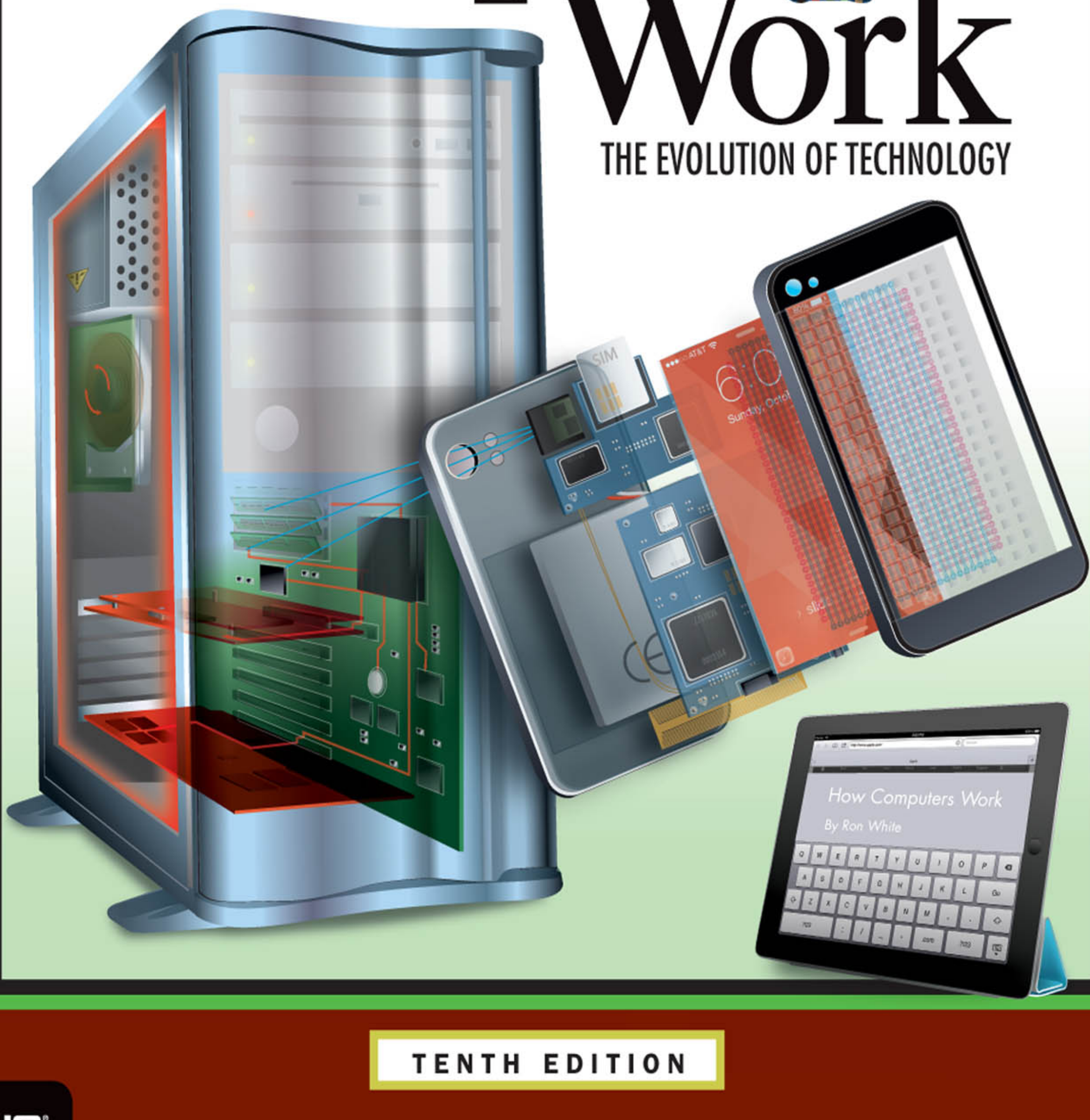
NATIONAL BESTSELLER
MORE THAN 2,000,000 How It Works SOLD

BY RON WHITE
ILLUSTRATED BY TIM DOWNS

How Computers Work

THE EVOLUTION OF TECHNOLOGY



TENTH EDITION

QUE

SPECIAL OFFER ONLY $4.99!

