

Hervé J. Franceschi
Loyola University Maryland

ANDROID

APP DEVELOPMENT



JONES & BARTLETT
LEARNING



World Headquarters
Jones & Bartlett Learning
5 Wall Street
Burlington, MA 01803
978-443-5000
info@jblearning.com
www.jblearning.com

Jones & Bartlett Learning books and products are available through most bookstores and online booksellers. To contact Jones & Bartlett Learning directly, call 800-832-0034, fax 978-443-8000, or visit our website, www.jblearning.com.

Substantial discounts on bulk quantities of Jones & Bartlett Learning publications are available to corporations, professional associations, and other qualified organizations. For details and specific discount information, contact the special sales department at Jones & Bartlett Learning via the above contact information or send an email to specialsales@jblearning.com.

Copyright © 2018 by Jones & Bartlett Learning, LLC, an Ascend Learning Company

All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

The content, statements, views, and opinions herein are the sole expression of the respective authors and not that of Jones & Bartlett Learning, LLC. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply its endorsement or recommendation by Jones & Bartlett Learning, LLC, and such reference shall not be used for advertising or product endorsement purposes. All trademarks displayed are the trademarks of the parties noted herein. *Android App Development* is an independent publication and has not been authorized, sponsored, or otherwise approved by the owners of the trademarks or service marks referenced in this product.

There may be images in this book that feature models; these models do not necessarily endorse, represent, or participate in the activities represented in the images. Any screenshots in this product are for educational and instructive purposes only. Any individuals and scenarios featured in the case studies throughout this product may be real or fictitious, but are used for instructional purposes only.

09365-0

Production Credits

VP, Executive Publisher: David D. Cella
Executive Editor: Matt Kane
Acquisitions Editor: Laura Pagluica
Associate Editor: Taylor Ferracane
Associate Production Editor: Rebekah Linga
Director of Marketing: Andrea DeFronzo
Marketing Manager: Amy Langlais
VP, Manufacturing and Inventory Control: Therese Connell

Composition: S4Carlisle Publishing Services
Cover Design: Kristin E. Parker
Rights & Media Specialist: Merideth Tumas
Media Development Editor: Shannon Sheehan
Cover Image: © Fon_nongkran/Shutterstock
Printing and Binding: RR Donnelley
Cover Printing: RR Donnelley

Library of Congress Cataloging-in-Publication Data

Names: Franceschi, Hervé, author.

Title: Android app development / Herve J. Franceschi, Loyola University Maryland.

Description: Burlington, Massachusetts : Jones & Bartlett Learning, [2017] | Includes bibliographical references and index.

Identifiers: LCCN 2016038890 | ISBN 9781284092127

Subjects: LCSH: Android (Electronic resource) | Mobile computing. | Application software—Development.

Classification: LCC QA76.76.A65 F73 2017 | DDC 005.3—dc23

LC record available at <https://lcn.loc.gov/2016038890>

6048

Printed in the United States of America

20 19 18 17 16 10 9 8 7 6 5 4 3 2 1



DEDICATION

To Kristin, H  l  na, and Louis

TABLE OF CONTENTS

Preface	ix
1 Basics of Android, First App: HelloAndroid	1
1.1 Smartphones and Their Operating Systems	2
1.1.1 Smartphones	2
1.1.2 Android Phones	2
1.1.3 Apps and Google Play	2
1.2 Development Environment for Android Apps	3
1.3 Our First App: HelloAndroid	3
1.3.1 Skeleton App	3
1.3.2 GUI Preview	8
1.3.3 XML Files: activity_main.xml, colors.xml, styles.xml, strings.xml, dimens.xml	8
1.3.4 The MainActivity Class	16
1.4 Running the App Inside the Emulator	18
1.5 Debugging the App with Logcat	24
1.6 Using the Debugger	27
1.7 Testing the App on an Actual Device	28
1.8 The App Manifest and the Gradle Build System	30
1.8.1 The AndroidManifest.xml File: App Icon and Device Orientation	30
1.8.2 The Gradle Build System	34
2 Model View Controller, GUI Components, Events	39
2.1 Model-View-Controller Architecture	40
2.2 The Model	41
2.3 GUI Components	41
2.4 RelativeLayout, TextView, EditText, and Button: Tip Calculator, Version 0	44
2.5 GUI Components and More XML Attributes: Tip Calculator, Version 1	53
2.6 Styles and Themes: Tip Calculator, Version 2	62
2.7 Events and Simple Event Handling: Coding the Controller, Tip Calculator, Version 3	69
2.8 More Event Handling: Tip Calculator, Version 4	74
3 Coding the GUI Programmatically, Layout Managers	87
3.1 Model View Controller Architecture	88
3.2 The Model	88
3.3 Creating the GUI Programmatically, TicTacToe, Version 0	91
3.4 Event Handling: TicTacToe, Version 1	96
3.5 Integrating the Model to Enable Game Play: TicTacToe, Version 2	100
3.6 Inner Classes	102
3.7 Layout Parameters: TicTacToe, Version 3	105

