Hacking the Kinect with OpenNI, NITE, and Java

Andrew Davison

Kinect Open Source Programming Secrets

Hacking the Kinect with OpenNI, NITE, and Java

Andrew Davison

New York Chicago San Francisco Lisbon
London Madrid Mexico City Milan New Delhi
San Juan Seoul Singapore Sydney Toronto

The McGraw-Hill Companies

Cataloging-in-Publication Data is on file with the Library of Congress

Kinect Open Source Programming Secrets: Hacking the Kinect with OpenNI, NITE, and Java

1 2 3 4 5 6 7 8 9 0 DOC DOC 1 0 9 8 7 6 5 4 3 2

ISBN 978-0-07-178317-0
MHID 0-07-178317-2

Sponsoring Editor
Roger Stewart

Editorial Supervisor
Patty Mon

Project Manager
Rohini Deb,
Cenveo Publisher Services

Acquisitions Coordinator
Molly Wyand

Copy Editor
Marilyn Smith

Proofreader
Claire Splan

Indexer
Karin Arrigoni

Production Supervisor
George Anderson

Composition
Cenveo Publisher Services

Illustration
Cenveo Publisher Services

Art Director, Cover
Jeff Weeks

Cover Designer
Jeff Weeks

Kinect Open Source Programming Secrets

About the Author

Andrew Davison (Hat Yai, Thailand) received his Ph.D. from Imperial College in London, and was a lecturer at the University of Melbourne for six years, before moving to Prince of Songkla University in Thailand. He has also taught in Bangkok, Khon Kaen, and Hanoi. His research interests include scripting languages, logic programming, visualization, and teaching methodologies. These led to an interest in teaching game programming and natural user interfaces. He is the author of two game programming books: *Killer Game Programming in Java* (O'Reilly, 2005) and *Pro Java 6 3D Game Development* (Apress, 2007). He is also the coauthor with Carol Hamer of *Learn BlackBerry Games Development* (Apress, 2010).

Acknowledgments

As always, I thank my wife, Supatra, and son, John, who have again been a source of strength.

I must acknowledge my department and faculty members at Prince of Songkla University in Thailand, who have given me the time and resources for completing this work. I heartily recommend Thailand as a wonderful place to live.

Finally, regards to Roger Stewart, Patty Mon, Rohini Deb, Marilyn Smith, Claire Splan, and the rest of the team at McGraw-Hill and Cenveo Publisher Services, who helped me focus on the important things.

Introduction

It was at my university's 2011 Open Day that I realized the importance of the Kinect sensor. Returning from lunch, I saw a very long line of people waiting to get into one of our computer labs. A lab of PCs is nothing special to today's computer-savvy kids, so I walked over to see what the fuss was about. People wanted to dance in a virtual disco, boogying on down in front of a Kinect.

To my mind, the Kinect has achieved two remarkable things. One is that it introduced the wonders of computer vision to a public waiting for the next Big Thing. Also, the Kinect is the first consumer-level (affordable) device that makes it comparatively easy to program with motion sensing, skeletal tracking, and natural user interfaces using gestures. That's why the Kinect has caught the imaginations of the hacker/developer community, won a slew of awards for best gadget and innovative engineering, and been awarded a Guinness World Record for the fastest selling consumer electronics device.

But notice that I said "comparatively easy to program." It's fair to say that there has been a rather dismal lack of documentation on the Kinect's APIs. This book has grown out of my online tutorials (at http://fivedots.coe.psu.ac.th/~ad/kinect), which aim to remedy that problem.

Why Buy This Book?

I can suggest four reasons for buying this book:

- It offers a unique choice of Kinect programming tools.
- It explains the official Java wrappers for those tools.
- It covers topics not found elsewhere.
- It provides depth, but with brevity.

Unique Programming Tools

This is the only book on programming the Kinect using the OpenNI library, NITE, and Java (as of April 2012, when this book went to press). Of course, this claim won't mean much if you don't know OpenNI or NITE, so let me briefly describe them.

OpenNI (or more correctly, the OpenNI framework) is open source software for accessing the Kinect's sensors. OpenNI has been ported to multiple platforms, such as Windows, Mac OS X, and Linux. It was developed by PrimeSense, the company that worked with Microsoft on building the Kinect. NITE adds higher-level features to OpenNI, including user tracking and hand gestures. It was also developed by PrimeSense, and although it isn't open source, it is available for free. OpenNI and NITE have been around since the early days of the Kinect, and a vast number of fun projects use their APIs. The OpenNI and NITE developer forums are very active, providing a lot of help.