How Computers Work

Interactive Web Version

The *How Computers Work, Interactive Web Version* is an enhanced digital copy of the *How Computers Work* print book, which can be accessed via a web browser. It includes many new and exciting features that take the one-of-a-kind *How Computers Work* print book experience to the next level. Here's a look at just a few of the new features you'll find in the *Interactive Web Version:*

Designed for the Web—This new digital edition of *How Computers Work* is built from the ground up to work in today's most popular web browsers. Whether you use Windows, Mac, or a tablet, you'll find a viewing experience that works for you.

Animations—See first-hand how morphing software morphs a photo or how video compression and resolution changes the quality of a digital video. Watch as an interactive firewall blocks dangerous data, while letting safe packets through. Have you ever wondered just how a transistor physically controls the flow of electrons to turn itself on and off? You'll see it in action here!

Interactive Elements—Directly interact with more than a dozen of the topics covered in this book. Drag data into a processor to see how it divides up a work load and computes an answer. Or use a virtual laser to photoetch a micochip. You'll even control the flow of light through an LCD monitor as you learn how it generates an image!

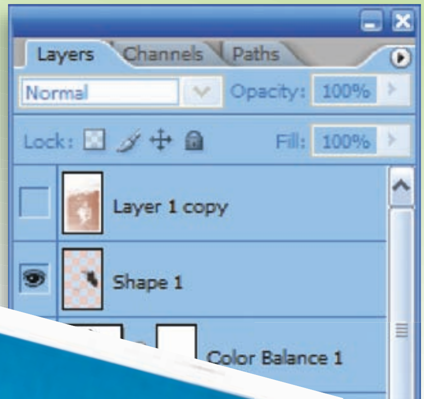
How to Purchase the Interactive Web Version:

Register this eBook to unlock a single-use discount, which will allow you to purchase a copy of the Interactive Web Version of *How Computers Work* (ISBN: 9780133096781) for only $4.99, a savings of $25!*

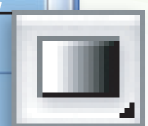
Follow the steps below:

1. Go to quepublishing.com/register and log in or create a new account.
2. Enter the ISBN: 9780789749840.
   (NOTE: please enter the print book ISBN provided to register the eBook you purchased)
3. Answer the challenge question as proof of purchase.
4. Click on the "Access Bonus Content" link in the Registered Products section of your account page, to be taken to the page where your discount code is available.
5. Go to www.quepublishing.com/title/9780133096781.
6. Add the product to your Cart.
7. Click on "Proceed To Checkout".
8. Enter your discount code in the "Enter a New Discount Code" box.
9. Complete payment information.

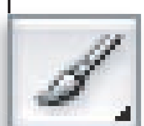
Discount may not be combined with any other offer and is not redeemable for cash. Offer subject to change.



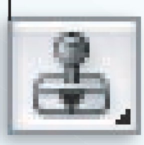
Layers provide a way of working on a duplicate of the photograph within the same file and then controlling how changes are blended into the original picture. Two duplicate layers of the washed-out image that emerged from color changes were *multiplied* to increase the contrast and color depth of the photo.



Gradient tool fills a sky that has lost all hint of color, permitting the blue to fade as it approaches the horizon.



Airbrush brings out clouds that have been submerged by the blue gradient. The airbrush also adds a hint of eyeballs that have been lost entirely in the shadows of the mother's eyes.



Cloning tool covers bigger and more complex flaws by copying, through a sort of artistic worm-hole, good portions of the photo to replace flawed areas with the same control you have using a brush. Here, some of the dark trees on the right were replaced with light trees from the left of the photo.