3.8	Alert Dialogs: TicTacToe, Version 4	110
3.9	Splitting the View and the Controller: TicTacToe, Version 5	115
4	Multiple Activities, Passing Data between Activities, Transitions, Persistent Data	127
4.1	Model: The Mortgage Class	128
4.2	Using a <code>TableLayout</code> for the Front Screen GUI: Mortgage Calculator App, Version 0	131
4.3	Using a <code>RelativeLayout</code> for the Second Screen GUI.	135
4.4	Connecting Two Activities: Mortgage Calculator App, Version 1	140
4.5	The Life Cycle of an Activity	145
4.6	Sharing Data between Activities: Mortgage Calculator App, Version 2	150
4.7	Transitions between Activities: Mortgage Calculator App, Version 3	155
4.8	Handling Persistent Data: Mortgage Calculator App, Version 4	161
5	Menus, SQLite	173
5.1	Menus and Menu Items: Candy Store App, Version 0	174
5.2	Icon Items, Candy Store App, Version 1	180
5.3	SQLite: Creating a Database, a Table, and Inserting Data: Candy Store App, Version 2	188
5.4	Deleting Data: Candy Store App, Version 3	197
5.5	Updating Data: Candy Store App, Version 4	202
5.6	Running the Cash Register: Candy Store App, Version 5	208
6	Managing the Device Orientation	221
6.1	The Configuration Class	222
6.2	Capturing a Device Rotation Event	227
6.3	Strategy 1: One Layout XML File per Orientation.	229
6.4	Strategy 2: One Layout XML File for Both Orientations, Modify the Layout by Code.	233
6.5	Strategy 3: Manage Layout and Orientation 100% by Code	236
7	Touches and Swipes	249
7.1	Detecting a Touch Event	251
7.2	Handling a Swipe Event: Moving a <code>TextView</code>	256
7.3	The Model.	259
7.4	The View: Setting Up the GUI, Puzzle App, Version 0.	261
7.5	Moving the Pieces, Puzzle App, Version 1	265
7.6	Solving the Puzzle, Puzzle App, Version 2	270
7.7	Gesture and Tap Detection and Handling	277
7.8	Detecting a Double Tap, Puzzle App, Version 3	284
7.9	Making the App Device Independent, Puzzle App, Version 4	289
8	Graphics, Animations, Sounds, and Gaming	297
8.1	Graphics	298
8.2	Making a Custom View, Drawing, Duck Hunting App, Version 0.	302
8.3	The Model.	308
8.4	Animating an Object: Flying the Duck, Duck Hunting App, Version 1	313
8.5	Handling Touch Events: Moving the Cannon and Shooting, Duck Hunting App, Version 2	320
8.6	Playing a Sound: Shooting, Collision Detection, Duck Hunting App, Version 3	324

