structure and Options.....	51
Android Data Values: Assets to Define Fixed Application Constants.....	52
Android Anim Folder: Assets Defining Vector or Tween Animation	54

Android Animator: Assets for User Interface Property Animation	54
Android Raw Folder: Pre-Optimized Video and Audio Asset Files	54
Android XML: Arbitrary XML and Configurations	55
Updating Android Studio: Upgrading an SDK over Time	55
Summary	57
■ Chapter 4: Introduction to XML: Defining Android Apps, UI Design, and Constants	59
Extensible Markup Language: XML Overview	60
XML Naming Schema: Tag and Parameter Repository	61
XML Syntax: Containers, Brackets, and Nesting	64
XML Referencing: Chain XML Constructs Together	66
XML Constants: Adding New Constants Using XML	68
XML Dimensions: Editing Dimensions Using XML	70
Alternate XML Resource: Dimensions for Tablets	71
XML Styles: Editing Styles or Themes Using XML	72
XML Colors: Define Application Color Using XML	74
Configuring an App Using XML: Android Manifest	75
UI Design Editor: XML Markup Generation	76
Summary	90
■ Chapter 5: Introduction to Java: Objects, Methods, Classes, and Interfaces	91
The Three Versions, or Editions, of Java	92
A Foundation of OOP Constructs: An Object	92
Some Programming Terms: Variable, Method, and Constant	95
Java Constructs: Create Your Own Objects	96
The Java Class: Java Code Structure Container	96
The Java Method: Java Code Function Definition	98
Constructor Methods: The Java Object Blueprint	102
Instantiating Objects: The Java “new” Keyword	104
Extend an Object’s Structure: Java Inheritance	105
Java Interfaces: Defining Class Usage Patterns	107

Logical Collection of Classes: Using a Package.....	110
The API.....	111
Modifiers: Data Type, Access, Inheritance	112
Java Access Modifiers: Four Levels of Access	112
Non-Access Modifiers: Static, Final, and Abstract	114
Analyzing Your MainActivity.java Class.....	117
The AppCompatActivity Class: Spans OS Versions	118
Summary.....	119
■ Chapter 6: Android User Interface Design: Using Activity, View, and ViewGroup Classes	121
How Activity, View, and ViewGroup Classes Interrelate	122
The ViewGroup Class: A Known Direct Subclass of View.....	124
The View Class: A Foundation of User Interface Design	126
The Activity Class: A User Interface Design Container.....	127
Creating UI Design from Scratch	129
Add a CheckBox User Interface Element to your Design	134
Add a TextView User Interface Element for Your Title	136
Add a Button User Interface Element for Interactivity	137
Squashing Bugs (Removing Errors) in the Design Editor.....	139
Eliminate Any Remaining Errors Using the XML Text Editor.....	143
Summary.....	145
■ Chapter 7: Making Apps Interactive: Events and Intents	147
About Intent Objects: The Android Intent Class	148
Intent Types: Explicit Intent versus Implicit Intent	149
IntentFilter: Construct an Implicit Intent Definition.....	150
Instantiating an Intent Object: Passing App Context.....	151
Explaining Context: The Android Context Class.....	152
Event Processing: Using Events with Event Listeners.....	153
Events: Turning Device User Interaction into Events.....	153
Event Listener: Java Methods Process UI Widget Events	154
Event Handler: Java Methods Process Global Event Type.....	155

Creating a Second Activity: The <code>UniverseActivity</code> Class.....	155
Adding Event Listeners to the Activity Button Objects.....	161
Adding Intent Processing to your Event Handling.....	165
Emulating Hardware: Creating an AVD to Test Your App.....	168
Running the Application: Building the App Using Gradle.....	173
Summary.....	175
■ Chapter 8: Android Design Patterns: UI Design Paradigms	177
Android Design Patterns: Ensuring App Visual Quality	178
Material Design: i3D Animated User Experience Designs.....	178
Hardware Devices: Code Design Patterns Across Devices	179
Pure Android: Application Design Branding Conformance.....	182
Creating a Sliding Drawer: UI Only When Users Need It.....	186
Creating Menu Structures for a UI Design: The Android Menu Interface	195
Deprecated Java Code: Researching Replacement APIs.....	205
Summary.....	207
■ Chapter 9: Android Graphic Design: Making UI Designs Visual.....	209
Imaging Concepts, Formats, and Techniques.....	210
The Foundation of Digital Images: The Pixel.....	210
The Shape of a Digital Image: The Aspect Ratio	211
Coloring Your Digital Images: RGB Color Theory	212
Image Compositing Transparency: Alpha Channels	215
Algorithmic Image Compositing: Blending Modes.....	216
Masking Digital Imagery: Using Alpha Channels	216
Smoothing Edges: The Concept of Anti-Aliasing.....	217
Optimizing Digital Images: Compress and Dither	218
Using Indexed Color Imagery: Dithering Pixels.....	220
Android Image Formats: Lossless versus Lossy.....	222
Creating Android <code>NinePatchDrawable</code> Assets.....	223
Installing the Draw 9-Patch Source PNG32 Image	224
Exploring Android Studio's 9-Patch Editor.....	225
Using Your <code>NinePatchDrawable</code> Asset in Android.....	231

