Microsoft

Start Here!™

Learn the Kinect™ API

Rob Miles

Ready to learn Kinect programming?

Learn the fundamentals of programming with the Kinect™ API—and begin building apps that use motion tracking, voice recognition, and more. If you have experience programming with C#—simply start here! This book introduces must-know concepts and techniques through easy-to-follow explanations, examples, and exercises.

Here's where you start learning Kinect

- Build an application to display Kinect video on your PC
- Have Kinect take photographs when it detects movement
- Draw on a computer screen by moving your finger in the air
- Track your body gestures and use them to control a program
- Make a program that understands your speech and talks back to you
- Play a part in your own augmented reality game
- Create an "air piano" using Kinect with a MIDI device

GET CODE SAMPLES ONLINE
Ready to download at
http://go.microsoft.com/FWLink/?Linkid=252996

For system requirements, see the ***Introduction.***

ISBN: 978-0-7356-6396-1

U.S.A. $34.99
Canada $36.99
[*Recommended*]

Programming/Microsoft Kinect

Start Here! Learn the Kinect API

For Skill Level: Some experience required
Prerequisites: Beginner-level C# skills

DEVELOPER ROADMAP

Start Here!
- Beginner-level instruction
- Easy to follow explanations and examples
- Exercises to build your first projects

Step by Step
- For experienced developers learning a new topic
- Focus on fundamental techniques and tools
- Hands-on tutorial with practice files plus eBook

Developer Reference
- Professional developers; intermediate to advanced
- Expertly covers essential topics and techniques
- Features extensive, adaptable code examples

Focused Topics
- For programmers who develop complex or advanced solutions
- Specialized topics; narrow focus; deep coverage
- Features extensive, adaptable code examples

microsoft.com/mspress

About the Author

Rob Miles, Microsoft® MVP for Device Application Development, has taught computer programming for more than 25 years. He also consults on a wide range of commercial software projects.

Microsoft®

Microsoft

Start Here!™

Learn Microsoft®

Kinect API

Rob Miles

ISBN: 978-0-735-66396-1

1 2 3 4 5 6 7 8 9 LSI 7 6 5 4 3 2

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at *mspinput@microsoft.com*. Please tell us what you think of this book at *http://www.microsoft.com/learning/booksurvey*.

Acquisitions and Developmental Editor: Russell Jones

Production Editor: Kristen Borg

Editorial Production: Tiffany Rupp, S4Carlisle Publishing Services

Technical Reviewer: Peter Robinson

Copyeditor: Heath Lynn Silberfeld

Indexer: WordCo Indexing Services, Inc.

Cover Design: Jake Rae

Cover Composition: Karen Montgomery

Illustrator: S4Carlisle Publishing Services

To Gus

Contents at a Glance

Contents

Introduction

The Kinect sensor provides a genuinely new way for a computer to make some sense of the world around it. The fusion of a camera, a directional microphone system, and a depth sensor into a single, mass-market device provides an opportunity for software developers to advance the field of computer interaction in all kinds of exciting ways.

It is now possible to create programs that use the Kinect sensor to create a computer interface with the ability to recognize users and understand their intentions using a "natural" user interface consisting of gestures and spoken commands. In addition, the device's capabilities have a huge range of possible applications, from burglar alarms to robot controllers.

Start Here! Learn the Kinect™ API gives you an overview of how the Kinect sensor works and how the Kinect for Windows SDK exposes each of the data sources. The book introduces each of the sensors in the context of solving a well-defined problem. The full source code is provided for each example program. You will also find plenty of ideas for further development of both the sample programs and your own applications.

In addition to an overview of the Kinect for Windows SDK, this book explores the fundamentals of the signals being processed: how video, audio, depth, and 3D skeleton information can be represented in a program. Also included is coverage of specific programming issues that are highly relevant to the creation of programs that deal with large streams of data from sensors, including memory allocation, creating unmanaged code to improve performance, and threading. If you want to learn more about these aspects of program development, you will find good coverage and sample code that works. Although this book doesn't cover every Kinect for Windows SDK, it provides a solid starting point for experimentation and further development.

Who Should Read This Book

This book is intended to be read by C# developers who have a Kinect sensor, either from an Xbox 360 or a Kinect for Windows device, and want to find out how to use the Kinect for Windows SDK to create programs that can process video, sound, and depth views and perform skeleton tracking. If you have an idea for a product based on the Kinect sensor, you can use this book to get a solid grounding in the technology—and you might even be able to use some of the sample code as the basis of your first steps along the road to a working solution.

Assumptions

This book expects that you have a reasonable understanding of .NET development using the C# programming language. You should be familiar with the Visual Studio 2010 development environment and object-oriented programming development.

All the examples are provided in the C# language. It will be helpful (although not required) if you have some experience with Windows Presentation Foundation (WPF) development. In addition, some examples make use of the XNA game development framework. The key development principles important to the development of Kinect software are explained in some detail, so you can use the text to broaden your programming knowledge.

Who Should *Not* Read This Book

If you have never programmed before, you will not find sufficient background on the C# language to be able to understand the examples. If you want to learn how to use the language, you might consider reading John Mueller's *Start Here!™ Learn Microsoft® Visual C#® 2010* (Microsoft Press, 2011) and/or John Sharp's *Microsoft® Visual C#® 2010 Step by Step* (Microsoft Press, 2011).

The text of this book provides coverage of the managed code Application Programmer Interface (API) supported by the Kinect for Windows SDK. So if you are a C++ developer who wishes to learn how to interact with the Kinect sensor from unmanaged C++ programs, you will find that the code samples supplied will not provide this information.

Organization of This Book

This book is divided into four sections, each of which builds on the previous section to give you an overview of the Kinect sensor, the Kinect for Windows SDK, and how to create programs that make use of the data. Part I, "Getting Started," provides an overview of how the sensor works and how you can get a Kinect sensor connected to and working with your computer. Part II, "Using the Kinect Sensor in Programs," covers the fundamentals of sensor initialization and then introduces each of the data sources, video, depth, and sound. Part III, "Creating Advanced User Interfaces," shows how the Kinect SDK performs body tracking and how a program can use this information. It also shows how data from the sensors can be combined to produce augmented-reality applications. Finally, Part IV, "Kinect in the Real World," shows how you can use the

Kinect to interact with external devices. This section provides additional programming insight and identifies future directions for exploring this fascinating new sensor.

Conventions and Features in This Book

This book presents information using conventions designed to make the information readable and easy to follow:

- Boxed elements with labels such as "Note" provide additional information or alternative methods for completing a step successfully.
- A plus sign (+) between two key names means that you must press those keys at the same time. For example, "Press Alt+Tab" means that you hold down the Alt key while you press the Tab key.
- A vertical bar between two or more menu items (e.g. File | Close), means that you should select the first menu or menu item, then the next, and so on.