9	Fragments	337
9.1	The Model.	338
9.2	Fragments.	340
9.3	Defining and Adding a Fragment to an Activity Using a Layout XML File, Hangman, App Version 0	341
9.4	Adding GUI Components, Styles, Strings, and Colors, Hangman, App Version 1.	346
9.5	Defining a Fragment Using a Layout XML File and Adding the Fragment to an Activity by Code, Hangman, App Version 2.	352
9.6	Defining and Adding a Fragment to an Activity by Code, Hangman, App Version 3.	359
9.7	Communication between Fragments and Their Activity: Enabling Play, Hangman, App Version 4	362
9.8	Using an Invisible Fragment, Hangman, App Version 5	365
9.9	Making a Fragment Reusable, Hangman, App Version 6.	367
9.10	Improving the GUI: Processing the Keyboard Input Directly, Hangman, App Version 7.	371
10	Using Libraries and Their APIs: Speech Recognition and Maps	383
10.1	Voice Recognition	384
10.2	Voice Recognition Part A, App Version 0.	387
10.3	Using the Google Maps Activity Template, App Version 1	391
10.4	Adding Annotations to the Map, App Version 2.	400
10.5	The Model.	403
10.6	Displaying a Map Based on Speech Input, App Version 3.	406
10.7	Controlling Speech Input, App Version 4.	414
10.8	Voice Recognition Part B, Moving the Map with Voice Once, App Version 5	418
10.9	Voice Recognition Part C, Moving the Map with Voice Continuously, App Version 6.	425
11	Using the GPS and Location Services	433
11.1	Accessing Google Play Services, GPS App, Version 0	434
11.2	Using the GPS to Retrieve Our Location, GPS App, Version 1.	440
11.3	Model for Distance and Time to Destination	445
11.4	Distance and Time to Destination, GPS App, Version 2.	447
11.5	Updating Distance and Time to Destination, GPS App, Version 3	453
12	Using Another App within the App: Taking a Photo, Graying It, and Sending an Email	463
12.1	Accessing the Camera App and Taking a Picture, Photo App, Version 0.	464
12.2	The Model: Graying the Picture, Photo App, Version 1	471
12.3	Defining Shades of Gray Using SeekBars, Photo App, Version 2.	477
12.4	Improving the User Interface, Photo App, Version 3.	485
12.5	Storing the Picture, Photo App, Version 4.	490
12.6	Using the Email App: Sending the Grayed Picture to a Friend, Photo App, Version 5.	499
13	XML and Content Apps	509
13.1	Parsing XML, DOM, and SAX Parsers, Web Content App, Version 0	510
13.2	Parsing XML into a List, Web Content App, Version 1	518

13.3 Parsing a Remote XML Document, Web Content App, Version 2 521

13.4 Displaying the Results in a `ListView`, Web Content App, Version 3 526

13.5 Opening a Web Browser Inside the App, Web Content App, Version 4 531

14 Making an Android Widget 539

14.1 Steps in Making a Widget, Temperature Widget, Version 0 540

14.2 Styling the Widget, Temperature Widget, Version 1 549

14.3 Updating the Data in the Widget, Temperature Widget, Version 2 552

14.4 Updating the Data in the Widget by Clicking on It, Temperature Widget, Version 3 554

14.5 Retrieving the Temperature Data from a Remote Source, Temperature Widget, Version 4 557

14.6 Using an Activity to Customize the Widget, Temperature Widget, Version 5 570

14.7 Hosting the Widget in the Lock Screen, Temperature Widget, Version 6 578

15 In App Advertising 587

15.1 The View, Stopwatch App, Version 0 588

15.2 The Controller: Running the Stopwatch App, Version 1 593

15.3 Improving the Stopwatch App, Version 2 596

15.4 Placing the Ad, Stopwatch App, Version 3 598

15.5 Placing the Ad in a Fragment, Stopwatch App, Version 4 607

15.6 Managing the Life Cycle of the `AdView`, Stopwatch App, Version 5 610

16 Security and Encryption 617

16.1 Symmetric and Asymmetric Encryption 618

16.2 Symmetric Encryption: The Model (AES), Encryption App, Version 0 619

16.3 Symmetric Encryption: Adding a View, Encryption App, Version 1 625

16.4 Asymmetric Encryption: Adding RSA to the Model, Encryption App, Version 2 630

16.5 Symmetric and Asymmetric Encryption: Modifying the View, Encryption App, Version 3 634

APPENDIX A Retrieving the Height of the Status Bar and the Action Bar Dynamically.643

APPENDIX B Setting the Font Size of a `TextView` Dynamically. 649

APPENDIX C How to Download and Install Google Play Services, Use Maps.655

APPENDIX D The `AsyncTask` Class. 665

INDEX. 673



PREFACE

Purpose of This Text and Its Audience

Android App Development covers the essential topics of Android Development: XML resources, including styles, XML defined GUIs, programmatically defined GUI, event-driven programming, activity life cycle, how to manage several activities, transitions between activities, persistent data (including SQLite), how to manage orientation, fragments, how to make a device-independent app, and how to use an app within another app. It covers “fun” topics such as touches, swipes, graphics, sounds, and game programming. Finally, it also covers more specialized topics of interest: maps, voice recognition, how to make a content app including remote data retrieval and XML parsing, GPS, how to make a widget, in-app advertising, and encryption.

This course ideally follows a Java course. Although it is preferable that students have been exposed to event handling programming prior to taking an Android course, it is not a requirement. GUI programming and event handling are thoroughly explained.