- Using NinePatchDrawable Assets in an App..... 232
- Creating Multi-state PNG32 Image Assets 235
- The ImageButton Class: Multi-state Button..... 240
 - The States: Normal, Pressed, Focused, Hovered 242
- Creating Android Multi-state ImageButtons 243
- Summary..... 249
- **Chapter 10: Android Animation: Image and Procedural Animation..... 251**
- Frame Animation: Concepts and Techniques..... 252
 - Frame Animation: Cels, Frames, and Terminology 252
 - Android Image Format: PNG, GIF, JPG, WebP, BMP 252
 - Optimizing Frames: Color Depth and Frame Count..... 253
 - Animation Resolution: Pixels Add to File Size!..... 254
- Frame Animation: Using AnimationDrawable 255
 - Creating Frame Animation: XML Frame Definition 256
- Creating Frame Animation in XML and Java 258
 - Create the XML Frame Animation Definition File 258
 - Create ImageView and AnimationDrawable Objects..... 265
- Android Tween Animation: Vector Concepts 268
 - Procedural Concepts: Rotate, Scale, Translate..... 268
 - Procedural Data Values: Ranges and Pivot Point 269
 - Procedural Animation Compositing: Alpha Values 270
 - Procedural Timing: Using Duration and Offsets 271
 - Procedural Loops: RepeatCount and RepeatMode..... 272
- Android Animation Class: Tween Animation 273
- Creating Tween Animation Using XML Markup..... 274
 - Create an /anim Folder: Tween Animation Assets..... 274
 - Android ScaleAnimation Class: Animated Scaling 275
 - The Scale Transform: Configuration Parameters 276
 - Android AnimationSet Class: Transform Grouping 278
 - AnimationSet Container: Groups and Subgroups..... 279
 - Java Code: Tying Two Animation Types Together 280

Complex Animation: Android Interpolators	283
Creating Complex Animation Using XML Markup	283
Android Interpolator Interface: Motion Curves.....	284
Java Code: Two Widgets Use the Tween Animation	286
Procedural Animation or Frame Animation?.....	287
The Animator Class: Parameter Animation	289
Summary.....	289
Chapter 11: Digital Video: Streaming Video, MediaPlayer, and MediaController Classes.....	291
Creating a Video App: FullscreenActivity	292
The FrameLayout Class: Framing DV Content	295
FrameLayout.LayoutParams Nested Class: Gravity	297
The VideoView Class: A VideoPlayer Widget	299
The VideoView Lifecycle: Video Playback Stages	300
Create a VideoView Layout Design with your XML	303
Digital Video Concepts: Bitrates and Codecs.....	305
Digital Video in Android: MPEG4 H.264 and WebM	306
Digital Video Compression: Bitrate and Streams	307
Digital Video Optimization: Codec and Settings.....	308
Creating Digital Video Content: Terragen4	311
Digital Video Compression: Sorenson Squeeze 11	314
Creating a Digital Video Folder: Raw Resources.....	319
The Uri Class: Referencing the Video Data	321
The Uri.parse() Method: Loading Your VideoView	323
Android MediaPlayer: VideoPlayback Engine	328
Android MediaController: A VideoTransport	329
Add a Video Transport UI Using MediaController	330
Loop Digital Video: Using OnPrepareListener.....	334
Streaming Digital Video: Using HTTP URL in URI.....	336
Summary.....	337

- **Chapter 12: Digital Audio: Sequencing Audio Using SoundPool 339**
 - Audio Waves: History, Concepts, and Theory 340
 - Foundation of Analog Audio: Sound Waves of Air 340
 - Digital Audio: Samples, Resolution, and Frequency 341
 - Digital Attributes: HD, Stream, and Bitrate 343
 - Android Digital Audio: Digital Audio Formats 343
 - Digital Audio Optimization: Device Compatible 346
 - Audio Sequencing: Concepts and Principles 347
 - Audio Synthesis: Concepts and Principles 348
 - Raw Audio Data Optimization: Memory Footprint 349
 - Digital Audio Synthesis and Sequencing Caveats 350
 - Audacity 2: Creating Digital Audio Assets 351
 - Audacity 2.1.2: Installing Software and Codecs 352
 - Free Audio: Locate DigitalAudioSequencer Audio 354
 - Digital Audio Optimization: Concepts and Formats 356
 - DigitalAudioSequencer: ScrollingActivity 360
 - Android SoundPool: Digital Audio Engine 368
 - Add SoundPool Engine to DigitalAudioSequencer 370
 - Android AudioAttributes: Configuring SoundPool 374
 - Configuring a SoundPool Using AudioAttributes 376
 - Summary 385
- **Chapter 13: Android Services and Threads: Background Processing 387**
 - Android’s Service Class: Characteristics 388
 - Controlling Your Service: Privacy and Priority 390
 - Processes or Threads: Foundational Information 391
 - Spawn a Process: android:process XML Parameter 392
 - The Process Life Cycle: Keeping a Process Alive 393
 - Thread Caveats: Don’t Interfere with UI Thread 397
 - Should Android Apps Use Services or Threads? 398

