

Extracted from:

Rapid Android Development

Build Rich, Sensor-Based Applications with Processing

This PDF file contains pages extracted from *Rapid Android Development*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2013 The Pragmatic Programmers, LLC.

All rights reserved.

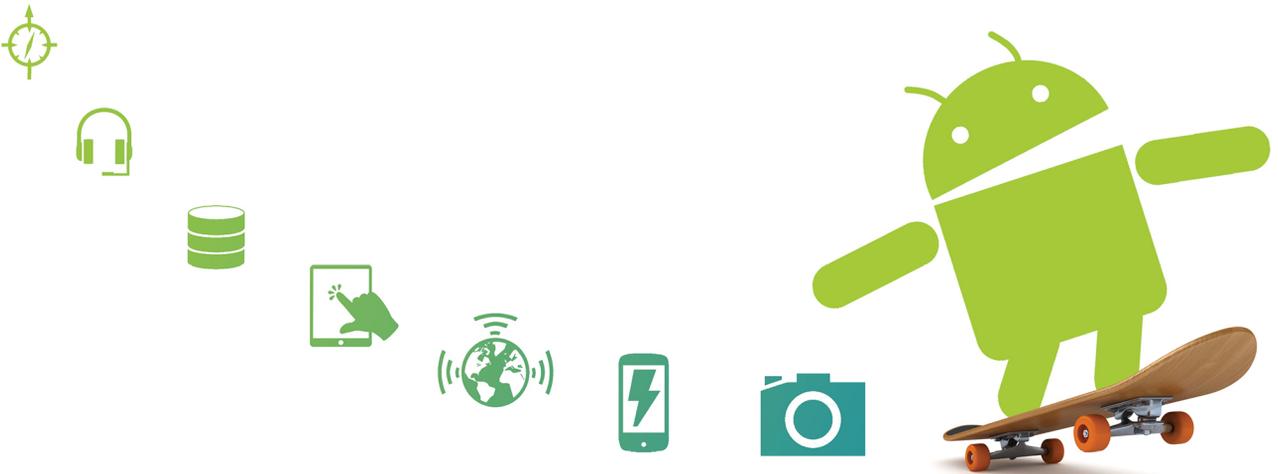
No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

Rapid Android Development

Build Rich, Sensor-Based
Applications with Processing



Daniel Sauter

Foreword by Jer Thorp, Co-founder,
The Office for Creative Research

Rapid Android Development

Build Rich, Sensor-Based Applications with Processing

Daniel Sauter

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at <https://pragprog.com>.

The Android robot is reproduced from work created and shared by Google and is used according to terms described in the Creative Commons 3.0 Attribution License (<http://creativecommons.org/licenses/by/3.0/us/legalcode>).

The team that produced this book includes:

John Osborn (editor)
Potomac Indexing, LLC (indexer)
Molly McBeath (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Ellie Callahan (support)

For international rights, please contact rights@pragprog.com.

Copyright © 2013 The Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-93778-506-2
Encoded using the finest acid-free high-entropy binary digits.
Book version: P3.0—May 2015

Preface

Processing is a favorite among artists and designers and widely popular among developers who look for a productivity edge.¹ The programming language and environment has developed from a sketching tool to a production environment for a range of operating systems and platforms. The *Android mode*, introduced to Processing with the release of version 2.0, now makes it as easy to develop Processing apps for the Android as for the desktop.

Initiators Ben Fry and Casey Reas have promoted software literacy since 2001 using Processing, a free open source tool that can be used by individuals at any level of programming experience. The Processing project thrives on the support of its generous online community, whose members encourage collaboration and code sharing and are responsible for one of Processing's most important features: its libraries.

Libraries have made Processing the versatile and capable coding environment that it is today. Members have contributed more than 130 libraries to Processing over the last decade. I have extensively used Processing in the classroom during the past eight years and have realized various projects with it, sometimes in conjunction with Processing's sister project, Arduino.² In 2010, I started the *Ketai* (the Japanese term for cell phone culture) library with Jesus Duran,³ which brings Android hardware features to Processing and makes it possible to work with sensors and hardware devices using simple and easy-to-read code.

In comparison, developing Android apps in Java using the standard Eclipse IDE entails a much steeper learning curve,⁴ one that requires a programmer to master both the syntax of a modern object-oriented language and the features of a complex development environment. In Processing, we can see results

-
1. <http://processing.org/>
 2. <http://arduino.cc/>
 3. <http://code.google.com/p/ketai/>
 4. <http://developer.android.com/sdk/eclipse-adt.html>

immediately because we are working with a straightforward syntax and a wide range of libraries and tools designed specifically to support visually lavish and highly interactive applications.