Sharpen tool restores definition to edges that have become blurred through fading or by the retouching itself.

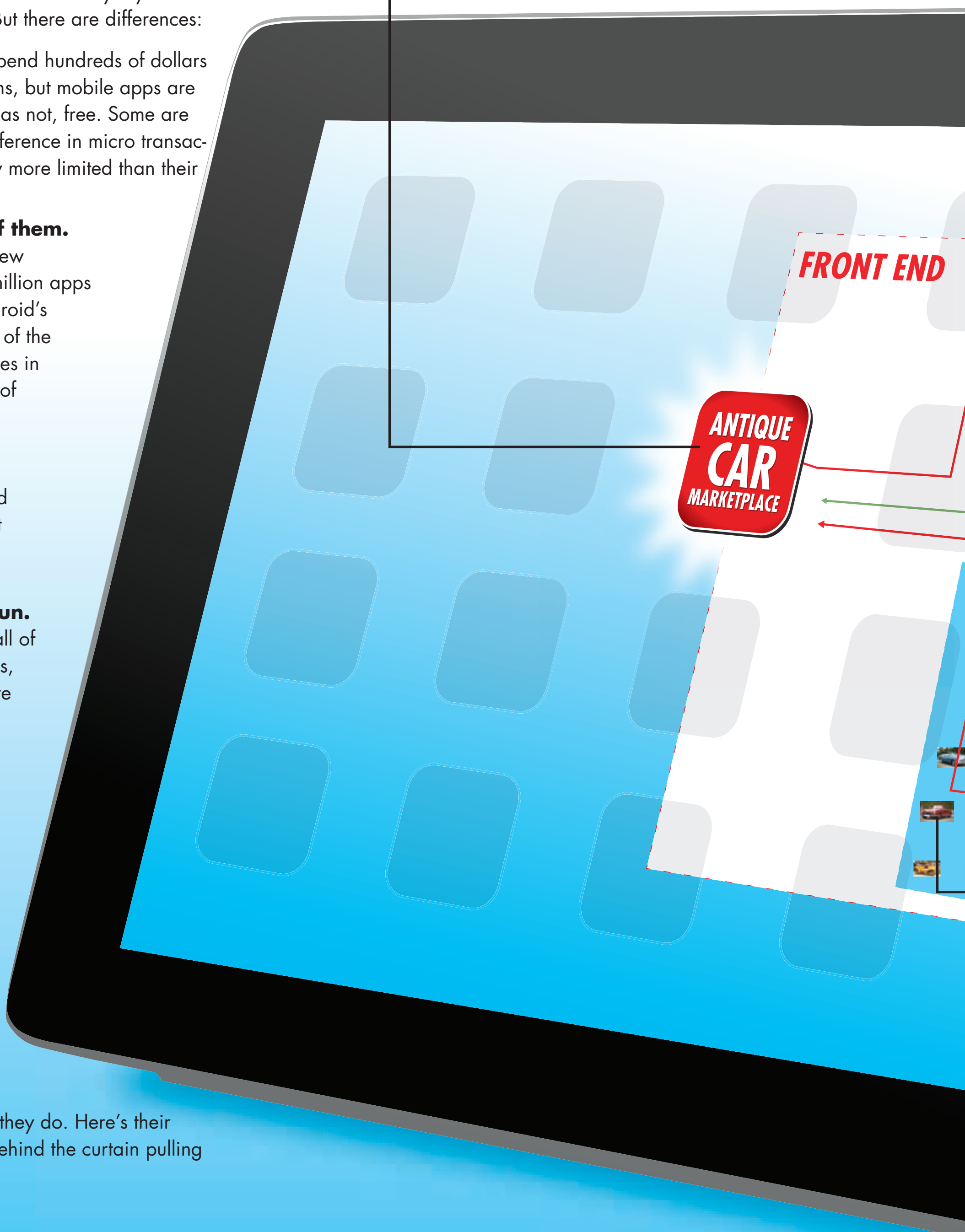
Variations are used as a final touch so the retoucher is able to see and choose from a selection of thumbnail versions of the same photograph in which hues and brightness are slightly varied. This enables the retoucher to see, at a glance, which variation produces the most pleasing result.