Coverage and Approach

The approach in this book is to use an app throughout each chapter to illustrate the concepts presented. The app typically uses the Model View Controller architecture. It is constructed in a very progressive manner starting with version 0, then version 1, and so on. We progress through the app versions, adding concepts and topics as needed by the app. Thus, each app is manageable in the classroom. The instructor can choose to download the Model and explain it briefly rather than cover it in detail. The Model classes use straightforward Java code and do not need much explanation. We also try to use apps that are simple yet meaningful and that the students can easily understand and relate to. It is possible to only cover the early versions of an app if the instructor does not want to cover some of the chapter’s topics. Furthermore, many chapters are independent of each other and do not have to be covered in the sequence in which they are presented in this book.

Pedagogy

Within each chapter, and therefore within each app, the instructor can pick what topics he or she chooses to cover and in what depth as well as at what pace. The Model, which typically uses basic Java classes and is Android independent, can be downloaded and briefly explained. That allows the instructor to concentrate on the View and the Controller, which include Android-specific topics. Screenshots, examples, and tables are included throughout each chapter to illustrate the

concepts at hand and the current state of the chapter app. Software Engineering and Common Error boxes highlight software engineering tips and common errors throughout the chapters. Chapter summaries are provided at the end of each chapter followed by exercises, problems, and projects to test students' knowledge. Each chapter includes extensive multiple-choice, fill in the code, and write an app exercises.

SOFTWARE ENGINEERING: When defining an `EditText`, choose the appropriate value for `android:inputType` in order to match the data we expect the user will input.

- ▶ Model: `TipCalculator` class
- ▶ View: `activity_main.xml`
- ▶ Controller: `MainActivity` class

COMMON ERROR: We need to add an `activity` element in the Android Manifest.xml file whenever we add an activity to our app. Otherwise, the app will crash when we try to go to that activity.

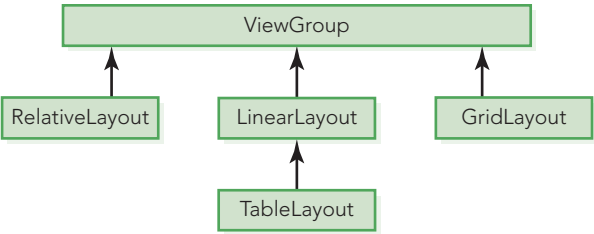


FIGURE 2.2 Selected layout manager classes

TABLE 2.6 The <code>android:textSize</code> and <code>android:hint</code> XML attributes of <code>TextView</code>		
XML Attribute	Method	Description
<code>android:textSize</code>	<code>void setTextSize(int unit, float size)</code>	Sets the size of the text
<code>android:hint</code>	<code>void setHint(int resource)</code>	Sets the hint text to display

Chapter-by-Chapter Overview

The first nine chapters cover essential concepts of Android app development. The last seven chapters cover more specific topics.

The appendices include mini-chapters on concepts that are used in several chapters: retrieving the dimensions of a device, including the status and action bars, sizing the font of a TextView dynamically, performing a task in the background using AsyncTask (to retrieve data from a remote location), and using Google Play Services. This helps make most of the chapters independent so that the instructor can pick and choose which chapters to cover.

A lot of the fundamental concepts are covered in Chapters 1, 2, 3, and 4, so we recommend instructors cover those chapters early and in that order. Chapter 2 covers the concepts of XML resources, including strings, colors, and styles extensively, as well as event driven programming. These concepts are reused throughout the book. Chapter 3 is the first chapter in which we build the GUI programmatically. We also do so in Chapters 5, 7, and 13. Thus, it is better to cover Chapter 3 early. Chapter 4 is the first chapter that involves multiple activities. Chapters 5, 10, 12, and 13 also involve multiple activities. Thus, Chapter 4 should also be covered early. Some of the material in Chapter 7 (touches and gestures) is used in Chapter 8 (graphics, animation, and sound), so they should be covered in this order.

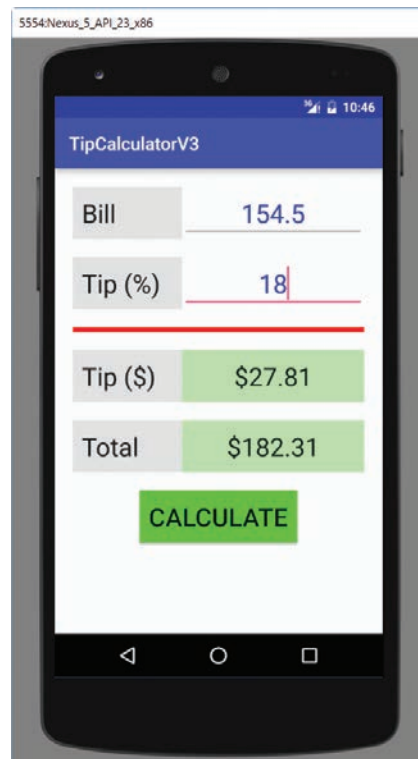
The following is a summary of the topics covered in each chapter:

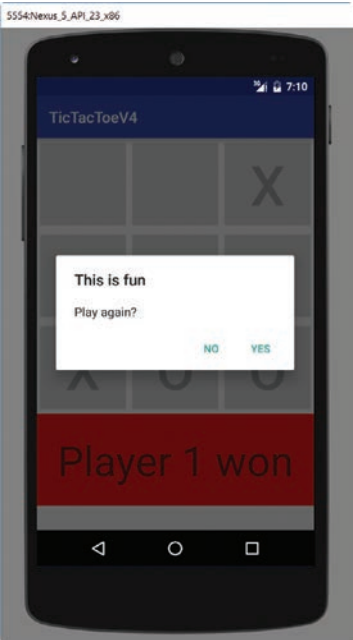
Chapter 1: Basics of Android, First App: Hello World

We look at how to use the Android Studio development environment: how to use the emulator, how to run the app on a device, how to output to Logcat, using the debugger. We also look at the various resources associated with an app: the AndroidManifest.xml file, the various xml files to define string, color, dimension resources, and the xml file defining the GUI.

Chapter 2: Model View Controller, GUI Components, Events

We explain the Model View Controller architecture and make our first app, a tip calculator, using the MVC. We define a GUI using a RelativeLayout and TextViews, EditTexts and a Button. We explain how to separate the various styles used in an app from the contents of the app (similar to CSS in web design). We cover event handling: clicking on a button and typing on a soft keyboard.





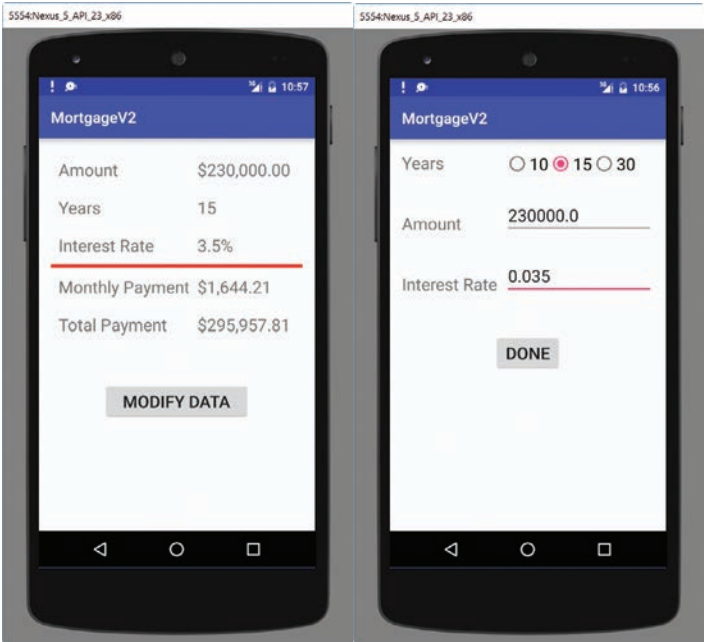
Chapter 3: Coding the GUI Programmatically, Layout Managers

Continuing with the MVC architecture, we explain how to define a GUI and set up event handling programmatically for a tic-tac-toe app. We explain how to use inner classes, layout parameters, and alert dialogs.

Chapter 4: Multiple Activities, Passing Data between Activities, Transitions, Persistent Data

We cover how to include multiple activities within an app, and explain how to pass data between activities. We explore the life-cycle methods of an activity and how they can be used, particularly in the context of an app with multiple activities. We demonstrate how to set up an animated transition between one activity and another. We explore the RelativeLayout class in greater depth as well as the TableLayout class. Finally, we show

how to deal with persistent data. We use a mortgage app as the vehicle for the concepts presented: the first screen displays mortgage data, including the monthly payment, and the second screen enables the user to edit the mortgage characteristics (amount, interest rate, and number of years).



Chapter 5: Menus, SQLite

We show how to include a menu in an app and how to use SQLite to handle persistent data. We use a ScrollView when generating a GUI programmatically. The app is a candy store manager that enables the user to add, update, and delete candies, as well as ring the cash register for a customer using the Toast class.