Android users expect a rich, interactive mobile user experience from their phones and tablets, one that takes full advantage of their touch screens, networking hardware, sensors for geolocation and device orientation, built-in cameras, and more. In this book, we'll learn how to create apps for Android devices that take full advantage of their many built-in hardware affordances.

Introducing Processing for the Android

Android is based on the Java programming language. Processing is also based on Java, making it the perfect platform for developing Android apps using Processing's straightforward syntax. The addition of the Android mode was a natural progression for Processing in order to streamline application development while targeting a broad range of operating systems and devices with one comprehensive programming language. In fact, Processing's software architecture allows us to mix in Java statements and packages, Android statements and packages, and Processing sketches and libraries wherever we feel like it.

This book focuses on Android apps developed in Processing. There are no differences between the Android mode and Processing's Java mode for desktop applications when it comes to the basic programming structure and syntax. Android-specific features that require device sensors and hardware are not available on the desktop and therefore are not usable in Processing's Java mode. They are available, however, as soon as we switch to Android mode.

In the last chapter of the book we'll discuss cross-platform challenges for mobile apps and introduce Processing's JavaScript mode. HTML5 web apps developed in Processing run on all modern browsers found on smart phones, tablets, and desktops today. While interoperability is an important factor, we will limit our discussion of web apps to the last chapter, as we can't access many of the hardware sensors and devices that make for exciting apps.

All core Processing methods are identical across modes, so when we develop Android apps we can also consider and use code examples written for Java mode. The Processing website contains a complete reference for all Processing methods.⁵ So does the IDE, which ships with a packaged reference that we

5. <http://processing.org/reference/>

can use without a connection to the Web; it's available from the Processing menu by selecting Help → Reference.

Let's take a look at some of the main advantages to developing Android apps in Processing:

- If you are new to programming, Processing for Android is much easier to learn than Java. If you are an experienced Java programmer already, Processing is a great programming environment for rapid prototyping of graphics and sensor-heavy apps.
- Processing uses straightforward syntax. In comparison to Java, it is more concise.⁶ Processing doesn't require you to understand advanced concepts such as classes or screen buffering to get started, yet it makes them accessible to any advanced users who want to use them. This makes Processing programs shorter and easier to read.
- The lightweight programming environment installs quickly and is easy to use. Processing is available for GNU/Linux, Mac OS X, and Windows. If you work with multiple computers or want to help someone else get started quickly, being up and running in a few minutes can make all the difference.
- Processing for Android supports OpenGL.⁷ When working with GPU-accelerated 2D and 3D graphics and geometry, lights, or textures, comprehensive OpenGL support is essential to ensure reasonably high frame rates and a fluid user experience.
- The latest version of Processing supports three application environments, or modes. Applications written in Java mode will run on Linux, Mac, or Windows systems. Programs written in Android mode will run on Android devices, and those written in JavaScript mode will run in any HTML5 browser. The Android mode is designed for creating native Android apps.
- Once your sketch prototyping is done, you can easily move your work to Eclipse for debugging and deployment. Processing lets you export your sketches as Android projects in the File → Export Android Project menu, creating an android directory with all the necessary files in it.

This list of advantages should provide you all the evidence you need to conclude that Processing is a great environment for developing Android apps. Your projects can scale in scope and context: from sketch to prototype and

6. http://wiki.processing.org/w/Java_Comparison

7. <https://www.opengl.org>

from prototype to market-ready application, from CPU-focused graphics rendering to hardware-accelerated GPU-focused rendering, from Processing statements and libraries to Android and Java statements and packages, and from a particular operating system and device to other operating systems and devices. You won't need to worry about a different last-minute route or an alternative solution for your software projects. Projects can grow with you and will let you enjoy the iterative process of design and development.

