Preface

We created Processing to make programming interactive graphics easier. We were frustrated with how difficult it was to write this type of software with the programming languages we usually used (Java and C++), and were inspired by how simple it was to write interesting programs with the languages of our childhood (Logo and BASIC). We were most influenced by Design By Numbers (DBN), a language we were maintaining and teaching at the time (and which was created by our research advisor, John Maeda).

Processing was born in spring 2001 as a brainstorming session on a sheet of paper. Our goal was to make a way to sketch (prototype) the type of software we were working on, which was almost always full-screen and interactive. We were searching for a better way to test our ideas easily in code, rather than just talking about them or spending too much time programming them in C++. Our other goal was to make a language for teaching design and art students how to program and to give more technical students an easier way to work with graphics. The combination is a positive departure from the way programming is usually taught. We begin by focusing on graphics and interaction rather than on data structures and text console output.

Processing experienced a long childhood; it was alpha software from August 2002 to April 2005 and then public beta software until November 2008. During this time, it was used continuously in classrooms and by thousands of people around the world. The language, software environment, and curricula around the project were revised continuously during this time. Many of our original decisions about the language were reinforced and many were changed. We developed a system of software extensions, called *libraries*, that have allowed people to expand Processing into many unforeseen and amazing directions. (There are now over 100 libraries.)

In fall 2008, we launched the 1.0 version of the software. After seven years of work, the 1.0 launch signified stability for the language. We launched the 2.0 release in spring 2013 to make the software faster. The 2.0 releases introduced better OpenGL integration, GLSL shaders, and faster video playback with GStreamer. The 3.0 releases in 2015 make programming in Processing easier with a new interface and error checking while programming.

Now, fourteen years after its origin, Processing has grown beyond its original goals, and we've learned how it can be useful in other contexts. Accordingly, this book is written for a new audience—casual programmers, hobbyists, and anyone who wants to explore what Processing can do without getting lost in the details of a huge textbook. We hope you'll have fun and be inspired to continue programming. This book is just the start.

While we (Casey and Ben) have been guiding the Processing ship through the waters for the last twelve years, we can't overstate that Processing is a community effort. From writing libraries that extend the software to posting code online and helping others learn, the community of people who use Processing has pushed it far beyond its initial conception. Without this group effort, Processing would not be what it is today.

How This Book Is Organized

The chapters in this book are organized as follows:

- Chapter 1: Learn about Processing.
- Chapter 2: Create your first Processing program.
- Chapter 3: Define and draw simple shapes.
- Chapter 4: Store, modify, and reuse data.
- Chapter 5: Control and influence programs with the mouse and the keyboard.
- Chapter 6: Transform the coordinates.
- Chapter 7: Load and display media including images, fonts, and vector files.
- Chapter 8: Move and choreograph shapes.
- Chapter 9: Build new code modules.
- Chapter 10: Create code modules that combine variables and functions.
- Chapter 11: Simplify working with lists of variables.
- Chapter 12: Load and visualize data.
- Chapter 13: Learn about 3D, PDF export, computer vision, and reading data from an Arduino board.

Who This Book Is For

This book is written for people who want a casual and concise introduction to computer programming, who want to create images and simple interactive programs. It's for people who want a jump-start on understanding the thousands of free Processing code examples and reference materials available online. *Getting Started with Processing* is not a programming textbook; as the title suggests, it will get you started. It's for teenagers, hobbyists, grandparents, and everyone in between.

This book is also appropriate for people with programming experience who want to learn the basics of interactive computer graphics. *Getting Started with Processing* contains techniques

that can be applied to creating games, animation, and interfaces.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width italic`
: Shows text that should be replaced with user-supplied values or by values determined by context.

This element signifies a tip, suggestion, or general note.

This element indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from Make: books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a sig-

nificant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: *"Getting Started with Processing* by Casey Reas and Ben Fry. Copyright 2015 Casey Reas and Ben Fry, 978-1-457-18708-7."