Creating a Service: AmbientAudioService	400
Configuring AndroidManifest to add a <service>.....	402
Service: Background Processing Services.....	403
Configure AmbientAudioService: Play Audio	403
Starting a Service: Using .startService()	410
Summary.....	413
■ Chapter 14: Android Content Providers: Datastore Concepts.....	415
Overview of Android Content Providers: Sharable Data	416
Database Fundamentals: Concepts and Terms	417
SQLite: An Open Source Database Engine.....	419
Android's Built-In SQLite DBMS Content Providers	420
Android 1.5 Contacts Database Contact Provider.....	421
Deprecated Database Structures: Software Upgrades	421
The Android MediaStore Content Providers.....	425
The Android CalendarContract Content Providers	425
Referencing the Content Provider: Using a Content URI.....	426
Creating a Basic Activity: The SQLiteProvider Project	428
Examining and Testing Your SQLiteProvider Bootstrap.....	430
Configuring the Manifest: Uses SDK and Permissions	433
Creating Your Dummy Contact Database Using an AVD	436
RelativeLayout: Create Morphing User Interface Design.....	441
Creating Your RelativeLayout UI for MainActivity.....	442
Transform the MainActivity Class for Database Access	454
Creating Your Custom .listContacts() Database Access Method	455
Writing to a Database: Using the ContentValues Object.....	463
Summary.....	472
Index.....	473

About the Author



Wallace Jackson has been writing for leading multimedia publications about his work process for interactive new media content development since the advent of *Multimedia Producer Magazine* nearly two decades ago, when he wrote about advanced computer processor architecture for a special issue centerfold (removable “mini-issue” insert) distributed at the SIGGRAPH trade show. Since then, Wallace has written for a significant number of other popular publications about his work in interactive 3D and new media advertising campaign design, including *3D Artist Magazine*, *Desktop Publishers Journal*, *Cross Media Magazine*, *AV Video and Multimedia Producer Magazine*, *Digital Signage Magazine*, and *Kiosk Magazine*.

Wallace Jackson has authored more than half a dozen Android Development book titles for Apress, including several titles in the popular *Pro Android* series. This particular *Android Apps for Absolute Beginners* title has been rewritten entirely from scratch four times, and this fourth edition is one of the most thorough and comprehensive Absolute Beginner Android titles to be found in the market.

Wallace is currently the CEO of **Mind Taffy Design**, a new media content production and digital campaign design and development agency located in North Santa Barbara County, halfway between clientele in Silicon Valley to the north and in Hollywood, “The OC,” and San Diego to the south. Mind Taffy also produces interactive 3D content for major brands around the world from their content production studio on Point Concepcion Peninsula in the California Central Coast area. Mind Taffy Design has created open source technology (HTML5, Java, and Android) and digital new media content deliverables for more than a quarter century (since 1991) for a large number of the top-branded manufacturers in the world, including Sony, Samsung, IBM, Epson, Nokia, TEAC, Sun, SGI, Dell, Compaq, ViewSonic, Western Digital, CTX International, KDS USA, KFC, ADI, and Mitsubishi.

Wallace received his undergraduate degree in Business Economics from the University of California at Los Angeles (UCLA) and his graduate degree in MIS Design and Implementation from the University of Southern California (USC). His postgraduate degree from USC is in Marketing Strategy. He also completed the USC Graduate Entrepreneurship Program at USC's popular Marshall School of Business MBA program. You can connect with Wallace at: <http://www.linkedin.com/wallacejackson> and follow him on Twitter @wallacejackson if you like, or visit the iTVset.com or iTVclock.com websites to see his i3D HD and UHD work.

About the Technical Reviewer



Chaim Krause presently lives in Leavenworth, Kansas, where the U.S. Army employs him as a Simulation Specialist. In his spare time he likes to play PC games and occasionally develops his own. He has recently taken up the sport of golf to spend more time with his significant other, Ivana. Although he holds a BA in Political Science from the University of Chicago, Chaim is an autodidact when it comes to computers, programming, and electronics. He wrote his first computer game in BASIC on a Tandy Model I Level I and stored the program on a cassette tape. Amateur radio introduced him to electronics while the Arduino and the Raspberry Pi provided a medium to combine computing, programming, and electronics into one hobby.

Acknowledgments

I would like to acknowledge all my fantastic editors and the support staff at Apress who worked long hours and toiled so very hard on this book to make it the ultimate *Absolute Beginner* Android title.

Matthew Moodie, for his work as the Lead Editor on the book, and for his experience and guidance in the process of making this book one of the great Android Absolute Beginner development titles.

Mark Powers, for his work as the Coordinating Editor on the book, and for his constant diligence in making sure I hit or surpassed my deadlines.

Chaim Krause, for his work as the Technical Reviewer on the book, and for making sure I didn't make any programming mistakes. Java code with mistakes does not run properly, if at all, unless they are very lucky mistakes, which is quite rare in computer programming these days.

Frank Serafine, my close friend, the world's finest and most respected sound designer, and popular rock musician, for contributing the background audio sample used in this book. This audio sample is from his stellar (no pun intended) work on some of the world's most popular science fiction as well as action adventure movies and television shows, including but not limited to *Star Trek* and *Hunt for Red October*.