Who This Book Is For

The book is written for the following readers:

- *Readers with some programming experience:* Readers with a basic understanding of programming concepts can quickly learn the Processing language as they work their way through the examples. Processing is that easy to learn.
- *Intermediate Processing users:* Readers looking to create Android apps from within the Processing IDE can maintain a good balance between simplicity and versatility.
- *Educators who teach courses on mobile technologies:* Teachers often navigate the academic triangle of limited time, limited budget, and classes without prerequisites. This book brings advanced mobile features within the reach of students with little or no prior programming experience using a free tool that does not require developer licenses or subscriptions.
- *Java and Android developers:* Experienced developers look for a productivity gain. Because Processing builds on Java, developers can use their Java code and knowledge with Processing, leveraging a host of libraries for productivity gains.
- *JavaScript and web developers:* Processing.js syntax is identical to standard Processing syntax, making it easy to create JavaScript-powered web applications that can run inside browsers without plugins or other modifications. Processing.js also takes advantage of WebGL hardware acceleration.
- *Arduino users and hobbyists:* Some readers have experience with the Processing language by using it to program the Arduino electronics platform and are interested in adapting Android phones or tablets for use as sensing devices, controllers, or graphics processors.

Prerequisites

If you have never programmed in Processing or any other language before, you can turn to two excellent sources to get you up to speed; I've listed them at the end of this paragraph. You need to have an idea of the basic principles of programming to fully enjoy the book, such as the use of variables, conditionals, and loops. If you feel a little shaky with any of those concepts, I recommend you get one of the two books and keep it close by for frequent consultation. If you have scripted or programmed before, even if only at a basic level, you should be able to follow the examples in this book with a close read.

[Getting Started with Processing \[RF10\]](#)

This casual, inexpensive book is a concise introduction to Processing and interactive computer graphics.⁸ Written by Processing's initiators, it takes you through the learning process one step at a time to help you grasp core programming concepts.

[Processing: A Programming Handbook for Visual Designers and Artists, Second Edition \[RF11\]](#)

This book is an introduction to the ideas of computer programming within the context of the visual arts.⁹ It targets an audience of computer-savvy individuals who are interested in creating interactive and visual work through writing software but who have little or no prior experience.

What's in This Book

This book will have you developing interactive sensor-based Android apps in no time. The chapters include previews of all the classes and methods used for the chapter projects, as well as a description of the particular sensor or hardware device that we'll be working with. Small projects introduce the basic steps to get a particular feature working, which leads up to a more advanced chapter project.

Part I of the book gets you started with the touch screen, Android sensors, and cameras. [Chapter 1, Getting Started, on page ?](#), walks you through the steps of installing Processing and the Android SDK. We'll write a simple app and run it in the emulator and on an Android device. [Chapter 2, Working with the Touch Screen Display, on page ?](#), shows you how to use mouse position, finger pressure, and multitouch gestures on the touch screen panel

8. Available at <http://shop.oreilly.com/product/0636920000570.do>.

9. Available at <http://mitpress.mit.edu/books/processing-1>.

while also providing details on the support for color that Processing provides. [Chapter 3, *Using Motion and Position Sensors*, on page ?](#), introduces us to all the device sensors built into an Android. We'll display accelerometer values on the screen, build a motion-based color mixer, and detect a device shake.

In Part II, we'll be working with the camera and location devices found on most Androids. [Chapter 4, *Using Geolocation and Compass*, on page ?](#), shows us how to write location-based apps. We'll determine our location, the distance to a destination and to another mobile device on the move, and calculate the speed and bearing of a device. [Chapter 5, *Using Android Cameras*, on page ?](#), lets us access the Android cameras through Processing. We'll display a camera preview of the front- and back-facing cameras, snap and save pictures to the camera's SD card, and superimpose images.

In Part III, we'll learn about peer-to-peer networking. [Chapter 6, *Networking Devices with Wi-Fi*, on page ?](#), teaches us how to connect the Android with our desktop via Wi-Fi using the Open Sound Control protocol. We'll create a virtual whiteboard app, where you and your friends can doodle collaboratively, and we'll build a marble-balancing game, where two players compete on a shared virtual board. [Chapter 7, *Peer-to-Peer Networking Using Bluetooth and Wi-Fi Direct*, on page ?](#), shows us how to use Android Bluetooth technology to discover, pair, and connect Android devices. We'll create a remote cursor sketch and build a survey app to share questions and answers between devices. [Chapter 8, *Using Near Field Communication \(NFC\)*, on page ?](#), introduces us to the emerging short-range radio standard designed for zero-click interaction at close proximity and is expected to revolutionize the point-of-sale industry. We'll read and write NFC tags and exchange data between Android devices via NFC and Bluetooth.