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at *bookpermissions@makermedia.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of plans and pricing for enterprise, government, education, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like Maker Media, O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and hundreds more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

Maker Media, Inc.
1160 Battery Street East, Suite 125
San Francisco, California 94111
800-998-9938 (in the United States or Canada)
http://makermedia.com/contact-us/

Make: unites, inspires, informs, and entertains a growing community of resourceful people who undertake amazing projects in their backyards, basements, and garages. Make: celebrates your right to tweak, hack, and bend any technology to your will. The Make: audience continues to be a growing culture and community that believes in bettering ourselves, our environment, our educational system—our entire world. This is much more than an audience, it's a worldwide movement that Make: is leading—we call it the Maker Movement.

For more information about Make:, visit us online:

Make: magazine: *http://makezine.com/magazine/*
Maker Faire: *http://makerfaire.com*
Makezine.com: *http://makezine.com*
Maker Shed: *http://makershed.com/*

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at: *http://shop.oreilly.com/product/0636920031406.do*

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

Acknowledgments

For the first and second editions of this book, we thank Brian Jepson for his great energy, support, and insight. For the first edition, Nancy Kotary, Rachel Monaghan, and Sumita Mukherji gracefully carried the book to the finish line. Tom Sgouros made a thorough edit of the book and David Humphrey provided an insightful technical review.

We can't imagine this book without Massimo Banzi's *Getting Started with Arduino* (Maker Media). Massimo's excellent book is the prototype.

A small group of individuals has, for years, contributed essential time and energy to Processing. Dan Shiffman is our partner in the Processing Foundation, the 501(c)(3) organization that supports the Processing software. Much of the core code for Processing 2.0 and 3.0 has come from the sharp minds of Andres Colubri and Manindra Moharana. Scott Murray, Jamie Kosoy, and Jon Gacnik have built a wonderful web infrastructure for the project. James Grady is rocking the 3.0 user interface. We thank Florian Jenett for his years of diverse work on the project including the forums, website, and design. Elie Zananiri and Andreas Schlegel have created the infrastructure for building and documenting contributed libraries and have spent countless hours curating the lists. Many others have contributed significantly to the project; the precise data is available at *https://github.com/processing*.

The Processing 1.0 release was supported by Miami University and Oblong Industries. The Armstrong Institute for Interactive Media Studies at Miami University funded the Oxford Project, a series of Processing development workshops. These workshops were made possible through the hard work of Ira Greenberg. These four-day meetings in Oxford, Ohio, and Pittsburgh, Pennsylvania, enabled the November 2008 launch of Processing 1.0. Oblong Industries funded Ben Fry to develop Processing during summer 2008; this was essential to the release.

The Processing 2.0 release was facilitated by a development workshop sponsored by New York University's Interactive Telecommunication Program. The work on Processing 3.0 was generously sponsored by the Emergent Digital Practices program at the University of Denver. We thank Christopher Colemen and Laleh Mehran for the essential support.

This book grew out of teaching with Processing at UCLA. Chandler McWilliams has been instrumental in defining these classes. Casey thanks the undergraduate students in the Department of Design Media Arts at UCLA for their energy and enthusiasm. His teaching assistants have been great collaborators in defining how Processing is taught. Hats off to Tatsuya Saito, John Houck, Tyler Adams, Aaron Siegel, Casey Alt, Andres Colubri, Michael Kontopoulos, David Elliot, Christo Allegra, Pete Hawkes, and Lauren McCarthy.

Through founding the Aesthetics and Computation Group (1996–2002) at the MIT Media Lab, John Maeda made all of this possible.

1/Hello

Processing is for writing software to make images, animations, and interactions. The idea is to write a single line of code, and have a circle show up on the screen. Add a few more lines of code, and the circle follows the mouse. Another line of code, and the circle changes color when the mouse is pressed. We call this *sketching* with code. You write one line, then add another, then another, and so on. The result is a program created one piece at a time.

Programming courses typically focus on structure and theory first. Anything visual—an interface, an animation—is considered a dessert to be enjoyed only after finishing your vegetables, usually several weeks of studying algorithms and methods. Over the years, we've watched many friends try to take such courses and drop out after the first lecture or after a long, frustrating night before the first assignment deadline. What initial curiosity they had about making the computer work for them was lost because they couldn't see a path from what they had to learn first to what they wanted to create.