System Requirements

You will need the following hardware and software to complete the practice exercises in this book:

- Windows 7, 32- or 64-bit version
- Visual Studio 2010, any edition (multiple downloads may be required if using Express Edition products)
- The Kinect for Windows SDK
- Computer that has a 1 GHz or faster processor (2 GHz recommended)
- 1 GB (32 bit) or 2 GB (64 bit) RAM
- 3.5 GB of available hard disk space
- 5,400 RPM hard disk drive
- DirectX 9 capable video card running at 1024 x 768 or higher-resolution display
- DVD-ROM drive (if installing Visual Studio 2010 from DVD)
- Internet connection to download software or chapter examples

Depending on your Windows configuration, you might require local administrator rights to install or configure Visual Studio 2010 and SQL Server 2008 products.

Code Samples

Most of the chapters in this book include exercises that let you interactively try out new material learned in the main text. All the sample projects can be downloaded from the following page:

http://www.microsoftpressstore.com/title/9780735663961

Follow the instructions to download the *KinectStartHereCompanionContent.zip* file.

Installing the Code Samples

Follow these steps to install the code samples on your computer so that you can use them with the exercises in this book:

1. Unzip the *KinectStartHereCompanionContent.zip* file that you downloaded from the book's website to a directory on your hard drive. It's best to create a directory near the root of your drive, such as C:\KinectExamples.
2. If prompted, review the displayed end user license agreement. If you accept the terms, select the accept option, and then click Next.

Note If the license agreement doesn't appear, you can access it from the same webpage from which you downloaded the *KinectStartHere CompanionContent.zip* file.

Using the Code Samples

The folder created by the Setup.exe program contains a subfolder for each chapter. In turn, these subfolders contain a number of subfolders, one for each example. The examples have the folder names provided in this book's text. Each contains the complete Visual Studio project and all the source code and resources required to build them. (To reduce the size of the download file, the examples do not contain the executable programs themselves; you will have to compile the example programs using Visual Studio run them.)

Note Some of the folder paths created by Visual Studio 2010 can be quite "deep"—that is, a folder may contain a subfolder and so on for a number of levels. Installing the sample code in a folder that is already deep in the folder hierarchy on your disk may lead to problems when you try to build the program, because some file systems in use on Windows PC systems have a restriction on the maximum length of a path to a file. If you encounter problems running the example programs, you may be able to solve the problem by moving the examples folder closer to the root of the drive you are using.

Acknowledgments

I'd like to thank the following people: Russell Jones for being such a patient and constructive editor, Peter Robinson for sterling duty on the technical editing front, and Tiffany Timmerman and Kristen Borg for breathing on the text and making it so much nicer to read. Finally, I'd like to thank the Kinect team for making such a fascinating product that is such fun to play with!

Errata and Book Support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site:

http://www.microsoftpressstore.com/title/ 9780735663961

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, please email Microsoft Press Book Support at mspinput@microsoft.com.

Please note that product support for Microsoft software is not offered through the addresses above.

We Want to Hear from You

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://www.microsoft.com/learning/booksurvey

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in Touch

Let's keep the conversation going! We're on Twitter:

http://twitter.com/MicrosoftPress

PART I

Getting Started

In this section you will learn what happens inside the Kinect sensor and how it collects data that lets it see and hear the environment around it. You'll also find out how the signals that it collects are sent over to your computer or Xbox 360. Finally, you will install the Kinect SDK and work with the software to build your first programs that use data from the sensor.

Chapter 1

An Introduction to Kinect

After completing this chapter, you will:

- Understand how the Kinect sensor generates data about the world around it
- Identify the key components of the Kinect sensor and how they work
- Appreciate how the sensors and the Kinect provide useful signals to a connected computer or console

The Kinect Sensor

UNTIL RECENTLY COMPUTERS HAD A very restricted view of the world around them, and users had very limited ways of communicating with computers. Over the years, computers have acquired cameras and audio inputs, but these have been used mostly for *unrecognized* input; computers can store and play such content, but it has been very difficult to make computers *understand* input in these forms.

For example, when people hear a sound, they can make judgments about the distance and direction of the sound source relative to their own position. Until recently, computers had more trouble making such judgments. Audio information from a number of microphones does provide considerable information about the distance and direction of the audio source, but determining this information is difficult for programs to do. Similarly, a video picture provides an image of the environment for the computer to analyze, but a computer has to work very hard to extract information about the objects in pictures or video because an image shows a flat, two-dimensional representation of a three-dimensional world.

Kinect changes all this. The Kinect sensor bar contains two cameras, a special infrared light source, and four microphones. It also contains a stack of *signal processing hardware* that is able to make sense of all the data that the cameras, infrared light, and microphones can generate. By combining the output from these sensors, a program can track and recognize objects in front of it, determine the direction of sound signals, and isolate them from background noise.

Getting Inside a Kinect Sensor

To get an idea of how the Kinect sensor works, you could take one apart and look inside. (Don't do that. There are many reasons why taking your Kinect apart is a bad idea: it's hard to do, you will invalidate your warranty, and you might not be able to restore it to working condition. But perhaps the best reason not to take it apart is that I've already done it for you!)

Figure 1-1 shows a Kinect sensor when it is "fully dressed."

FIGURE 1-1 A Kinect sensor.

Figure 1-2 shows a Kinect with the cover removed. You can see the two cameras in the middle and the special light source on the left. The four microphones are arranged along the bottom of the sensor bar. Together, these devices provide the "view" the Kinect has of the world in front of it.

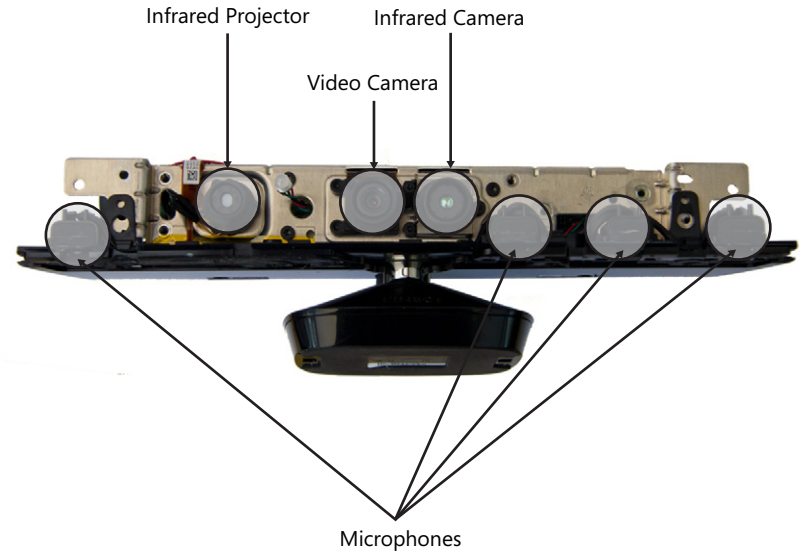


FIGURE 1-2 A Kinect sensor unwrapped.