Finally, I'd like to acknowledge **Oracle** for acquiring Sun Microsystems, and for continuing to enhance Java so that it remains the premiere open source programming language; and **Google**, for making Android the premiere open source operating system, and for acquiring ON2's VP8 and VP9 video codecs (WebM) and making these available to multimedia producers on the Android OS and HTML5 platforms, allowing open source video encoding performance similar to HEVC (H.265).

Introduction

The Android OS is currently the most popular operating system in the world. The Android OS runs on everything from smartwatches to HD or UHD smartphones to touchscreen tablets to e-book readers to game consoles to smartglasses to smartwatches to auto dashboards to new ultra-high definition interactive television sets (or iTV sets). If you want your apps to run everywhere, Android is the optimal solution.

There are even more types of consumer electronics devices, such as those found in the automotive, home appliance, security, robotics, drones, photography, industrial, and home automation markets, which are adopting the open source Android OS as their platform as time goes on. This book will show you how to develop applications for these new device-type verticals as they emerge into the market.

Since there are literally billions of Android consumer electronics devices owned by billions of people all over the world, it stands to reason that developing great Android applications for all these people might be an extremely lucrative undertaking, assuming that you have the right concept and design.

This book will help you go a long way toward learning how to develop Android applications that will run across the plethora of Android-compatible consumer electronics devices; and across all popular versions of the Android OS, including 32-bit Android 4.4 OS and the 64-bit Android 5, 6, and 7 OSes.

Developing an Android application that works well across all of these types of consumer electronics devices requires a very specific work process if you are an Absolute Beginner, involving leveraging all the Android Studio helper features, which I cover during this book. I had to write *Android Apps for Absolute Beginners, Fourth Edition*, from scratch because most of Android Studio 2.3's new features are targeted at Absolute Beginners. This book is intended for readers who are Absolute Beginners to Android development. Of course you must be technically savvy, but this book is for readers who are not yet familiar with computer programming concepts and techniques.

The book will be more advanced than previous editions of *Android Apps for Absolute Beginners*. The first edition of this book was a mere 300 pages, as Android 1.5 was the first version to appear on Android hardware devices (smartphones), and a second edition of this book was 33% longer, at 400 pages. I've expanded this version of the book even more.

I designed this book to be a more comprehensive overview of the Android application development work process than most beginning Android application development books, because, at this point, there is really no way to sugarcoat the Android application development process. The new Android Studio IDE is however attempting to help beginners code and design applications using some helper features and drag-and-drop visual design editors that we will cover in detail in the book.

To become the leading Android 7 application developer that you seek to become, you will have to understand, as well as master, XML markup, user interface design, Java 8 programming, as well as new media content creation. Once you have done this, hopefully by the end of this book, you will be able to create the vanguard user experience required to create popular, best-selling Android 7 apps.

Android apps used to be developed for 32-bit Android 1.x through 4.x using Eclipse ADT's IDE. Starting with 64-bit Android 5, Eclipse ADT IDE was replaced with Android Studio. Android applications are not developed via Android Studio alone (currently at version 2.3), but are also developed in conjunction with several key genres (2D, SVG, i3D, audio, video, imaging, SFX, etc.) of new media content development software packages.

For this reason, this book covers a wide variety of popular open source software packages, including GIMP 2.8.18, Planetside Terragen 4, Sorenson Squeeze Pro 11, and Audacity 2.1. These professional new media content production tools should be utilized in conjunction with developing your Android 7 applications. This book will show you exactly how to accomplish this, as well as how to download, install, update, configure, and actually use a number of the popular open source software packages.

This comprehensive Android 4/7 application development work process will allow you to experience exactly how the use of all of these multimedia content development software packages needs to fit into your overall Android application development work process. This 100% comprehensive "soup to nuts" multimedia-centric Android app development approach sets this 32-bit Android 4.4 and 64-bit Android 5/6/7 book title distinctly apart from all of the other Android application development titles that are currently on the market. This book covers an Android development process at a broad level while at the same time showing the Absolute Beginner Android application developer how to use an Android Studio handholding approach, by leveraging Android Studio's ability to code bootstrap application projects allowing you to simply add your application specifics, use pop-up Java and XML code helper dialogs, use code completion, utilize a new Visual Design Editor, and implement backward compatibility features.

Chapter 1 starts at the absolute ground level, explaining what Android is, where it came from, what it is used for, where it is going, what its benefits include, what is covered in this book, as well as what is not covered in this book, as well as what some of the new features in Android 7.1.1 (called Nougat) include.

Chapter 2 covers how to assemble your Android Studio 2.3 application development workstation from scratch. This starts by covering hardware requirements and considerations, and then downloading and installing the current Java SE 8 JDK and Android Studio 2.3, along with more than half a dozen powerful open source content development applications, including Fusion, Audacity, Inkscape, Blender, GIMP, Lightscape, Open Office, and more.