Official Java Wrappers

This is the only book that explains the official Java wrappers for OpenNI and NITE (again, as of April 2012). By "official," I mean that these bindings were developed by PrimeSense. Obvious advantages of Java include object-orientation, cross-platform support, availability for free, and many people (including you, probably) knowing how to program with it. Most important, programming in Java gives you access to a massive number of libraries—for graphics, networking, and beyond—that can be linked to the Kinect without much effort. For example, I'll demonstrate how to use the Java 3D graphics library and the Java binding for the OpenCV computer vision package.

Some Kinect books employ the Processing language, a simplified version of Java aimed at new programmers interested in generating graphics. Processing has many interesting libraries, but nothing like Java's vast collections.

There are other Java bindings for OpenNI, created by third-party developers. These APIs tend to have limited functionality and are not consistently maintained.

The main drawback of using the PrimeSense Java wrappers is their lack of documentation. As I explain in Chapter 1, I had to decompile the libraries' JAR files, and work out the correspondences between the Java source and the somewhat better documented C++ OpenNI/NITE APIs. (This is why including *Secrets* in the book's title isn't too excessive.)

A Wide Range of Topics

This book covers programming topics not found elsewhere. I start off with the basics, of course, with chapters on depth, infrared, and RGB imaging, point clouds, skeletal user tracking, hand tracking, and gesture support. Moving beyond that, I cover several novel and unusual features, including the following:

- Kinect gaming based around a version of the classic Breakout video game.
- Controls for the Kinect motor, LED, and accelerometer, which are not part of the standard OpenNI API. In fact, their absence is often held up as a serious drawback of the API. It's actually quite easy to add these capabilities using a custom-built USB driver.

- 3D graphics programming in the point cloud and skeletal tracking examples, using Java 3D.
- A computer vision example that demonstrates how to link the Kinect to the popular (and powerful) OpenCV library.
- The creation of new body gestures (inspired by the FAAST system), which are not part of the limited NITE repertoire.
- A new type of GUI component controlled by hand gesturing, illustrated with three examples: a button, dial, and slider. These components are controlled without the help of mouse or keyboard.

Depth with Brevity

This book describes a lot of complicated code but, unlike some rather hefty programming tomes, you won't find all the code tediously printed on these pages. Instead, you can download it from the book's website at http://fivedots.coe.psu.ac.th/~ad/kinect/. In addition, I've been adding supplementary chapters to the website, including ones discussing speech recognition and the Kinect microphone array.

I've also cut down on the page count by assuming that you have some Java programming experience (perhaps you've taken a one-semester course on Java at college). That means that I don't explain how to create a JFrame or search for substrings. I don't describe every line of every program, but concentrate on the important Kinect-specific material.

What This Book Is Not About

The fact that I'm powering ahead without wasting time explaining about classes, objects, and inheritance could be viewed as a disadvantage. This is not a book for first-time Java programmers.

Also, I don't have the space to seriously explain the topics of 3D graphics or computer vision. I introduce them (in the form of Java 3D and JavaCV) in enough detail so that you can understand my examples, but there is a lot of material that I don't mention. When I get to these topics, I'll suggest resources for finding more information.

Contents at a Glance

Contents

Chapter 1

Getting Started

This chapter introduces the Kinect sensor, explains how to install the developer software on Windows (which can be a bit tricky), looks at some examples that come with the download, and provides an overview of the Java API for programming the Kinect. It even includes a very simple Java program for testing the API.

Later chapters will start programming in earnest, looking at imaging (depth detection, camera, and infrared), point clouds, skeletal user tracking in 2D and 3D, tilt motor control, hand tracking, Kinect games, gesture support, the Kinect and OpenCV, and gesture-controlled graphical user interface (GUI) components.

I'll be programming the Kinect with OpenNI and NITE, not OpenKinect, CL NUI, or Microsoft's Kinect for Windows SDK. The 3D features (in the point cloud and skeletal tracking examples) are implemented using Java 3D. My OpenCV coding will be with its JavaCV binding. I'll be using the Java API released with OpenNI and NITE. I won't be using Processing libraries such as Simple-OpenNI.

You can find detailed code examples at my website: http://fivedots.coe.psu.ac.th/~ad/kinect/.

The Kinect Sensor Hardware

The Kinect sensor was originally intended to be a motion-sensing input device for the Xbox 360, allowing the user to control games via gestures and spoken commands. Key hardware components are an RGB camera, a depth sensor, multiarray microphones, a tilt motor, and a three-axis accelerometer. Figure 1-1 illustrates the components of the Kinect sensor.

Soon after the Kinect's launch in November 2010, third-party developers started releasing software to allow it to be used on platforms other than just the Xbox (such as Windows, Linux, and the Mac). Microsoft eventually came out with a Windows 7–based SDK in June 2011. In Februray 2012, Microsoft released a new version of the Kinect specifically aimed at desktop PCs running Windows 7 or 8, aptly called "Kinect for Windows," featuring a new "Near Mode" that lets gesture control be used as close as 40 cm. However, all the examples in this book use the older (and cheaper) original Kinect.