Figure 1-3 shows all the hardware inside the Kinect that makes sense of the information being supplied from all the various devices.

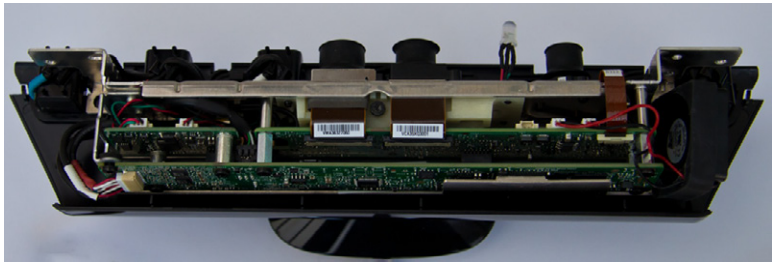


FIGURE 1-3 The Kinect sensor data processing hardware.

To make everything fit into the slim bar form, the designers had to stack the circuit boards on top of each other. Some of these components produce quite a bit of heat, so a tiny fan that can be seen on the far right of Figure 1-3 sucks air along the circuits to keep them cool. The base contains an electric motor and gear assembly that lets the Kinect adjust its angle of view vertically.

Now that you have seen inside the device, you can consider how each component helps the Kinect do what it does, starting with the "3D" camera.

The Depth Sensor

Kinect has the unique ability to "see" in 3D. Unlike most other computer vision systems, the Kinect system is able to build a "depth map" of the area in front of it. This map is produced entirely within the sensor bar and then transmitted down the USB cable to the host in the same way as a typical camera image would be transferred—except that rather than color information for each pixel in an image, the sensor transmits distance values.

You might think that the depth sensor uses some kind of radar or ultrasonic sound transmitter to measure how far things are from the sensor bar, but actually it doesn't. This would be difficult to do over a short distance. Instead, the sensor uses a clever technique consisting of an infrared projector and a camera that can see the tiny dots that the projector produces.

Figure 1-4 shows the arrangement of the infrared projector and sensor.

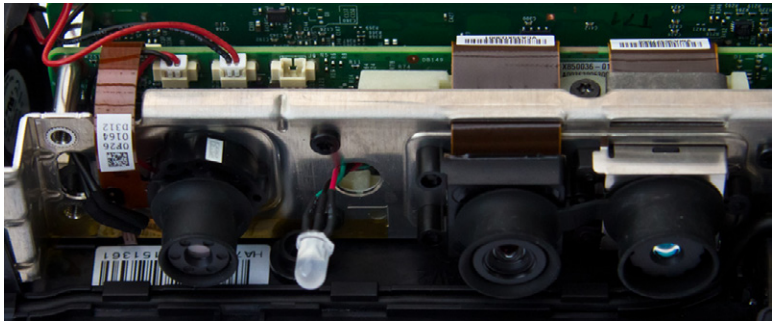


FIGURE 1-4 The Kinect infrared projector and camera.

The projector is the left-hand item in the Figure 1-4. It looks somewhat like a camera, but in fact it is a tiny infrared projector. The infrared camera is on the right side of Figure 1-4. In between the projector and the camera is an LED that displays the Kinect device status, and a camera that captures a standard 2D view of the scene. To explain how the Kinect sensor works, I'll start by showing an ordinary scene in my house. Figure 1-5 shows my sofa as a person (okay, a camera) might see it in a room.

FIGURE 1-5 My sofa.

In contrast, Figure 1-6 shows how the Kinect infrared sensor sees the same view.

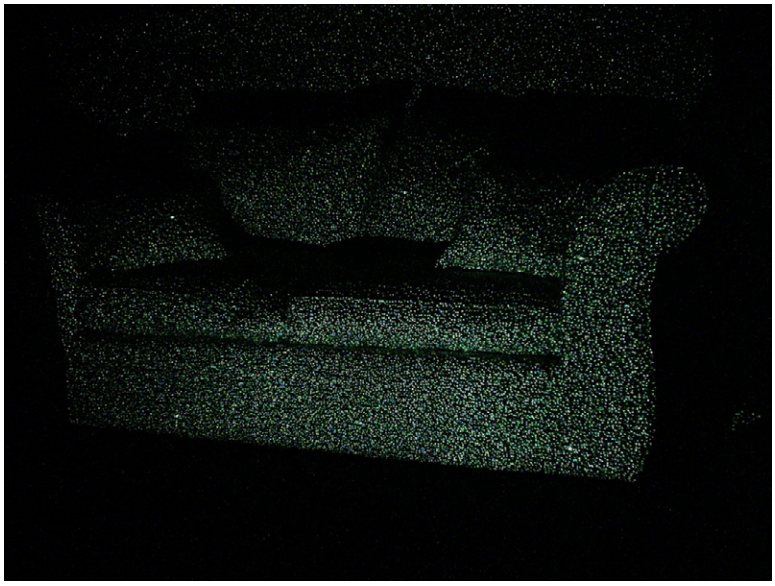


FIGURE 1-6 The sofa as the Kinect infrared sensor sees it.

The Kinect infrared sensor sees the sofa as a large number of tiny dots. The Kinect sensor constantly projects these dots over the area in its view. If you want to view the dots yourself, it's actually very easy; all you need is a video camera or camcorder that has a night vision mode. A camera in night vision mode is sensitive to the infrared light spectrum that the Kinect distance sensor uses.

Figure 1-6, for example, was taken in complete darkness, with the sofa lit only by the Kinect. The infrared sensor in the Kinect is fitted with a filter that keeps out ordinary light, which is how it can see just the infrared dots, even in a brightly lit room. The dots are arranged in a pseudo-random pattern that is hardwired into the sensor. You can see some of the pattern in Figure 1-7.

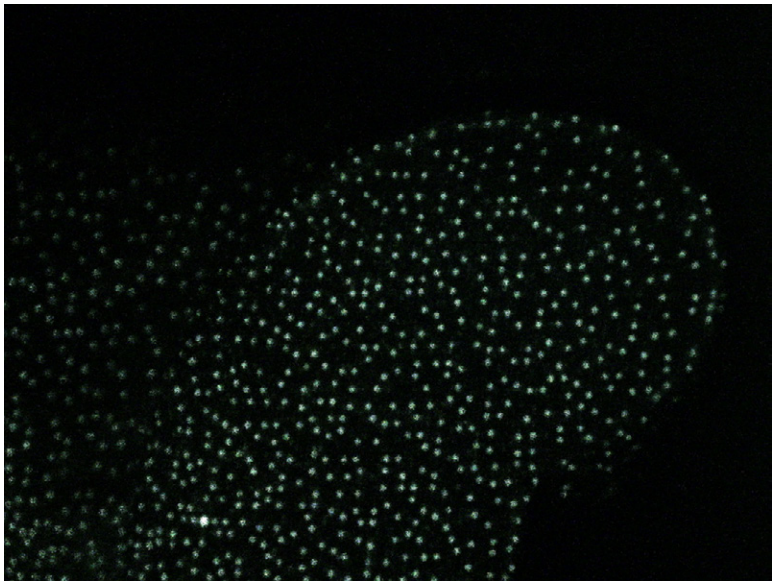


FIGURE 1-7 The dot pattern on the sofa arm.