How Apps Fuel Mobile Devices

THE APPS that run on your mobile devices are the smaller offspring of **applications**, which have had their heyday on the big desktop computers and laptops. But there are differences:

- **Apps are cheap.** You could spend hundreds of dollars for Windows or Mac applications, but mobile apps are at most a few bucks or, as often as not, free. Some are ad-driven, some make up the difference in micro transactions, and others are functionally more limited than their desktop counterparts.
- **There are a heck of a lot of them.** As I'm writing this, a study by New Fossil had found there are 1.5 million apps in the Apple App Store and Android's Google Play. On average, each of the more than two billion smartphones in the world run an average of 42 of these apps. Of course, with this many apps it should come as no surprise that one in four apps is abandoned after the first use and more than 60 percent of apps at the Apple Store are never even downloaded.
- **They're a heck of a lot of fun.** Apps are not made to manage all of L.A.'s traffic lights, detect cancers, or manage a space station. There are plenty of useful productivity apps out there, but they're generally for a single, dedicated purpose. Still more are, for the most part, intended for enjoyment—watching movies, listening to music, social networking, and game playing. When you consider that the processing power, storage capacity, and RAM in any mobile device is dwarfed by a cheap desktop PC from the last century, it's a wonder they perform as well as they do. Here's their secret: There's a big computer behind the curtain pulling the strings.

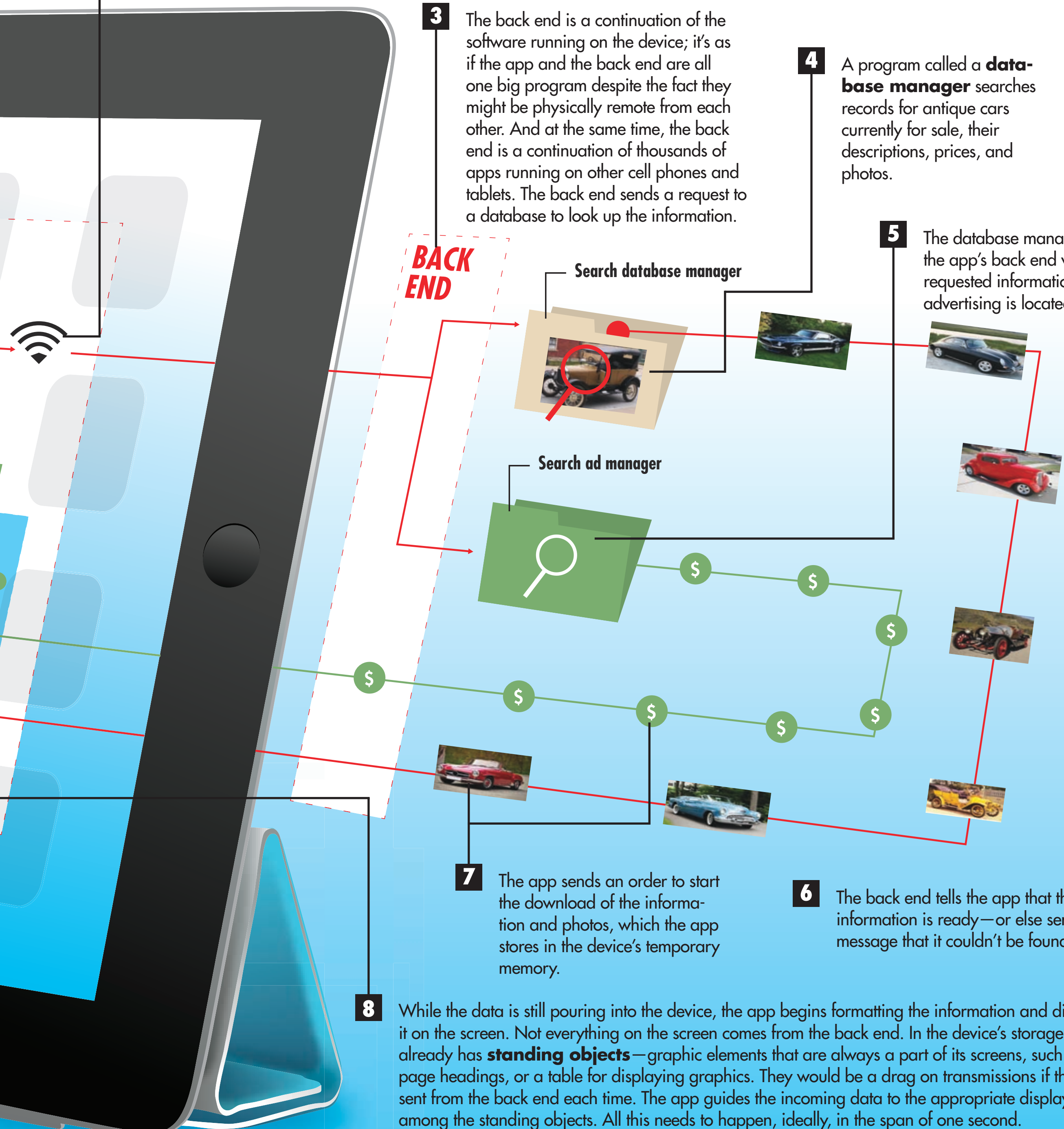
1 As an example, let's say a user taps a selection on his device that links to a display of antique cars for sale.