Processing offers a way to learn programming through creating interactive graphics. There are many possible ways to teach coding, but students often find encouragement and motivation in immediate visual feedback. Processing's capacity for providing that feedback has made it a popular way to approach pro-

gramming, and its emphasis on images, sketching, and community is discussed in the next few pages.

Sketching and Prototyping

Sketching is a way of thinking; it's playful and quick. The basic goal is to explore many ideas in a short amount of time. In our own work, we usually start by sketching on paper and then moving the results into code. Ideas for animation and interactions are usually sketched as storyboards with notations. After making some software sketches, the best ideas are selected and combined into prototypes (Figure 1-1). It's a cyclical process of making, testing, and improving that moves back and forth between paper and screen.

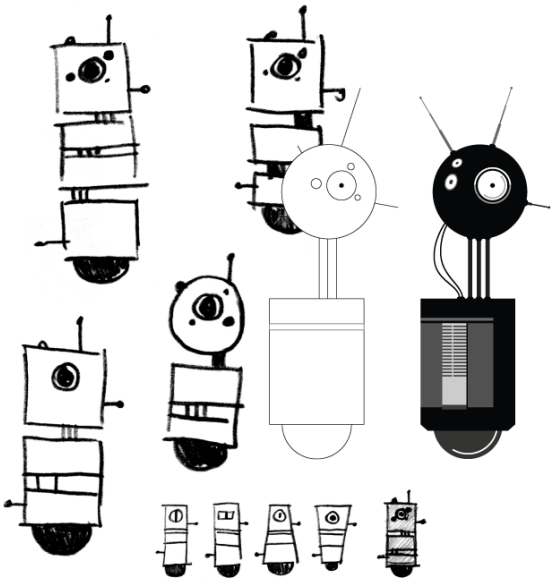

Figure 1-1. *As drawings move from sketchbook to screen, new possibilities emerge*

Flexibility

Like a software utility belt, Processing consists of many tools that work together in different combinations. As a result, it can

be used for quick hacks or for in-depth research. Because a Processing program can be as short as one line or as long as thousands, there's room for growth and variation. More than 100 libraries extend Processing even further into domains including sound, computer vision, and digital fabrication (Figure 1-2).

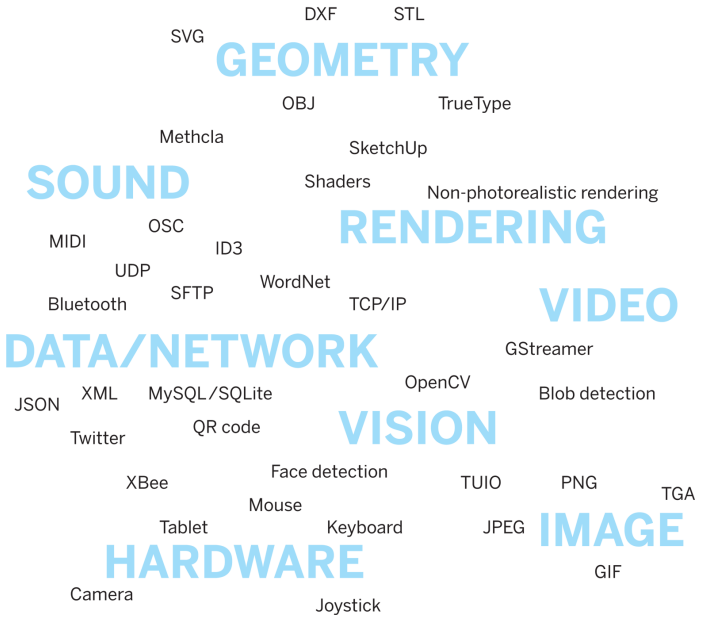

Figure 1-2. *Many types of information can flow in and out of Processing*

Giants

People have been making pictures with computers since the 1960s, and there's much to be learned from this history. For example, before computers could display to CRT or LCD screens, huge plotter machines (Figure 1-3) were used to draw images. In life, we all stand on the shoulders of giants, and the titans for Processing include thinkers from design, computer graphics, art, architecture, statistics, and the spaces between. Have a look at Ivan Sutherland's *Sketchpad* (1963), Alan Kay's *Dynabook* (1968), and the many artists featured in Ruth Leavitt's *Artist and Computer* (Harmony Books, 1976). The ACM SIG-

GRAPH and Ars Electronica archives provide fascinating glimpses into the history of graphics and software.