A pseudo-random sequence is one that appears to be random, but it is actually mechanically generated and easy to repeat. What's important to remember here is that the Kinect sensor "knows" what the pattern looks like and how it is drawn. It can then compare the image from the camera with the pattern it knows it is displaying, and can use the difference between the two to calculate the distance of each point from the sensor.

To understand how the Kinect does this, you can perform a simple experiment involving a darkened room, a piece of paper, a flashlight, and a helpful friend. You need to adjust the flashlight beam so it's tightly focused and makes a small spot. Now, get your friend to stand about 5 feet (1.5 meters) away from you, slightly to your right. Ask your friend to hold the paper to the front of you, holding the torch in your left hand, shine the torch dot onto the piece of paper. Now ask your friend to move forward toward you. As the person comes closer, you will see that the dot on the paper moves a little to the left because it now hits the paper before it has traveled quite as far to the right.

Figure 1-8 shows how this works. If you know the place you are aiming the dot, you can work out how far away your friend is by the position of the dot on the paper. The impressive thing about the Kinect sensor is that it performs that calculation for thousands of dots, many times a second. The infrared camera in the Kinect allows it to "see" where the dot appears in the image. Because the software knows the pattern that the infrared transmitter is drawing, the hardware inside the Kinect does all the calculations that are required to produce the "depth image" of the scene that is sent to the computer or Xbox.

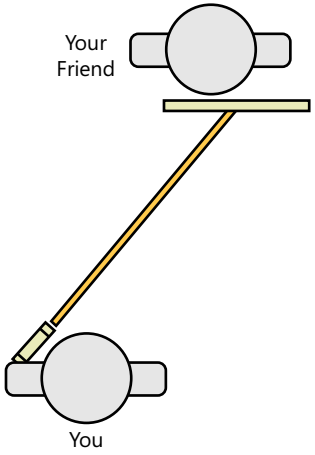


FIGURE 1-8 Showing how the Kinect distance sensor works.

This technique is interesting because it is completely different from the way that humans see distance. Each human eye gets a slightly different view of a scene, which means that the closer an object is to a human, the greater the difference between the images seen by each eye. The brain identifies the objects in the scene, determines how much difference there is between the image from each eye, and then assigns a distance value to each object.

In contrast, the Kinect sensor shines a tightly focused spot of light on points in the scene and then works out how far away that point is from the sensor by analyzing the spot's reflection. The Kinect itself doesn't identify any objects in a scene; that task is performed by software in an Xbox or computer, as you'll see later.

The Kinect Microphones

The Kinect sensor also contains four microphones arranged along the bottom of the bar. You can see them in Figure 1-2: two on the left and right ends, and two more on the right side of the unit. The Kinect uses these microphones to help determine from where in a room a particular voice is coming. This works because sound takes time to travel through air. Sound travels much more slowly than light, which is why you often hear a thunderclap long after seeing the corresponding bolt of lightning.

When you speak to the Kinect sensor, your voice will arrive at each microphone at different times, because each microphone is a slightly different distance away from the sound source. Software can then extract your voice waveform from the sound signal produced by each microphone and—using

the timing information—calculate where the sound source is in the room. If several people are in a room with the Kinect, it can even work out which person is talking by calculating the direction from which their voice is coming, and can then "direct" the microphone array to listen to that area of the room. It can then remove "unwanted" sounds from that signal to make it easier to understand the speech content.

From a control point of view, when a program knows where the speech is coming from (perhaps by using the distance sensor), it can direct the microphone array in that direction, essentially creating a software version of the directional microphones that are physically pointed at actors to record their voices when filming motion pictures.

Recognizing People with Kinect

One very popular use for the Kinect sensor is recognizing and tracking people standing in front of it. The Kinect sensor itself does not recognize people; it simply sends the depth image to the host device, such as an Xbox or computer. Software running on the host device contains logic to decode the information and recognize elements in the image with characteristic human shapes. The software has been "trained" with a wide variety of body shapes. It uses the alignment of the various body parts, along with the way that they move, to identify and track them.

Figure 1-9 shows the output produced by the body-tracking software as a "stick figure" with lines joining the various elements.

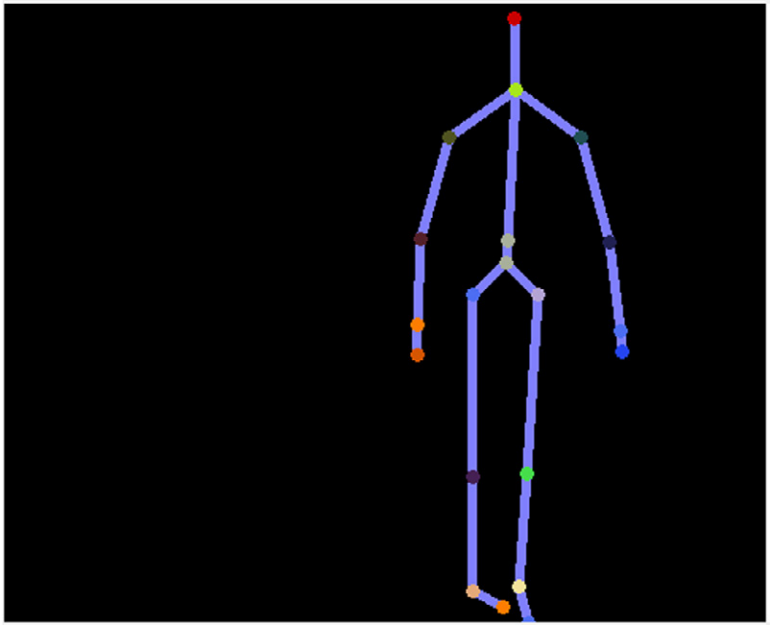


FIGURE 1-9 Skeleton information retrieved using the Kinect software.

The Kinect software can also recognize the height and proportions of a particular person. For example, this feature lets Xbox Live users "train" their Xbox so it recognizes them when they walk into a room.

Programming the Kinect

The software described in the previous sections, and which you'll see more of in this book, is called the *Kinect for Windows Software Development Kit* (SDK). Installing the SDK lets you write programs that use the power of the Kinect at different levels. You can obtain direct access to the low-level video and depth signals and create applications that use that low-level data, or you can make use of the powerful library features built into the SDK that make it easy for a program to identify and track users.

You can download the Kinect for Windows SDK for free. The SDK provides a set of libraries that you can add to your own programs and games so they can use the sensor. The SDK also contains all the drivers that you need to link a Kinect to your computer.