Part IV deals with data and storage, as all advanced apps require some sort of data storage and retrieval to keep user data up-to-date. In [Chapter 9, *Working with Data*, on page ?](#), we'll load, parse, and display data from text files and write data to a text file in the Android storage. We'll also connect to a data source hosted online to create an earthquake app that visualizes currently reported earthquakes worldwide. [Chapter 10, *Using SQLite Databases*, on page ?](#), introduces us to the popular SQLite database management system and Structured Query Language. We'll record sensor data into a SQLite database and query it for particular data attributes.

Part V gets us going with 3D graphics and cross-platform apps. [Chapter 11, *Introducing 3D Graphics with OpenGL*, on page ?](#), will show us how to work with 3D primitives, how virtual light sources are used, and how cameras are animated. [Chapter 12, *Working with Shapes and 3D Objects*, on page ?](#), deals with 2D

vector shapes and how to load and create 3D objects. [Chapter 13, *Sharing and Publishing Applications*, on page ?](#), opens up our mobile app development to a wide range of devices and platforms using the JavaScript mode in Processing. We'll discuss some of the benefits of web apps being able to run on all modern browsers and the range of limitations using built-in device hardware.

How to Read This Book

The five parts of the book can each be considered self-contained mini-courses that you can jump right into once you have completed Part I, have properly installed all the required software, and are up and running. While the book does progress in a step-by-step fashion from fundamental to advanced subjects, you can jump right into Part II, III, IV, or V if you need to tackle a particular family of sensors or Android features for a current project.

Whenever we undertake a project that builds on prior code, refines an earlier project, or revisits important concepts mentioned earlier in the book, we'll cross-reference those earlier sections accordingly; if you are using the ebook, you can use the link to jump directly to the referenced section.

Throughout our journey in this book, I encourage you to get inspired by the projects in the Processing exhibition (<http://processing.org/exhibition/>), learn from the Processing community (<http://forum.processing.org/>), use the code examples included in IDE File → “Examples...,” refer to the online tutorials (<http://processing.org/learning/>), collaborate with peers (<http://sketchpad.cc>), and learn from public sources, such as <http://www.openprocessing.org/>.

What You Need to Use This Book

For all the projects in this book, you need the following software tools. The first chapter guides you through installing those tools step by step.

- Processing 3¹⁰
- Android 4.0 Ice Cream Sandwich or higher¹¹ (2.3 Gingerbread is sufficient for all projects but [Chapter 7, *Peer-to-Peer Networking Using Bluetooth and Wi-Fi Direct*, on page ?](#), and [Chapter 8, *Using Near Field Communication \(NFC\)*, on page ?](#)).
- Ketai Sensor Library for Processing¹²
- Processing Android installation instructions¹³

10. <http://processing.org/download/>

11. <http://developer.android.com/sdk/>

12. <http://ketai.org>

13. <http://wiki.processing.org/w/Android#Instructions>

The projects in this book require at least one Android device. To complete Part III, you need two Android devices. This allows us to run and test the sketches on the actual hardware, use the actual sensors, and get the actual mobile user experience that is the focus of this book.

Tested Android Devices for This Book

The example code for the projects in this book has been tested on the following devices:



- Asus Transformer Prime Tablet with 32 GB memory (Ice Cream Sandwich, Jelly Bean)
- Samsung Galaxy SIII (Ice Cream Sandwich, Jelly Bean, KitKat)
- Samsung Nexus S (Ice Cream Sandwich, Jelly Bean, KitKat)
- Google Nexus 7 with 8 GB memory (Jelly Bean, KitKat)
- Google Nexus 6 with 25 GB memory (Lollipop), not shown.

All the code is available online. Feel free to comment and drop some feedback!

Online Resources

You can download the complete set of source files from the book's web page at http://pragprog.com/titles/dsproc/source_code. The compressed file available online contains all the media assets you need organized by chapter directories and

individual projects. If you're reading the ebook, you can also open the discussed source code just by clicking the file path before the code listings.

The online forum for the book, located at <http://forums.pragprog.com/forums/209>, provides a place for feedback, discussion, questions, and—I hope—answers as well. In the ebook, you'll find a link to the forum on every page next to a “report erratum” link that points to <http://pragprog.com/titles/dsproc/errata>, where you can report errors such as typos, technical errors, and suggestions. Your feedback and suggestions are very much appreciated.

Let's get started! Once we're done installing our software tools in [Chapter 1, *Getting Started*, on page ?](#), we are only minutes away from completing our first Android app.

Daniel Sauter

Associate Professor, School of Art, Media, and Technology at Parsons, The New School

daniel@ketai.org

New York City, 2015-03-31