Chapter 3 gives you an overview of how the Android Studio IDE, Java 8, XML, multimedia assets, and Android hardware devices are used together to create Android applications. You will be exploring the Android Studio 2.3 IDE, and learn how to have Android Studio create an application Java code and XML markup infrastructure for you to use to create your Android 7.1.1 application with. You will examine an Android Project structure and hierarchy, to see how everything comes together, and will learn about the project resource folder hierarchy and new media formats and genres that are supported in Android, of which there are many. You will learn about Android Drawables, Animation, and Menu capabilities, and how to pre-define data, constants, user interface design, and multimedia assets for your applications by using XML markup. You will learn how to update your Android Studio 2.3 IDE, so you can keep your Android development workstation current, as new versions are released.

Chapter 4 teaches you all about the XML markup language, including how to use the new Android Studio Visual Design Editor to generate entire XML markup code listings that define complex user interface design. We will dissect what XML parent and child tags are used for in UI layout containers and how to create functional UI elements in a UI design using widgets. You will also look at how XML can be used to define your application constants (fixed settings, UI designs, themes, and assets that will not change during your application's usage). We will do all of this while expanding on the Hello World application that you created in Chapter 3, by adding 2D graphics and a user interface design to make the application more professional, and show you how to add appeal to your bootstrap Android 7.1.1 app.

Chapter 5 serves as a Java 8 primer for those not yet exposed to Java 8, and as a review for those who have been exposed to Java 8 before. You learn all about the Java 8 SE programming language, including packages, classes, methods, constants, variables, interfaces, modifiers, keywords, versions, objects, and OOP concepts and techniques. All of this is additionally demonstrated using a sample Java code project and its structures. Thus, the first third of this Android 7.1.1 book is foundational material, which explains how the Android 7 OS works together as a whole, as well as how each of these components works in and of itself. You will build on this learning material throughout the remainder of the book.

Chapters 6 explores user interface design concepts, techniques, and workflows for Android Studio 2.3 – specifically your Visual Design Editor usage, and the Android View, ViewGroup, and Activity classes, which user interface design code, UI layout containers, and UI elements (called widgets) come from. You learn how View and ViewGroup subclasses are used to create UI layout containers filled with UI widgets, and how an Activity screen hosts and displays these, and all about the Activity class life cycle, and how these organize the application Java 8 code into logical phases (create, start, pause, resume, stop, destroy). You create a user interface design from scratch, for the project you created in Chapters 3 and 4, including graphics (Drawable objects) assets, UI design elements, and Android 7 OS themes.

Chapter 7 takes your static application, and makes it interactive, by using Android 7 Intent objects and coding Java event handling and event listening constructs for user interface elements defined using XML. This is where the book starts to get more complicated, both conceptually, with real-time event queue processing and application component communication using Intent objects, as well as Java 8 code-wise, with deeply nested event processing structures, explicit Intent objects, and implicit Intent Filter concepts. You will learn about the Android Intent class and objects and about what operating system events do to allow applications to know what is going on in the operating environment and what users are doing with the hardware and with user interface designs (covered during Chapter 6).

Chapter 8 examines Pure Android UI design patterns. These are guidelines for making your Android applications look and feel like they belong on the popular Android platform. This chapter covers the new Material Design paradigm, introduced in 64-bit Android 5, and advanced concepts such as i3D UI design, real-time automated shadowing, and animated user interface design concepts. We cover different types of hardware devices that run Android OS, and the different types of Android APIs, such as the core API for smartphones and tablets; the Wear 2.0 API, for smartwatches; the Glass API, for smartglasses; the Auto API, for automobile dashboards; and the Android TV API, for iTV Sets. I show you how to have Android Studio create one of these design paradigms, the sliding drawer UI design, for you, and how to examine the XML and Java code to understand how it works under the hood. We then test and configure this application using an AVD emulator and an Android manifest XML definition file.

Chapters 9 delves into the concepts and work process of Digital Imaging for Android, including how to create NinePatch drawables; how to use GIMP 2.8.18 to create image assets; Android Drawable classes; supported digital image formats; and digital image concepts such as pixels, resolution, alpha channels, aspect ratio, color depth, dithering, compositing, blending modes, anti-aliasing, and similar concepts that are supported across Android's advanced digital image compositing, rendering, and Porter-Duff blending and transfer (compositing) algorithms. I show you how to have Android Studio 2.3 create a navigation drawer UI design pattern for you, and how to create multi-state image buttons.

Chapter 10 brings your application into the fourth dimension using 2D animation engines in Android such as the frame (bitmap) AnimationDrawable class and the tween (vector) Animation class. This chapter outlines the necessary animation concepts, data formats, and classes, and shows you how to create frame (or bitmap) animation, tween (vector, or procedural) animation, and a hybrid (frame and procedural combined) animation. We also add these animation assets to the NavDrawerPattern project that you created in Chapter 9, to show how different animation approaches will work inside of one of the five different pure Android design patterns you'll create during the course of this book.

Chapter 11 shows you how to stream Digital Video content to your Android application by using the Uri, MediaPlayer, and MediaController classes. This chapter goes into detail regarding digital video concepts, formats, and data footprint optimization, and also shows you how to create a 3D digital video asset from scratch, using Planetside Software's Terragen 4 and Sorenson Squeeze Desktop Pro 11. You'll code a DigitalVideoMedia project using yet another Android design pattern (FullscreenActivity) and address how to code a captive (part of the APK) video, and a streaming (from an external server) video asset.