You can use the Kinect SDK from a *managed code* programming language (such as C# or Visual Basic.NET) or from unmanaged C++. The SDK provides a set of objects that expose properties and methods you can use in your programs. The following chapters explore how you can write programs that use these objects to create some novel and fun programs that support completely new ways of interacting with a computer.

The next chapter describes how to install the SDK on your computer and get it connected and talking to the Kinect.

Kinect for Xbox and Kinect for Windows

You can write programs that use either the Kinect for Xbox sensor or the Kinect for Windows sensor. The Kinect for Xbox sensor has been set up to allow it to be most effective when tracking the figures of game players. This means that it can track objects that are up to 12 feet (4.0 meters) away from the sensor but cannot track any objects that are closer than 24 inches (80 cm). The Kinect for Windows sensor has been set up to allow it to track a single user of a computer, and it has much better short-range performance as it is able to track objects as close to the sensor as 12 inches (40 cm).

The Kinect for Windows SDK was, as the name implies, primarily created for use with the Kinect for Windows sensor, but it will also work with an Xbox 360 Kinect sensor. Microsoft engineers will provide support into the future for Xbox Kinect from this SDK, but for best results, particularly if you want to track objects very close to the sensor bar, you should invest in a Kinect for Windows sensor device. The Kinect for Windows device can even track individual finger movements and gestures of the computer user.

The bottom line is that if you have an Xbox 360 with a Kinect device attached to it, you can use that sensor to have some fun learning how to create programs that can see, measure distance, and hear users. However, if you want to get serious about providing a product of your own that is based on the Kinect sensor, you should target the Kinect for Windows device. If you want complete details of how this all works, read the detailed End User License here:

http://www.microsoft.com/en-us/kinectforwindows/develop/sdk-eula.aspx

Summary

This chapter gave you a look inside the Kinect sensor so you could see (without having to take your own Kinect apart) how complex it is. You saw that the Kinect contains two cameras (one infrared camera and one video camera) and a special infrared transmitter that produces a grid of dots that measure the distance of objects from the Kinect and to compose a "depth map" of the image. You also learned that the Kinect sensor contains four microphones that can be used to remove background noise from an audio signal and to listen to sound from particular parts of a room.

You also saw that the Kinect sensor sends this data to a host device (Xbox or computer), which then processes the data in various ways, including recognizing the position, movement, and even the identity of people in front of the Kinect.

You also found out that two Kinect sensor bars are available, both of which can be used with the Kinect for Windows Software Development Kit (SDK). The Kinect for Xbox device has a good long-range performance for tracking game players, and the Kinect for Windows device has been optimized for shorter-range tracking so that a single computer user can use it to interact with a system that is nearby.

Chapter 2

Getting Started with Kinect

After completing this chapter, you will:

- Identify any prerequisites to work with Kinect on your computer
- Have installed the Kinect for Windows SDK on your computer
- Have connected the Kinect sensor bar and tested it on your machine

Kinect for Windows SDK Prerequisites

THE KINECT FOR WINDOWS SDK fits alongside an installation of Visual Studio 2010 on your Windows computer. It works on Windows 7. In this section we will look at the things you need to have to get the best out of your Kinect sensor.

Kinect Device

It should come as no surprise that you will need a Kinect device and its power supply along with a USB port so you can plug it into your computer. You can use either of two Kinect sensor bars with the Kinect for Windows SDK. You can use a Kinect sensor from an Xbox console, or you can use a Kinect for Windows sensor that has been optimized for computer use. The examples in this book will work with either sensor bar.

It is best if the Kinect is given exclusive use of a USB connection—that is, if you have a USB hub with your webcam, printer, and external hard disk plugged into it, you should not add the Kinect to the hub as well. The Kinect sensor can produce a lot of data, and it works best if it has exclusive use of its own USB connection.

Note You should plug the Kinect sensor into your computer *after* you have installed the Kinect for Windows SDK. When the SDK is installed, it also adds the USB drivers needed for Kinect; these are not provided as part of a standard Windows 7 installation.

Visual Studio

Before you install the Kinect for Windows SDK, you must make sure that you have Visual Studio 2010 installed on your machine. The SDK can be used with either C++, C#, or Visual Basic .NET. This text will focus on the use of C# to create managed applications that use the sensor, but the fundamentals of the way the libraries present data to your programs are the same. You can use any version of Visual Studio 2010, including those that are available for free from the Visual Studio Express website:

http://www.microsoft.com/express

DirectX Studio

Some of the C++ examples that are supplied with the Kinect SDK make use of the DirectX graphics SDK. If you want to compile and run these programs, you will need to have the DirectX SDK installed. You can download the SDK from here:

http://msdn.microsoft.com/en-us/directx

There is no need to install this SDK if you only plan to use the Kinect SDK from C# and Visual Basic .NET.

Installing the Kinect for Windows SDK

The Kinect for Windows SDK is a free download. The SDK also contains the USB drivers for the various elements inside the Kinect sensor itself. You can find the Kinect for Windows SDK at the Kinect for Windows website:

http://kinectforwindows.org

This site also contains links to detailed descriptions of the Kinect and other useful resources.

Note Although the SDK is provided free of charge, this does not mean that it is free for commercial purposes. Using the Kinect SDK for personal experimentation is not a commercial purpose. It is also not a commercial purpose to use the Kinect SDK in the process of teaching or academic research, even if you are regularly employed as a teacher or professor or if you intend to apply for research grants through such research. However, if you intend to sell a product based on the Kinect device, you should read the License Agreement.

Installing the Kinect SDK

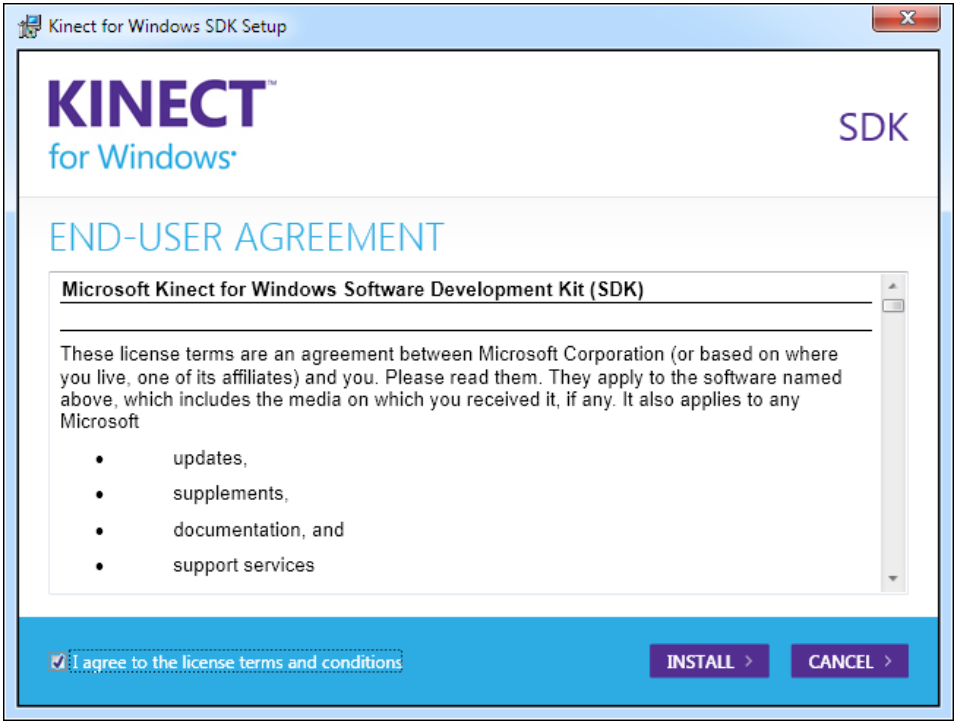
You should make sure that any older Kinect drivers that are not part of the Kinect system are removed from your system before you install the Kinect SDK. You should also make sure that Visual Studio 2010 is installed on your Windows computer (but not actually open) when you perform the install. If you have any problems you should check out the "Troubleshooting Your Kinect Installation" section at the end of this chapter. To install the Kinect SDK on your PC follow this sequence:

1. You can perform the installation of the Kinect SDK directly from the download webpage:

   http://www.microsoft.com/en-us/kinectforwindows/develop

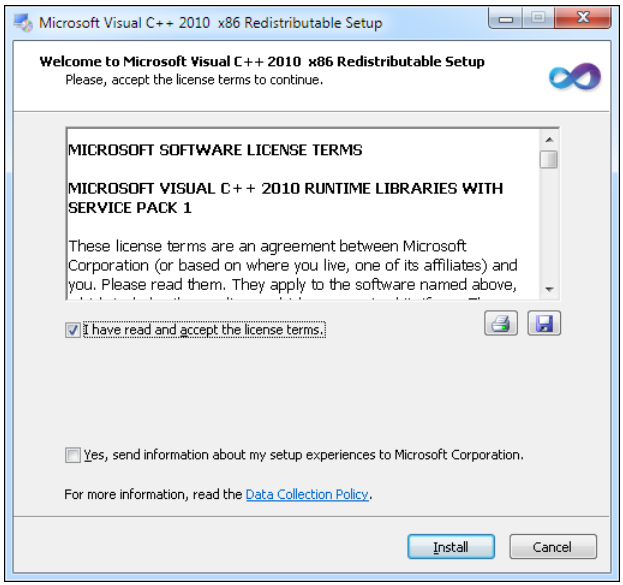
2. To do this you should click on the Download link to select the appropriate version for your system.

3. Your browser will ask you if you want to run or save the install file. You should select the Run option, as shown above. Click Run to start the installer, which will display the Welcome Screen as shown below.

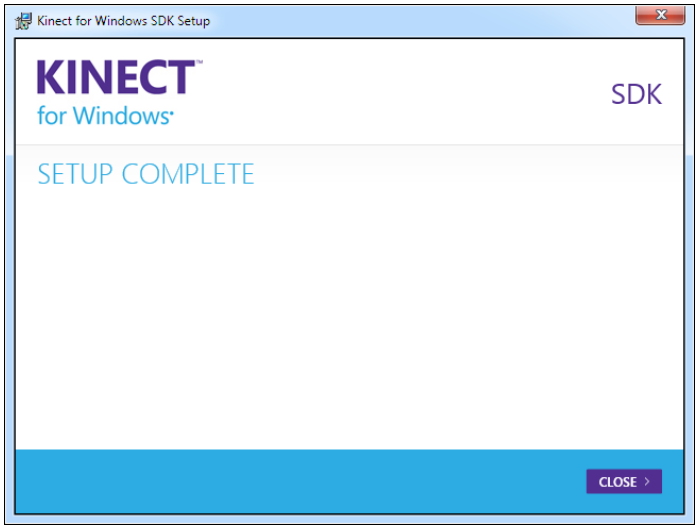


4. When the Install program starts, you will first see the Welcome Screen as shown previously. Select the tick box to accept the terms of the licensing conditions, and then click Install to begin installation.

5. Because this is a software installation on your computer, you may see a User Account Control dialog box confirming that you are going to allow the installer to make changes to the computer. Click Yes to continue.



6. The installation will now begin. During the installation it might be necessary to install some Visual C++ runtime components, as shown above. Just confirm the installation of each element in turn. Eventually you will see the completion dialog box, as shown below.



7. Once the installation has completed, you can create programs that use the Kinect for Windows SDK. You can also run programs that have been built using the Kinect SDK.

If you want to send your programs to Windows computer owners who will not be developing Kinect applications, the recipients must install the runtime version of Kinect for Windows. This contains the Kinect libraries and USB drivers, but it cannot be used to create new Kinect for Windows applications. The runtime version can be downloaded from the following website:

http://download.microsoft.com/download/E/E/2/EE2D29A1-2D5C-463C-B7F1-40E4170F5E2C/KinectRuntime-v1.0-Setup.exe

Connecting the Kinect Sensor Bar

After you have installed the Kinect SDK, you can connect the sensor bar to your computer. The Kinect sensor bar works with any Windows computer that has a USB connection.

Note Although you may not be using the Kinect sensor for playing games, you should still be mindful of how the sensor should be positioned and used. If you are using the sensor to detect movement and gestures, allow plenty of space around the device for operators to interact with the sensor. The sensor itself is not able to register depth information of objects that are closer than about 24 inches (800 mm), so make sure that it has a bit of breathing room in front of it.

Powering the Kinect Sensor

The Kinect sensor bar uses more power than is available from a standard USB connection. It needs about 1.5 amps of current, whereas a standard USB port on a computer is only able to supply 0.5 amp. A Kinect sensor bar can get the extra power in either of two ways. The newer, small Xbox 360 consoles have a specially modified USB connection on the back that can provide extra current. Owners of the older, larger Xbox 360s must use the Kinect power supply that is connected between the sensor bar and the console. The Kinect power supply allows use of the Kinect sensor bar with any device that has a standard USB connection.

The plug on the end of the wire coming from the Kinect sensor bar looks a bit like a USB plug, but in fact it is special and has one corner cut off so that it will not fit directly into a USB port in a desktop computer or laptop. If you force the Kinect plug into a standard USB socket, you will break the socket and do expensive damage to your system. Instead, use the Kinect power supply that is connected between the Kinect plug and the USB connection on your computer. The cable from the power supply includes a USB plug that can be fitted safely into a computer.

Note If you obtained your Kinect as part of an Xbox 360 and Kinect bundle, you might not have a Kinect power supply. In this case you will need to purchase a Kinect power supply to use the sensor on your computer.

Once you have positioned your sensor bar and connected it to a power source, you are ready to connect it to your computer.