Chapter 6: Managing the Device Orientation

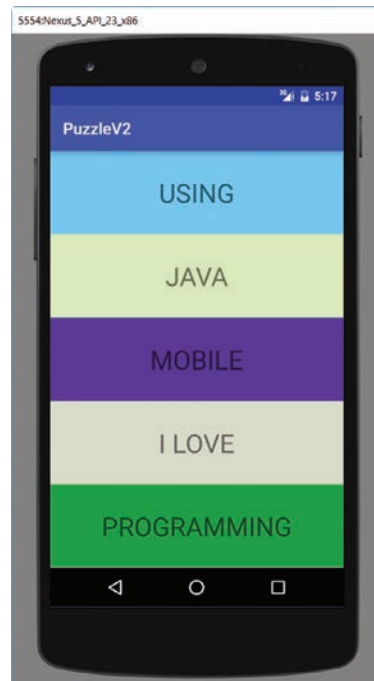
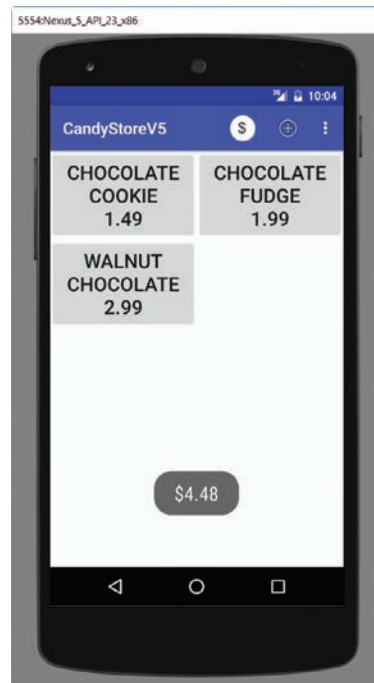
In this chapter, we show how to detect a change in the device orientation and explore various ways of coding the GUI so that an app works in both vertical and horizontal orientations. In particular, we dynamically retrieve the dimensions of the device so that we can size and position the GUI components accordingly. There is no meaningful app in this chapter, we simply explore the various ways of managing the GUI with orientation changes.

Chapter 7: Touches and Swipes

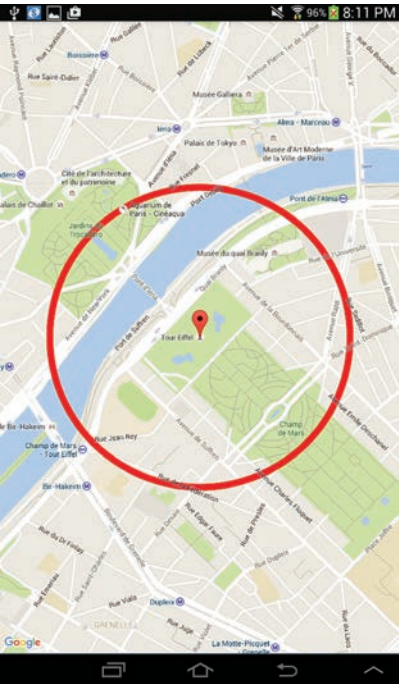
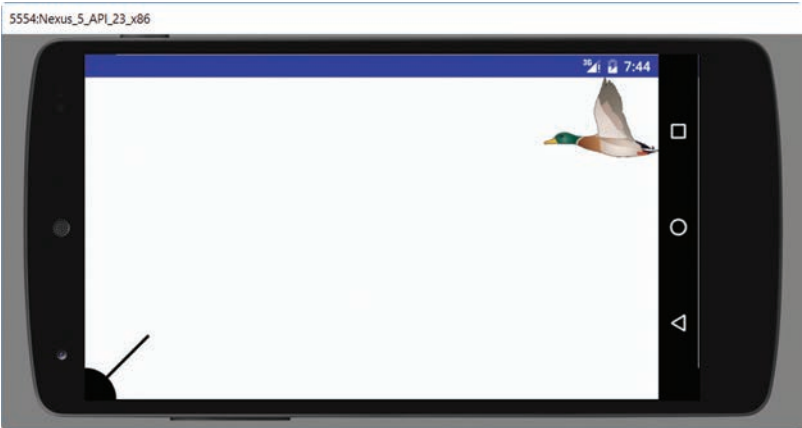
We first show how to detect and handle a touch or a swipe event. More generally, we also show how to detect and handle a gesture, single tap, or double tap event. We use touch and gesture handling to build a puzzle game app: the user can solve a puzzle by touching and moving the puzzle pieces. The puzzle colors are generated randomly and feedback is given to the user when he or she solves the puzzle.