Chapter 12 gets your Android hardware device’s speakers involved by adding digital audio assets to your application, using the powerful Android `SoundPool` class. This chapter goes into detail regarding digital audio concepts, formats, and data footprint optimization, and also shows you how to optimize a digital audio asset from scratch, using open source Audacity 2.1.2. You’ll code a fourth, `DigitalAudioSequencer` project, using yet another pure Android design pattern (`ScrollingActivity`) and you’ll code a children’s educational application that teaches kids the sounds that different animals make.

Chapter 13 offloads processing-intensive tasks to Android OS for background processing. You’ll learn about advanced operating system processing concepts like threads, processes, and services, which are used to perform background processing using the Android `Service` or `Thread` class. We’ll add this capability to the `DigitalAudioSequencer` application you created in Chapter 12 so it can play ambient background audio as a `Service` process, using the Android `MediaPlayer` operated by the `Service` class.

Chapter 14 looks into the Android SQLite database management system, as well as Android `ContentProvider`, `ContentResolver`, and `ContentValues` classes. You will learn about RDBMS database theory and how to use Android content providers to access databases.

Whereas the first five chapters of the book are foundational information, in the final five chapters of this book, you will learn about some of the more advanced development topics that normally would not be included in an Absolute Beginner title. I included these so that the important topics regarding leading-edge Android application development are all in this one, single, unified book. The included advanced topics include 2D animation, digital video, digital audio, threads, processes, and SQLite databases.

This book attempts to be the most comprehensive Absolute Beginners book for Android application development out there, by covering most, if not all of, the significant Android Studio 2.3 application development assistance features, and those core Android classes that will always need to be used to create leading-edge, 32-bit Android 4.4.4, or 64-bit Android 5.0 through 7.1.1, software applications.

It is the intention of this book to take you from being an “Absolute Beginner” in Android application development, to having a comprehensive, solid, intermediate knowledge of both 32-bit Android 4.4 and 64-bit Android 7.1.1 application development.

You should be advised that this book contains a significant amount of technical knowledge and work processes that may take more than one read-through to assimilate into your application development knowledge base (your current Android knowledge “quiver of arrows,” so to speak). This vast journey developing backwardly compatible applications for 32-bit Android 4.0 (API Level 15) through the 64-bit Android 7.1.1 (API Level 25, and later versions) will be well worth your time, however; rest assured.

Chapter 1

An Introduction to Android 7.0 Nougat

These days, you will see Android OS powered devices of every size and shape everywhere you look. They can be worn on your person, thanks to Android WEAR; used in an appliance, thanks to Android TV; and they are a part of your car, thanks to Android AUTO. Android devices will provide you entertainment in your living room taking the form of your iTV set; help you learn at school using a tablet; inform you in bed using an e-book reader; or excite you on the couch using an Android game console, such as the OUYA, the Razer Forge, or the nVidia Shield.

In this chapter, we will explore some basic history regarding Google's Android operating system (OS), to give you a high-level overview of the history of Android. We will look at the benefits of learning Android application development, and which open source programming languages and OSs Android is based upon. We will look at the percentage distribution amongst the different Android versions, and the new features in Android 7.0 Nougat.

The History of the Android OS: An Impressive Growth

The Android OS was originally created by **Andy Rubin** to be an OS for mobile phones. This happened around the dawn of the 21st century. In July of 2005, Google acquired Android and made Andy Rubin the Senior Vice President of Mobile Platforms for Google, where he remained until November of 2014. Many feel this acquisition of Android OS by Google was largely in response to the appearance of Apple's iPhone around that same time. However, there were enough other large players, such as RIM Blackberry, Nokia Symbian, and Microsoft Windows Mobile, that it was deemed to be a savvy business decision for Google to purchase the engineering talent of Android Incorporated along with its Android OS intellectual property. This allowed Google to insert their Internet search engine company into the emerging mobile market, which many now refer to as **Internet 2.0**.

Internet 2.0, or the **Mobile Internet**, allows users of consumer electronic products to access content via widely varied data networks, using portable consumer electronic devices. These currently include tablets, smartphones, phablets (a phone-tablet hybrid), game consoles, smartwatches, smartglasses, personal robots, drones, cameras, and e-book e-readers. These days, Android OS–based devices will also include those not-so-portable consumer electronics devices such as iTV sets, home media centers, automobile dashboards, automobile stereos, music players, home appliances, home control installations, and digital signage system set-top boxes.

This ever-growing Android phenomenon puts new media content such as games, 3D animation, interactive television, digital video, digital audio, e-books, and high-definition imagery into our lives at every turn. Android is one of those popular open source vehicles (others being HTML5 and JavaFX) that digital artists will increasingly leverage in order to be able to develop new media creations that their end users have never before experienced. Over the past decade, Android has matured and evolved, to become a stable, exceptionally reliable, embedded open source OS. An Android OS that started out with its initial version just a decade ago, once acquired by Google, has released stable OS versions 1.5, 1.6, 2.0, 2.1, 2.2, 2.3.7, 3.0, 3.1, 3.2, 3.3, 4.0, 4.1, 4.2, 4.3, 4.4, 5.0, 5.1, and 6.0.

