Processing is a didactic concept that is based on the idea that programming can be learned very easily through the creation of screen art. This 840-page book comes with an open-source programming environment called processing development environment (PDE), which has to be used together with the book (available for Linux, Mac OS X, and Windows). Rather than just reviewing the book itself, my review is based on my overall experience with this concept.

According to the introduction, the author is a graphical artist and painter who learned the art of programming by self-instruction. He began by implementing the pattern structure of one of his field paintings; namely, he wrote a program that approximated the content of the painting on canvas on the computer screen. According to him: "The really exciting thing ... happened when I was able to generate hundreds of painting variations by simply changing some of the values in the program." This quote describes the main idea of the book: motivate and teach computer programming through the creation of images and animations. As already mentioned, the concept is backed by the open-source programming environment PDE.

PDE was originally created at the Massachusetts Institute of Technology (MIT) by Ben Fry and Casey Reas, and provides a simplified view of Java. It basically abstracts from object orientation, and drastically simplifies many technical tasks, such as drawing in a window or mouse interaction. The version that I tested (0130 BETA, from October 20th, 2007) also featured functionality that goes far beyond the book, such as exporting a processing program as a Java Applet or executable application (Windows EXE, Mac OS X, or Linux binary executable), as well as the inclusion of libraries for PDF export or video and audio manipulation.

In the 13 chapters and two appendices of the book, Greenberg presents many fundamental programming concepts, always by guiding the reader through the creation of two-dimensional (2D) and three-dimensional (3D) graphics. The first five chapters are devoted to motivating and explaining the concept of processing, and include the very fundamental ideas of programming. Chapters 6 through 13 are meant to be followed by hands-on programming using PDE. Appendix A provides a reference to the processing application programming interface (API), and Appendix B, Math Reference, revises high school-level math concepts. Each hands-on chapter guides the user through a certain graphical concept (such as lines, curves, or shapes), and discusses picture projects and presents source code.

One of the disadvantages of the processing concept is that it relies heavily on math. Processing gets rid of many technical issues, but many examples still require an awareness of undergraduate math. I suspect this could restrict the readership of the book. Also, since processing programs are very small, I would have liked to see more comments, especially in the source code that is presented in the book. Sometimes, the book presents source code that fills two or three pages, with only five or six lines of comments.

Although the concept of teaching programming using a simplified programming language that produces graphics or animations has been practiced in academia for decades, this book has a slightly different goal. The book introduces programming, not for the sake of programming, but for the sake of art. The main goal is to educate designers and artists, not undergraduate computer scientists. This is, in my opinion, not only an innovative idea; the book in combination with the programming environment
also presents an excellent implementation of the concept. I recommend this work for anybody who wants to explore programming in a different way, or for a teacher who wants to get inspired for his or her own programming class. I also recommend downloading and playing around with the processing development environment (www.processing.org).

Reviewer: Gerald Friedland

Review #: CR134991 (0810-0959)