Android Programming Concepts

Trish Cornez
University of Redlands

Richard Cornez
University of Redlands
Dedication

For Bob
Contents

2.6.6 RatingBar 121
2.6.7 Spinner 121
2.7 Unique ID of a View Object and the R Class 123
2.8 The ViewGroup 125
2.8.1 RadioGroup 126
2.9 Adaptive Design Concepts—Screens and Orientations 142
2.10 TableRow and TableRow 153
2.11 Container Views 170
2.11.1 ListView, GridView, and ExpandableListView 171
2.11.2 ScrollView and HorizontalScrollView 171
2.11.3 SearchView 171
2.11.4 VideoView 172
2.12 Using an Adapter 172
Exercises 183

Chapter 3 Activities and Intents 185
3.1 Activity Lifecycle 185
3.2 Starting, Saving, and Restoring an Activity 191
3.3 Multiple Activities and the Intent Class 202
3.3.1 Explicit Intents 204
3.3.2 Implicit Intents 205
3.4 Handling Keyboard Visibility in an Activity 205
3.5 Passing Data between Activities 223
3.6 Basic Transitions between Activities 243
3.7 Scene Transitions 258
Exercises 276

Chapter 4 Fragments, ActionBar, and Menus 279
4.1 Fragmentation and Android Fragments 279
4.2 The Fragment Lifecycle 280
4.3 Action Bar 284
4.4 ActionBar Configurations 298
4.4.1 Overflow on the Action Bar 302
4.4.2 Adding an Action View 304
4.5 Responsive Design with Fragments 318
4.6 Animation in Fragment Transactions 338
4.7 ListViews and Adapters 352
4.8 Handling Click Events in a ListView 363
Exercises 371
## Chapter 5  Graphics, Drawing, and Audio  373
### 5.1 Graphics in Android 373
### 5.2 Adding and Manipulating ImageViews 374
### 5.3 Drawing and the Canvas Class 392
### 5.4 Recursive Drawing 402
### 5.5 Frame-by-Frame and Film Loop Animations 415
### 5.6 Animate Library 436
### 5.7 Audio 446
#### 5.7.1 SoundPool 447
#### 5.7.2 MediaPlayer 449
### Exercises 459

## Chapter 6  Threads, Handlers, and Programmatic Movement 461
### 6.1 Multithreading and Multicore Processing 461
### 6.2 Main Thread and Background Thread 464
### 6.3 Thread Approaches 465
#### 6.3.1 Implementing a Runnable Interface 465
#### 6.3.2 Extend the Thread Class 466
### 6.4 UI Modification and the Handler Class 466
### 6.5 Loopers 477
### 6.6 Canvas Movement and Views 490
#### 6.7 SurfaceViews 499
### 6.8 Efficient Threading 512
### 6.9 Materials and Functionality 525
### 6.10 AsyncTasks 541
### Exercises 553

## Chapter 7  Touch Gestures 555
### 7.1 Touchscreens 555
### 7.2 Touch Gestures 556
### 7.3 The Basics of Touch Events 557
### 7.4 Gesture Detector 559
### 7.5 The MotionEvent Class 570
### 7.6 The Drag-and-Drop Gesture 591
### 7.7 Fling Gesture 611
### 7.8 Fling Velocity 630
### 7.9 Multitouch Gestures 644
### Exercises 656
Chapter 8  Sensors and Camera  657
  8.1 Sensors and Mobile Devices 657
  8.2 Android Sensors 658
    8.2.1 Motion Sensors 658
    8.2.2 Environmental Sensors 660
    8.2.3 Positional Sensors 661
  8.3 Working with Sensors 661
  8.4 Coordinate System 666
  8.5 Accelerometer and Force 682
  8.6 Sensor Batching 694
    8.6.1 Step Counter and Step Detector 694
  8.7 Composite Sensors 696
  8.8 Camera 705
    8.8.1 The Built-in Camera Application 707
  8.9 Manipulating Photos 721
  Exercises 740

Chapter 9  File Storage, Shared Preferences, and SQLite  741
  9.1 Storing Data 741
  9.2 Shared Preferences 742
  9.3 File Storage—Internal and External Storage 744
  9.4 Android Database with SQLite 747
  9.5 SQLiteOpenHelper 751
  9.6 Adapters and AdapterViews 765
  Exercises 807

Index  809
Mobile device users experience their environments through a variety of computing screens. The devices most often used are computers, tablets, and phones. Users today increasingly expect a connected and highly personalized experience that is seamless across all connected devices, including television, home automation gadgets, wearable computers, and cars. Android is the operating system that powers many of these connected devices. As of early 2015, Android is the largest installed base of any mobile platform.