As of the writing of this book, Android 7.1.1 is in beta, with a projected release in Q1 of 2017. Android 7.1.1 should show up in 64-bit Android devices in 2017 and 2018. If you want to see the latest statistics regarding each of the previous Android OS revisions, directly from the Android developer website, you should visit this URL:

<http://developer.android.com/about/dashboards/index.html>

Table 1-1 shows this progression of all the popular versions of Android OS that have been installed on popular embedded OS consumer electronics products over the past decade. I wanted to collect all of this Android OS information together into one single infographic for you, so that you could get a “bird’s eye view” of the current historic progression of the Android OS. As you can see, there are certain Android market share “sweet spots.” In case you’re wondering what an **embedded OS** is, it’s like having an entire personal computer on a motherboard that’s small enough to fit in a handheld device, and which is powerful enough to run applications, or “apps.”

Table 1-1. Released Android OS Versions, Their Internal OS Names, API Levels, and Current Market Share

VERSION	CODENAME	API LEVEL	MARKET SHARE
1.5	Cupcake	3	Less than 0.1%
1.6	Donut	4	Less than 0.1%
2.0, 2.1	Eclair	5, 6, 7	Less than 0.1%
2.2	Froyo	8	Less than 0.1%
2.3.7	Gingerbread	9, 10	2.0% (Kindle Fire)
3.0, 3.1, 3.2	Honeycomb	11, 12, 13	Less than 0.1%
4.0, 4.0.4	Ice Cream Sandwich	14, 15	1.0%
4.1.2	Jelly Bean	16	6.8%
4.2.2	Jelly Bean Plus	17	9.4%
4.3.1	Jelly Bean Plus	18	2.7%
4.4.4, 4.4W	Kit Kat	19, 20	31.6%
5.0	Lollipop	21	15.4%
5.1	Lollipop	22	20.0%
6.0	Marshmallow	23	10.1%
7.0 and 7.1.1	Nougat	24 and 25	Less than 0.1% (so far)

Just like today's personal computers and laptops, the Internet 2.0 devices, such as smartphones, tablets, e-readers, smartwatches, and iTV sets, now feature quad-core (4 CPU) and even octa-core (8 CPU) computer processing power, as well as two gigabytes of system memory. This is approaching the power of a modern-day PC, such as the workstation you are going to set up during the next chapter of this book, which you can get for \$500 at Walmart. Mini-tower PCs feature octa-core 64-bit processors along with 6GB or 8GB of system memory, and a 750GB (or larger) hard disk drive with Windows 10, Fedora 24, or Ubuntu Mate 17.04.

The Android OS contains the power of a complete computer OS. It is based on the **Linux Kernel** open source platform, and Oracle (formerly Sun Microsystems) **Java 8 Standard Edition**, one of the world's most popular programming languages. Android 5 and 6 also use a 64-bit Linux Kernel, along with the Java 7 Standard Edition.

Note This term **open source** refers to software that has been developed collaboratively, usually by an **open community** of individuals, and is **freely available** for **commercial use** (or non-commercial use). Open source software also comes with all the **source code**, so that it can be further modified, if necessary. The Android OS is open source, though Google develops it internally before releasing the source code. From that point on, the source code is freely available for commercial use by software developers.

It's not uncommon for an Android device to have a 2.4GHz processor and 2GB of fast, computer-grade DDR3 memory. This rivals desktop computers of just a few years ago, and notebooks that are still currently available. You will continue to see this convergence of Internet 2.0 (mobile device) OSs with desktop OSs, such as we are seeing with Windows 10 and Windows Mobile, and with Chrome OS and Opera OS currently, as time goes on.

Once it became evident that Java, the Android OS, and open source software platforms were vanguard forces to be reckoned with, a bunch of the popular consumer electronics manufacturers, including Philips, Sony, HTC, Samsung, LG Electronics, and others, formed and then joined the **Open Handset Alliance (OHA)**. This was all done in order to put the momentum behind Google's open source Android platform, and it worked! Today, hundreds of leading branded consumer electronics manufacturers leverage Android as an OS on their consumer electronic devices. In fact, Android OS is used more than any other OS that has ever existed on the planet.

This development of the OHA is a significant benefit to Android developers. Android allows developers to create their applications using a single IDE, or integrated development environment, and now this support by the OHA enables developers to deliver their content across dozens of major branded manufacturers' hardware products, as well as across several different types of consumer electronic devices, including smartphones, iTV sets, e-book readers, smartwatches, game consoles, home media centers, set-top boxes, and touchscreen tablets. The Android OS affords developers a plethora of powerful content delivery tools and device playback possibilities, to say the least! You have realized this, as you are reading this book right now, so you can get in on all this power!

In summary, Android is a seasoned OS 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 the Linux Kernel, Java SE 8, and open standards such as XML, CSS3, MPEG-4, JPEG, PNG, MP3, OGG Vorbis, FLAC, SVG, WebM, WebP, OpenGL 3.2, WebKit, Vulkan, and HTML5. Android incorporates all of these open source resources, so that it can provide the free new media content and application delivery platform to Android developers, and an OS platform to consumer electronics manufacturers. Can you spell OPPORTUNITY? I sure can! It's spelled: **A-N-D-R-O-I-D!**

Advantage Android 7.0: How Can Android Benefit Me?