Chapter 8: Graphics, Animations, Sounds, and Gaming

We explore how to create a custom View, how to draw shapes and bitmaps, how to create a game by animating objects on the screen, handling events, and making sounds. We use the Timer and TimerTask classes to create a game loop and update the state of the game at a defined frequency. In this duck hunting app, a duck flies from the right of the screen in an animated fashion, using



four png frames. The user can control a cannon, a drawn shape, located at the bottom left of the screen by moving its barrel with touches. A double tap fires a bullet, a drawn shape, and we detect collision with the duck. If the duck is shot, a sound is played, we stop animating the duck, and it falls down vertically.



Chapter 9: Fragments

We cover various ways of using fragments within an activity, communication between two fragments, and how to make a fragment reusable. The app is a hangman game where the various fragments include the state of the game, the status of the game, the number of tries left, as well as an invisible fragment.

Chapter 10: Using Libraries and Their APIs: Speech Recognition and Maps

We use the Google maps activity template to display a map with annotations, and use speech recognition to change the map. We also explain how to use Google Play Services. In this app, the user can display a map centered on a city (London, Paris, Rome, or Washington, DC) by saying its name, and then move the map by saying north, south, west, or east.

Chapter 11: Using the GPS and Location Services

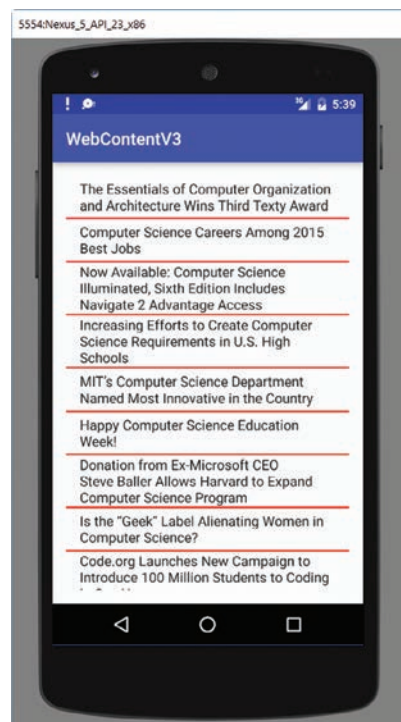
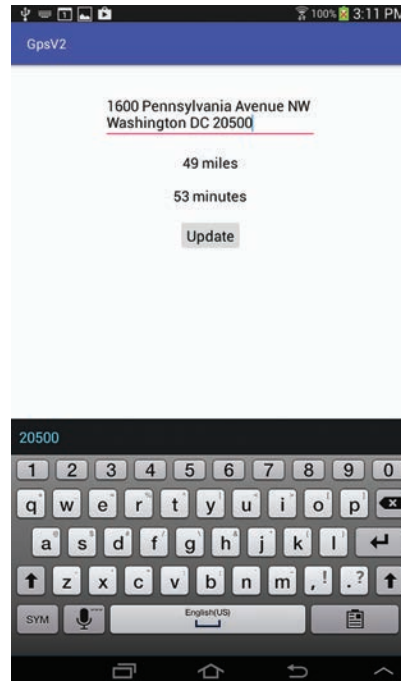
We explain how to use Google Play Services, and how to use a device's GPS within an app. The app manages the distance and time left to a destination, first enabling the user to update them, and eventually updating them both periodically and automatically.

Chapter 12: Using Another App within the App: Taking a Photo, Graying It, and Sending an Email

We explain how to use another app, such as a camera app, within an app, how to store files in external storage, and how to use an email app within an app. We also explore seek bars (sliders). The app opens a camera app and when the user takes a picture, enables the user to gray the picture using three seek bars. In the Model, we explain how to access each individual pixel of a Bitmap, both for reading and writing. After the user is done graying the image, the user can send an email to a friend and we automatically attach the grayed picture.

Chapter 13: XML and Content Apps

We parse local and remote XML documents using a SAX parser. We show how we can use the AsyncTask class to perform a task in the background of an app. We display results in a ListView and open a web browser inside the app. The app is a content app that retrieves data from a remote URL (<http://blogs.jblearning.com/computer-science/feed/>), parses its XML content into an ArrayList of items, and displays the items in a ListView. When the user clicks on one of the items, we open the browser at the URL corresponding to that item.