2 The device's front side, the computing circuitry built into it, sends the request over the Internet to the back end, a system of powerful computers run by Apple, Google, Microsoft, or the developers who created the app. If the device has a wireless connection, that's what the front and back ends use to communicate. If not, the phone resorts to its 3G or 4G cellular connection.

3 The back end is a continuation of the software running on the device; it's as if the app and the back end are all one big program despite the fact they might be physically remote from each other. And at the same time, the back end is a continuation of thousands of apps running on other cell phones and tablets. The back end sends a request to a database to look up the information.

4 A program called a **database manager** searches records for antique cars currently for sale, their descriptions, prices, and photos.



5 The database managers tell the app's back end where the requested information and the advertising is located.

6 The back end tells the app that the information is ready—or else sends a message that it couldn't be found.

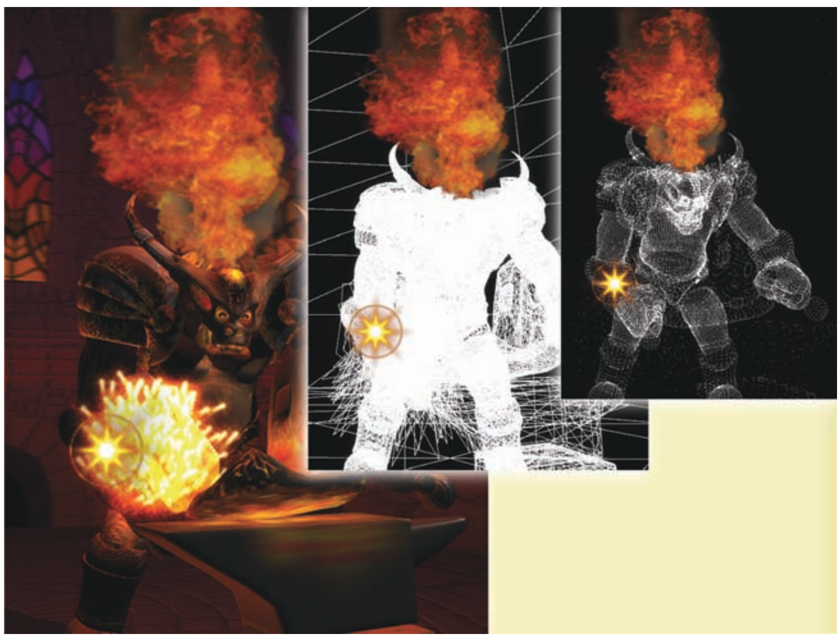
7 The app sends an order to start the download of the information and photos, which the app stores in the device's temporary memory.