There are simply too many benefits for the Android OS development platform to ignore your Android applications development workflow, and environment, for even one minute longer. We are going to get you set up with all of the latest IDE's, new media apps, programming languages, and Android OS SDKs and components during the next chapter, so that you will have an extremely valuable Android 7.1.1 multimedia content development workstation.

That's great, but how can Android benefit me, you might be thinking? First of all, Android is based upon open source technology, and it's free for commercial development use, with no up-front costs and no on-going royalties. Android, at its inception, was not as refined as expensive paid technologies from Apple, Adobe, and Microsoft.

During the past several decades, open source software technology has become equally as sophisticated as conventional paid software technologies. You will see this during this book as you work with your professional-level Android Studio 2.3 (IntelliJ) IDEA and the new media

content software that you'll be acquiring (for free) during the next chapter, when we put together an Android Studio content development workstation, 100% from scratch.

The increasing adoption of open source technology over paid software is clearly evident with Internet 2.0, as the majority of the consumer electronics manufacturers have chosen Android and HTML5, based on a Linux Kernel and using Java, JavaFX, and JavaScript, over Windows 10, Windows Mobile, iOS, and Macintosh OS/X OSes.

For this key reason, Android developers can develop applications not only for their smartphones, but also for new and emerging consumer electronics device ecosystems, which include never-before-seen products such as smartglasses and smartwatches, or UHD (4K, or IMAX) iTV sets, which are network compatible and available to connect to the Google Android (Play) Marketplace. The Android App Marketplace was rebranded by Google as Google Play, due to legal action brought by Apple Computer over what Google calls their Android application storefront.

The free nature of open source translates directly to more sales of more consumer electronics devices in more areas of your potential customer's lives, and this offers steadily increasing Android market share, and an ever-increasing incentive to develop for the Android 7.1.1 OS over "closed" technologies such as Windows or iOS, and over less popular and less prolific PC OSes, such as OS/X, Open Solaris, and Linux distributions.

In addition to being free for commercial use, the Android OS has one of the largest, wealthiest, and most innovative companies in modern-day computing currently behind it: Google. Add in the OHA, and you have more than a trillion dollars of mega-brand companies behind you, supporting your app development efforts. It certainly seems too good to be true; however, it's a fact: if you're an Android developer (which you are about to be, in about a dozen or so chapters), then you now have a supreme hardware and software sales and support team behind your new media content development business.

Finally, and most important, it's much easier to get your Android applications **published** than it is with those other platforms that are similar to Android 7.1.1 (I won't mention names here, to protect the not-so-innocent). We all know we would rather spend our time on applications development than on trying to get our Android apps approved for sale!

We've all heard those horror stories regarding major development companies waiting months, and sometimes years, for their apps to be approved for other app marketplaces. These problems are nearly non-existent on the open source Android platform. Publishing an app in the Google Play store is as easy as paying \$50, uploading your Android .apk file, and specifying whether you are offering a free or a paid download. Let's take a look at what we are going to cover during this book, and at what we are not going to be covering. Finally, we'll look at the new Android 7.0 "Nougat" operating system features, before we finish up with this first chapter.

The Scope of This Book

This book is an introduction to the core features and attributes of Android, and to the work process for developing applications for Android. The book is intended for absolute beginners; that is, people who have never created an application on the Android platform for a consumer electronic device. If you are already familiar with Android, then this book is not appropriate for you. I do not assume that readers know what Java is; or how XML works; or what styles or themes are; or what a codec, alpha channel, color depth, dithering, or a blending mode algorithm is.

All I know is that by the end of this book, you're going to appear as if you are speaking a foreign language when you start talking about new media Android application development in front of friends, family, and clientele, which ultimately will get you hired, and hopefully, well paid. Be advised that it will take far more than one book to learn the "ins and outs" of Android, so be sure to check out all of the other Android and Pro Android titles at <http://www.apress.com>.

What Is Covered in This Book

This book covers the basic and essential elements of Android application development, including but not limited to the following areas:

- The open source software development tools required to develop for the Android 7.x platform
- Where to get this free software development environment, as well as professional new media content creation tools that can be used in conjunction with the Android Studio 2.3 IDE
- How to properly install and configure the necessary tools for Android 7 application development, as well as for new media content creation tools that can be used with Android Studio 2.3
- Which third-party tools are useful to use in conjunction with the Android Studio 2.3 IDE
- Which OSs and platforms currently support development for Android using these tools
- The concepts and programming constructs for Java and XML, and their practical applications in creating Android applications
- How Android Studio 2.3 goes about setting up an Android application
- How Android OS defines Android application user interface (UI) components using the View class
- How Android OS controls UI component layout on the display screen using the ViewGroup class
- How Android can communicate with other Android applications using the Intent class
- How Android apps interface with content providers, datastores, resources, networks, and the Internet
- How Android alerts users to events that may be taking place, inside or outside of an Android app
- How Android applications are defined for publishing using the app's Android manifest XML file
- How Android applications can use threads for background processing using the Service class