Initially created by Android Inc. by a team led by Andy Rubin, the Android operating system was acquired by Google in 2005. The first commercial version of Android was released in 2008 on an HTC phone named Dream; also known as the T-Mobile G1. Since its initial release, the operating system has undergone an extreme metamorphosis, evolving quickly and frequently, with new and updated versions released at an unprecedented rate.

As a Linux-based system, Android is run as an open source project; this means that anyone can adapt the code for his or her own purposes. This permissive model makes Android unique in that it allows companies and developers to modify and distribute the software freely. Device manufacturers creating phones and tablets often customize the Android operating system to the specific needs of their particular mobile devices.

In this text, readers will learn how to design and implement applications that will run on a variety of Android-driven devices. Building sophisticated applications that are optimized, responsive, and able to perform complex interactions at fast speeds requires patience, skill, and practice. The concepts and techniques you will learn in this text will provide you with the building blocks needed to master the art of mobile programming.

**Text Objectives**

This text was conceived with two types of individuals in mind: programming students and professional software developers who wish to broaden their expertise. It is essential that readers know how to program in an OOP language, preferably Java,
Preface

before using this text. For non-Java programmers, familiarization with the Java API is recommended.

This text is intended as a textbook, not as a tutorial. We have designed the text, using an easy-to-understand and straightforward approach, to integrate key concepts relating to application development that students see daily on Android devices. Each chapter presents Android concepts and methodologies with complete abbreviated application examples that are relevant to current platforms.

How to Use This Text

The first three chapters provide an introduction to the foundation of application development. Chapter 1 incorporates two step-by-step tutorials to help readers get started in creating basic applications. Chapters 2 and 3 provide key core concepts for building well-designed applications. It is important that readers are comfortable with these early chapters before proceeding.

After reading the first three chapters, Chapters 4 through 9 do not need to be read in sequential order. Readers wishing to acquire the most invaluable concepts first should start with Chapter 4 and proceed to Chapters 7, 8, and 9; however, multithreading concepts (discussed in Chapter 6) are a prerequisite for the last three chapters. A detailed reading of Chapter 5 is not required for Chapter 6.

Instructor and Student Resource Material

The following ancillary materials are available on the text website:

go.jblearning.com/CornezAndroid

• Source code files for lab examples
• Instructor’s Manual containing solutions to end-of-chapter exercises
• Lecture Slides in PowerPoint format
• Test bank

Acknowledgments

We have received invaluable support from friends, students, and colleagues in the preparation of the text. The University of Redlands has provided the resources and means for us to complete the project. Jones & Bartlett Learning offered an excellent team of professionals who handled the book from manuscript to final production. We especially want to thank Laura Pagluica, Taylor Ferracane, Sara Kelly, and Abby Reip.
We are thankful to Jordan Vega and Sam Marrs for many useful suggestions. We are indebted to Jim Bentley and Pani Chakrapani for giving us the opportunity to schedule several mobile programming classes that allowed us to classroom-test portions of our text. Jim Bentley was kind enough to offer department funds for devices for some of our students, who might not otherwise have been able to participate.

We would like to thank the following reviewers, who offered us indispensable pedagogical and content guidance for revision:

Sonia Arteaga  
Hartnell College

Jeremy Blum, DSc  
Associate Professor of Computer Science  
Penn State Harrisburg

Georgia Brown, MS  
Instructor  
Northern Illinois University

George Dudas  
Instructor in Computer Science and Software Engineering  
Penn State Erie, The Behrend College

Shane Schartz  
Informatics  
Fort Hays State University

Robert Steinhoff, PhD  
Florida Memorial University

Michael Ziray  
Boise State University

Last but not least, thanks to our many students whose struggles, challenges, and successes gave us all the evidence we needed to improve the text.