8 While the data is still pouring into the device, the app begins formatting the information and displaying it on the screen. Not everything on the screen comes from the back end. In the device's storage, the app already has **standing objects**—graphic elements that are always a part of its screens, such as a logo, page headings, or a table for displaying graphics. They would be a drag on transmissions if they were sent from the back end each time. The app guides the incoming data to the appropriate display locations among the standing objects. All this needs to happen, ideally, in the span of one second.

CHAPTER

7

How Games Create New Worlds



WHEN I BOUGHT my first computer waaaay back in 1980, I told my wife that I could use it to make money. And as it turned out that was true. Within a few months I had snagged an assignment to create a database that more than covered the $3,000 I had paid for my Eagle II.

But deep in my heart of hearts, I wanted a computer so I could play games. I had seen some games on a friend's Apple, and in computer stores watched open-mouthed as Flight Simulator created a flexible, organic world in real time. Forget about word processing and electronic spreadsheets. Games on a computer were miraculous! It was as if the PC brought to life all the imaginings we had as children having a tea party with dolls or when we ran with our arms held straight out from our sides, pretending we were piloting a Northrop Black Bullet.

I had to get me one of those computers.

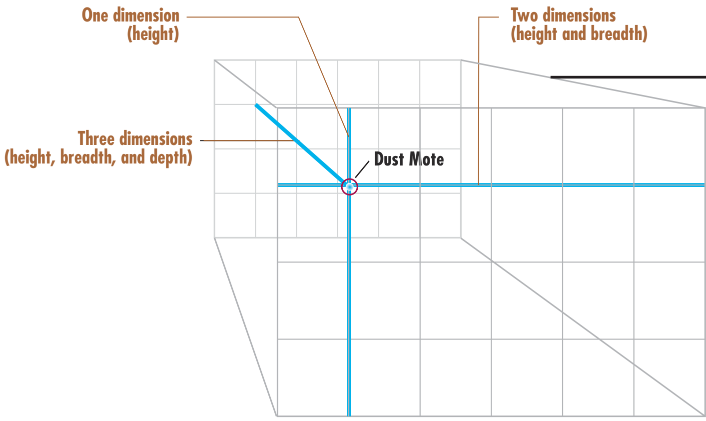
But as it turned out, my Eagle didn't handle graphics, as advertised, unless you added a circuit board and used programs no one had heard of then, or since. I was stuck playing **adventure games**, such as *Zork*, which were all text. I played by typing commands such as "Go north" and "pick up sword." (See the pseudocode for an adventure game in the Chapter 5 spread, "How a Program is a Roadmap.") Actually the adventure games weren't that bad, but it wasn't what I had in mind when I bought a computer for…work.

Today's computer games have made me forget that initial disappointment. Now graphics are photorealistic to the point of showing each blade of grass in a meadow being blown by a breeze independently of how the other thousands of blades are moving. At the same time games have captured realism, they have also transcended reality to add new spatial dimensions and to create entire worlds on which millions of people can play and interact with each other in real time via the Internet.