Chapter 14: Making an Android Widget

We cover the various steps to make a widget: we first make a very simple widget, then make it dynamic, then retrieve temperature data from a remoter website, and finally make the widget customizable by the user. We start with hard coded city and temperature, then give the widget a style, then add some dynamic data retrieving the date and time from the device, and enable the user to update the widget data by clicking on it. Next, we retrieve the temperature data from a remote website using the AsyncTask class. The data comes as a JSON string so we access the data we want within the JSON string (the temperature) and display it inside the widget. Finally, we make the widget customizable by enabling the user to set the city and state for which we retrieve the temperature data.

Chapter 15: In App Advertising

We place a Google ad in a simple app and explain the life-cycle methods of the AdView class. We also cover how to create a drawable resource and use it with a button. In this stopwatch app, we create three drawable resources (stop.xml, reset.xml, and start.xml) that we use as buttons for the app. We display a banner ad at the bottom of the screen and shows how we can retrieve a fake remote Google ad to test the app as well as how to use a live Google ad. There is no need to be a registered Google developer to ask for fake ads served by Google.



Chapter 16: Security and Encryption

We explain symmetric and asymmetric encryption, and how to use an encryption algorithm (AES) and an asymmetric encryption algorithm (RSA). There is no meaningful app in this chapter; the main objectives are to explain the various encryption schemes and algorithms, as well as how to use the classes and methods of the Java library that relate to encryption.

If students progress through all sixteen chapters, then they will have created the following apps:

- Hello Android (Chapter 1)
- Tip Calculator (Chapter 2)
- TicTacToe (Chapter 3)
- Mortgage Calculator (Chapter 4)
- Candy Store Manager (Chapter 5)
- Puzzle Game (Chapter 7)
- Duck Hunting Game (Chapter 8)
- Hangman Game (Chapter 9)

Map Display Using Speech Input (Chapter 10)
Travel App Using the GPS (Chapter 11)
Photo Graying (Chapter 12)
Content App (Chapter 13)
Temperature Widget (Chapter 14)
StopWatch with Advertising (Chapter 15)

Appendices

The appendices include the following:

- ▶ A micro-chapter on how to retrieve the height of the action bar and the status bar dynamically (including a micro-app for it): this is important if we intend to code the GUI programmatically and make the app device independent.
- ▶ A micro-chapter on how to size the font inside a TextView dynamically so that it fits perfectly on one line (including a micro-app for it): this is important if we display dynamic data whose length can vary significantly.
- ▶ How to use Google Play Services: this is important for apps involving maps, GPS, ads, and other services from Google.
- ▶ A micro-chapter on how to use the AsyncTask class (including a micro-app for it): this is important for apps that retrieve data from a remote location.

Instructor and Student Resources

Instructor and student resources are available for download at go.jblearning.com/FranceschiAndroid. These include:

- ▶ Access to the latest version of Android Studio
- ▶ Source code for all apps
- ▶ Answers to end of chapter multiple-choice and fill in the code questions
- ▶ Slides in PowerPoint format
- ▶ Test bank including sample midterm and final exams with mini-apps

Contacting the Author

If errors are found in this text, please contact the author at hjfranceschi@loyola.edu. Errata will be posted on the Jones & Bartlett catalog page at go.jblearning.com/FranceschiAndroid.

Acknowledgments

We would like to acknowledge the contributions of many partners, colleagues, and family members to this book.

First and foremost, we would like to thank our publisher, Jones & Bartlett Learning, especially Laura Pagluica, Acquisitions Editor, Taylor Ferracane, Associate Editor, Bharathi Sanjeev, Project Manager, and Rebekah Linga, Associate Production Editor.

Second, we extend our thanks to the reviewers:

Timothy E. Roden, PhD
Associate Professor of Computer Science
Lamar University, Texas

Jeremy Blum, DSc
Associate Professor of Computer Science
Pennsylvania State University, Harrisburg

Jamie Pinchot, DSc
Associate Professor of Computer and Information Systems
Robert Morris University

Allan M. Hart, PhD
Assistant Professor
Computer Information Science
Minnesota State University, Mankato

Dr. Marwan Shaban
Adjunct Professor
University of Central Florida

Georgia Brown, MS
Computer Science Department
Northern Illinois University

Roy Kravitz
Portland State University

Sonia M. Arteaga, PhD
Hartnell College

Dr. Shane Schartz
Assistant Professor of MIS, Informatics
Fort Hays State University

Finally, we recognize the support of my family, my wife Kristin, my daughter H  lena, and my son Louis, who provided support, feedback, and advice.