Installing the Kinect Sensor USB Drivers

The very first time that you plug the Kinect sensor bar into your Windows computer, it will automatically install all the USB drivers that are required. To ensure that you get the latest version of the drivers, your Windows computer will contact Windows Update during the install. It is therefore a good idea to connect the sensor bar for the first time when your computer has a working Internet connection.

Figure 2-1 shows the results of a successful Kinect installation. If the drivers do not install successfully, this may be because you have older drivers on your machine that need to be removed. Take a look in the "Troubleshooting Your Kinect Installation" section at the end of this chapter for details of how to search for and remove these drivers.

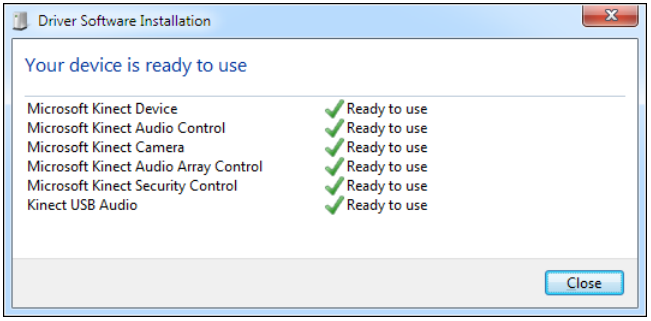


FIGURE 2-1 A successful driver installation.

Testing the Kinect Sensor Bar

The Kinect for Windows SDK is provided with some sample applications that you can use to demonstrate that the Kinect sensor is working correctly. Later in this book, we will take a look inside these applications to find out how they work.

The Kinect SDK Sample Browser

This sample allows you to demonstrate that the video and infrared cameras are working properly. It also gives a very good demonstration of the body-tracking abilities of the Kinect system. The program is supplied as part of the SDK and will be copied onto your computer when you install the

Kinect for Windows SDK on it. You can find the program on the Windows Start Menu in All Programs | Microsoft Kinect SDK v1.0 | Kinect SDK Sample Browser (Figure 2-2).

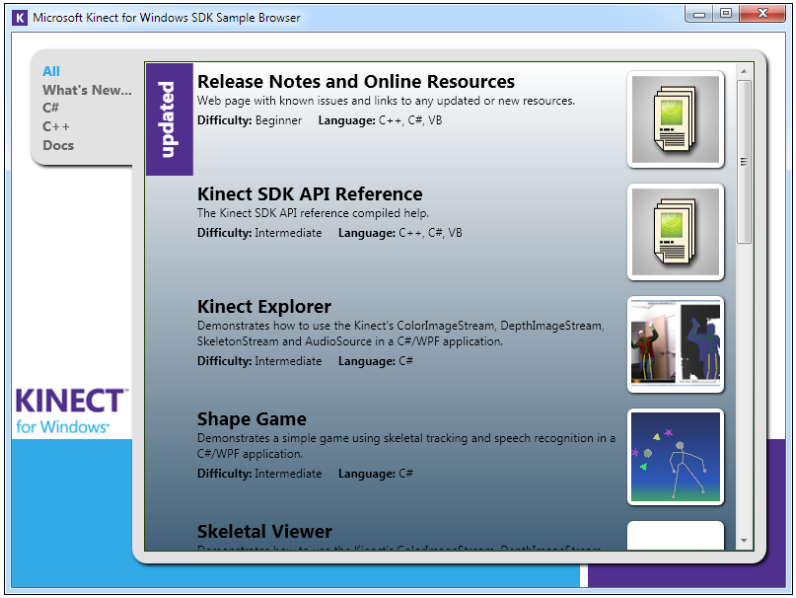


FIGURE 2-2 The Kinect SDK Sample Browser.

When you run the program, it displays a number of options that allow you to view documentation and run a number of sample programs, including the Kinect Explorer program (Figure 2-3).

FIGURE 2-3 Selecting the Kinect Explorer program.

If you click on the Kinect Explorer program, you get the option to read the documentation, install the sample code on your machine, and run the program.

Figure 2-4 shows the main screen displayed by Kinect Explorer. On the left is the image from the video camera, with the bones of any tracked skeletons displayed on top of it. On the right is the image from the "depth" camera. Points in the depth view that are different distances from the sensor are given different colors. The viewer also adds color to those parts of the depth view that have been identified as being part of a person in the scene. The display also shows the rate at which the display is being updated in frames per second (FPS). The sensors generate 30 frames per second. If the computer running Kinect Explorer is not fast enough to process and display each frame, this number will be lower.

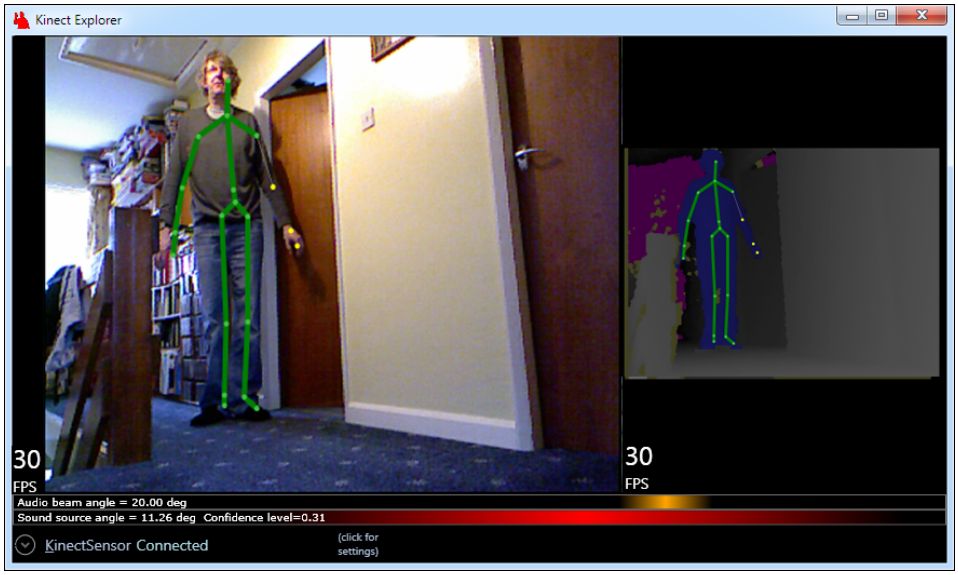


FIGURE 2-4 The Kinect Explorer main screen.

By clicking the down arrow at the bottom right of the screen, you can open the Settings menu, which allows you to configure the sensors in the Kinect device.

Figure 2-5 shows the options display. You can change the resolution of the color and depth cameras and also select the type of skeleton tracking that the program uses. You can also use the slider at the right side of the options to adjust the elevation angle of the sensor. This controls the motor in the base of the Kinect sensor and allows for adjustment of the angle of the sensor to get the best view of the scene.

FIGURE 2-5 Kinect Explorer with option screen.