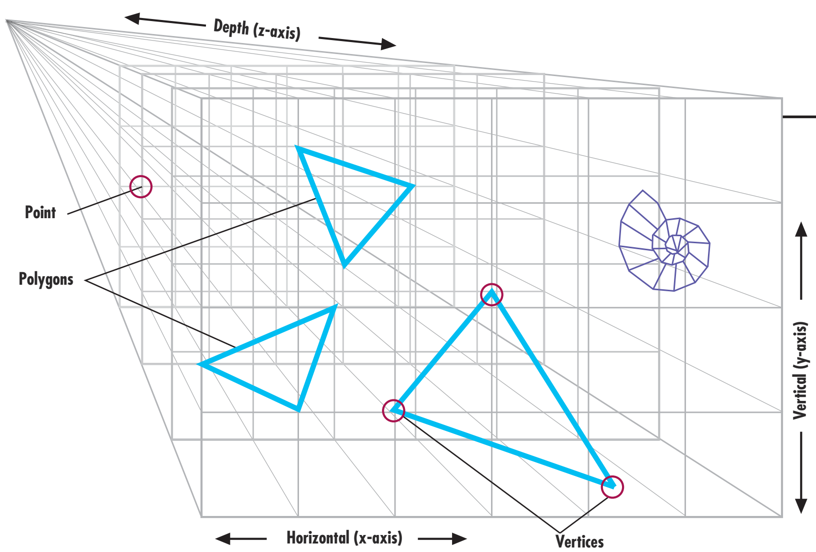
And the wonder of wonders is that you no longer need a computer—at least not a computer by the old definition of a desktop box, monitor, and keyboard. You don't even need a dedicated game console, such as the X-Box or PlayStation. You can now play just as many computer games on the new definition of computers, which includes smartphones and tablets. Angry Birds, anyone?

How Computers Plot a 3D World

Finding Your Place in Three Dimensions

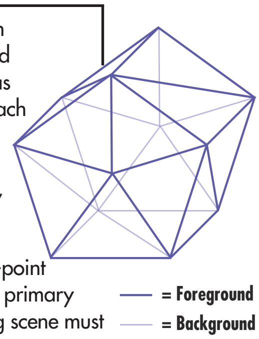


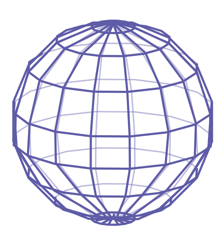
1 Imagine a speck of dust floating near your head. As long as it stays put, you have no problem telling someone exactly where the speck is: Six and a half feet off the floor, 29 inches from the north wall, and a foot from the west wall. You need only three numbers and some agreed-upon starting points—the floor, the walls—to precisely pinpoint anything in the universe. (For our purposes, we'll ignore that curved space thing Einstein came up with.) That's how 3D games got started, by using three numbers to determine the position of all the important points in the graphic rendition of the world they're creating. Of course, today's PC games are pinpointing 47 billion dots a second, but the principle's the same as you are putting numbers to the dust mote's location.



2 In 3D games, as in life, we locate points along three axes: horizontal, vertical, and distal—the x-axis, y-axis, and z-axis, respectively. In three-dimensional space as well as games, three points are all that's needed to define a two-dimensional plane. 3D graphics create entire worlds and their populations from 2D **polygons**, usually triangles, because they have the fewest angles, or **vertices**, making them the easiest and quickest polygon to calculate. Most times, even a square, rectangle, or curve consists of combinations of flat triangles. (The vertices, as you'll soon see, are mere anchor points for straight lines.)

3 Three-dimensional objects are created by connecting two-dimensional polygons. Even curved surfaces are made up of flat planes. The smaller the polygons, the more curved an object appears to be. The graphics processing unit on the video card (or cards) has a **geometry engine** that calculates the height, width, and depth information for each corner of every polygon in a 3D environment, a process called **tessellation**, or **triangulation**. The engine also figures out the current camera angle, or vantage point, which determines what part of a setting can be seen. For each frame, it rotates, resizes, or repositions the triangles as the viewpoint changes. Any lines outside the viewpoint are eliminated, or **clipped**. The engine also calculates the position of any **light sources** in relation to the polygons. Tessellation makes intense use of floating-point math. Without video cards with processors designed specifically for 3D graphics, the primary Pentium and Athlon CPUs in the computers would be woefully overtaxed. A changing scene must be redrawn at least 15 to 20 times a second for the eye to see smooth movement.





Polygon 3D object