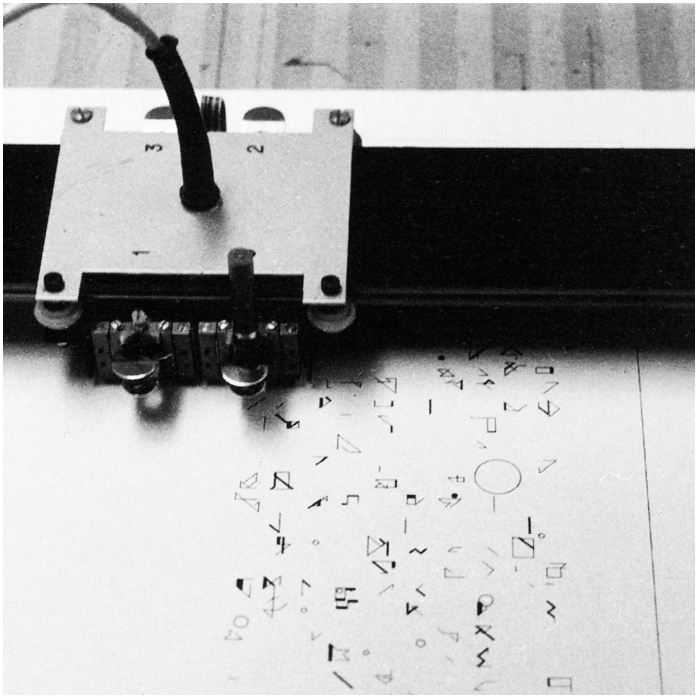

Figure 1-3. *Drawing demonstration by Manfred Mohr at Musée d'Art Moderne de la Ville de Paris using the Benson plotter and a digital computer on May 11, 1971. (photo by Rainer Mürle, courtesy bitforms gallery, New York)*

Family Tree

Like human languages, programming languages belong to families of related languages. Processing is a dialect of a programming language called Java; the language syntax is almost identical, but Processing adds custom features related to graphics and interaction (Figure 1-4). The graphic elements of Processing are related to PostScript (a foundation of PDF) and OpenGL (a 3D graphics specification). Because of these shared features, learning Processing is an entry-level step to programming in other languages and using different software tools.

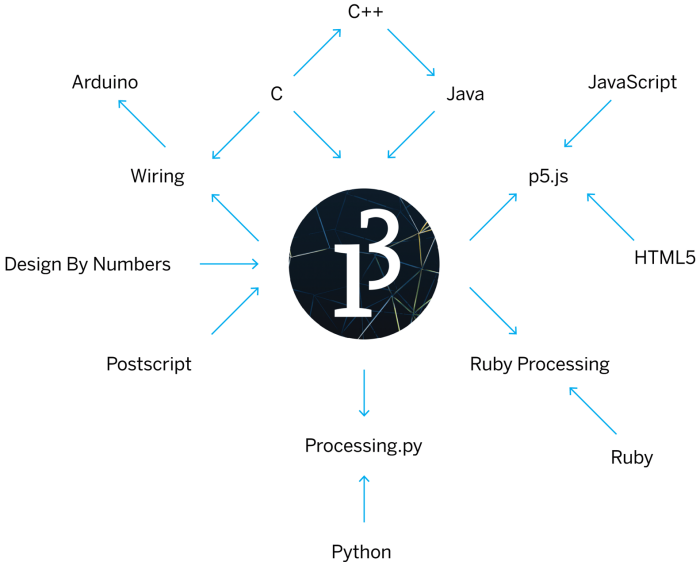

Figure 1-4. *Processing has a large family of related languages and programming environments*

Join In

Thousands of people use Processing every day. Like them, you can download Processing without cost. You even have the option to modify the Processing code to suit your needs. Processing is a *FLOSS* project (that is, *free/libre/open source software*), and in the spirit of community, we encourage you to participate by sharing your projects and knowledge online at Processing.org and at the many social networking sites that host Processing content. These sites are linked from the Processing.org website.