The Kinect Explorer program also shows how Kinect uses the four microphones in the sensor bar to locate sound. It displays the angle from the sensor to any sound source that it detects as well as the angle of the audio beam that it has directed at the sound. If you make a noise in front of the sensor, you will see that the display changes to display where in front of the sensor the sound came from. In the display in Figure 2-5, the indicator underneath the right-hand "30 FPS" shows the direction in which the microphone is being aimed, with the broader area underneath giving the broad area from where the sound is coming.

Kinect Explorer provides a very good introduction to the capabilities of the sensor. You will discover how each part of the Kinect sensor works and how to use it from your programs in the coming chapters of this book.

Troubleshooting Your Kinect Installation

Most of my installations of the Kinect for Windows SDK and the sensor bar have had no problems. However, you might find the following troubleshooting tips useful.

Remove Old SDK Installations

Ensure that you have removed all the previous Kinect Beta SDKs. These can be removed using the Control Panel at Control Panel\Programs\Programs and Features.

Ensure That Visual Studio 2010 Is Installed but Not Running During Installation

During the Kinect SDK installation the installer will add some environment settings that are picked up by Visual Studio 2010. For this to complete successfully, it is important that Visual Studio is not running on the computer when the Kinect SDK is installed.

Ensure That There Are No Windows Updates in Progress

The installation process will modify some system files that might be in use during a Windows Update. Before you start the Kinect SDK installation, you should check in the Control Panel at Control Panel | System and Security | Windows Update to make sure that no updates are in progress. You also should check to see if any updates are waiting to perform a reboot.

Ensure That the Kinect Is Powered Correctly

If Kinect fails to install all the USB drivers when it is plugged into the Windows computer for the first time, it may be because the sensor bar is not receiving any power. Make sure that the Kinect power supply is plugged in and that the green light on the power connector is lit.

If the Kinect is showing a steady red light, this may mean that the power supply is not correctly connected. When the Kinect is working correctly, the indicator light on the front of the sensor bar should flash green.

Remove Any Old USB Drivers

Make sure that any older Kinect drivers that are not part of the Kinect system are removed from your system before you install the Kinect SDK. If you have any problems with the Kinect device not being properly recognized because you have used other drivers, you can do the following:

1. Ensure that the Kinect sensor is not connected to your computer.
2. Open up a new command prompt running as an Administrator user. The best way to do this is to click the Start button, type **CMD** into the search box that appears, and then hold down CTRL+SHIFT and press Enter. If you get this right you will be rewarded with a User Account Control dialog box asking for permission to allow the Command Processor to make changes to this computer. Click OK.
3. Next, you need to set an environment variable to tell the Device Manager that you want to see all the hardware devices registered for this computer, not just the ones that are active at the moment. In the Command box, give the following command:

```
SET DEVMGR_SHOW_NONPRESENT_DEVICES=1
```

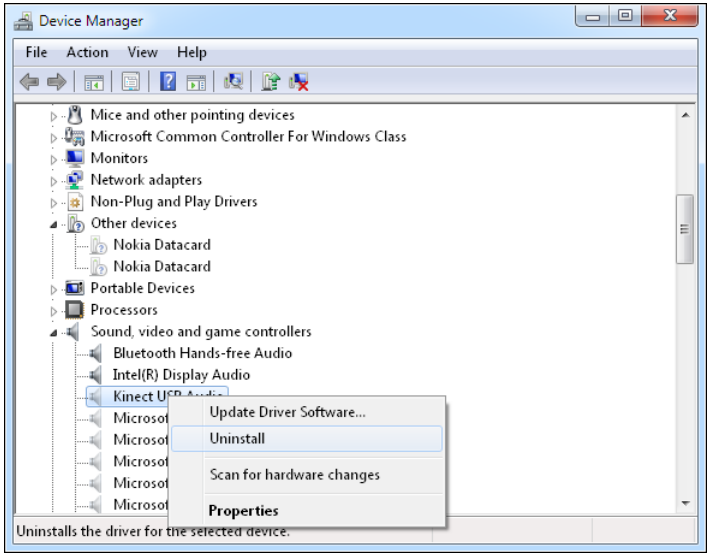
Note If you type this command incorrectly, you won't see an error of any kind, but the process won't work correctly as the Device Manager will not show you non-present devices.

4. Now you can give the command to start the Device Manager:

```
devmgmt.msc
```

5. Next, open the View menu and select Show Hidden Devices. This is actually quite fun, as now you will see every device that has ever been connected to your computer.

Note Your computer installation will look slightly different from this one.



6. If you use your machine like I use mine, you will see 50 or so different disk drives: one for every memory key that has been plugged in over the years. Look through the device tree for items with the word *Kinect* in the name, or the name of the package you are removing. Look in the "Human Interface Devices," "Sound, Video and Game Controllers," and "Universal Serial Bus Controllers" parts. To remove a driver, right-click on it in the list and then select Uninstall from the properties menu for that driver, as shown above. If the dialog that appears has a checkbox marked "Remove Driver Software Files," then you should select this so that the driver files are no longer on the machine.

Note You must be careful to remove drivers only for the Kinect sensor bar. If you are not sure which drivers are being loaded, you could plug the Kinect sensor in before you remove the driver and note what happens in the Device Manager when you do this. Drivers that become active at this point should be removed.

7. Once you have removed all the drivers, exit Device Manager and close the command prompt. Now you can plug in the sensor bar and the latest versions of the drivers should be loaded.

Summary

In this chapter you have seen how to get a Kinect sensor bar working with a Windows 7 computer and had a quick glimpse of its capabilities. In the next chapter you will write some code of your own to use the signals that the sensor bar produces.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

About the Author

Rob Miles wrote his first computer game on the original Commodore PET in Microsoft Basic, after learning to program some time before that at school, where he began by writing his first programs on cards using a hand punch, posting them off to a distant mainframe and getting a message back two weeks later that he'd omitted a semicolon. A good many years have gone by since then. He's still omitting semicolons, but the turnaround has improved quite a bit.

Rob has been at the University of Hull in the United Kingdom for over 30 years now, moving from the Computer Center to Electronic Engineering to Computer Science, where he teaches programming (in C# of course) and software engineering, among other things. He also had a hand in quite a few industrial projects, and considers it a matter of great personal pride to be the man who wrote the software that puts the date stamps on Budweiser beer cans, as well as many other products. Rob has also been known to turn out bad verse, the highlight of this being a whole page of poetry for The Independent (a British newspaper). He is a Microsoft Most Valuable Professional (MVP) for Windows Phone and has been a judge and competition captain for the Imagine Cup Software Design Challenge for a few years.

Rob lives happily in East Yorkshire in the United Kingdom with number one wife Mary (she calls him "husband zero") and a pinball machine. His kids, David and Jenny, return every now and then so that they can play happy families properly. You can find out more about Rob's interesting times at *www.robmiles.com*.

What do you think of this book?

We want to hear from you!

To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Tell us how well this book meets your needs—what works effectively, and what we can do better. Your feedback will help us continually improve our books and learning resources for you.

Thank you in advance for your input!

Microsoft® Press