In the last chapter, we considered television as a model for investigating the influences that video games might have on our culture. We'll use the same strategy here, focusing on learning and education. First we'll look at the allegedly adverse effects that TV may have on learning and the potential educational advantages of TV. Then we'll discuss computer games and education, arguing that the educational potentialities of the games are enormous. Even now it's easy to identify a number of indirect educational benefits the games provide.

TV as Teacher

Numerous surveys indicate that the average child watches about five hours of television a day. Between the ages of two and five, very impressionable years, many children spend almost two-thirds of their waking life watching TV. Older children often spend more time watching TV than they do attending school. Are children learning anything useful during all these hours? Some critics suggest that the answer is no, that little formal learning takes place while we watch since TV doesn't really engage our minds. Learning is done most efficiently when we are forced to use the material being presented — when we have to repeat it, study it, write it down, paraphrase it, and digest it. The fact that most of us can't remember anything from a TV show that we saw a year or two ago — or even more recently — suggests how little we actually retain from TV.

Does this kind of evidence prove that we don't learn from watching TV? No, and in fact quite a different view comes from Gavriel Salomon, an educational psychologist at the Hebrew University of Jerusalem, who claims that what children learn from TV depends on the show.\(^1\) Citing the work of Yale psychologist Jerome L. Singer,\(^2\) Salomon points out that children attend more to the fast pace of Sesame Street but learn more, or at least remember more, from Mr. Rogers' Neighborhood. Apparently the fast pace and changing of scenes in Sesame Street do not allow sufficient time for processing and memorizing the information, even though the children attend to it. In other words, the children may like the show better, but they do not learn more from it.

This doesn't imply that children learn nothing from Sesame Street. Salomon had the unique opportunity to study children who were first being exposed to the program when it was

---


brought to Israeli television in the fall of 1971. He was especially curious about whether exposure to the program enhanced the development of any specific skills.

Since the program was being broadcast nationally and was very popular, it was almost impossible to have a control group—a group of children who didn’t view it at all. Instead the children in Salomon’s study were allowed to watch the program as much as they wanted and were tested at several different times. In this way, children who watched the show a great deal could be compared to those who watched it very little.

The program was broadcast in English with a Hebrew voice-over, but despite this drawback, it had wide appeal. Over 90 percent of the children claimed to watch at least parts of Sesame Street, and about half claimed to watch the entire program all the time. Did the heavy viewers perform differently from the light viewers on tests of mental ability? The answer appears to be yes. The children who watched the show more tended to perform better on subsequent tests of mental skills such as letter matching, number matching, and picture-number matching than did their little colleagues who watched it less. In these tests a child selects the correct letter or number that matches a given letter, number, or picture. Sesame Street, of course, emphasizes just these sorts of skills.

In a later study, children were randomly assigned to watch either eight hours of Sesame Street on a wide screen or to watch an equal number of hours of adventure and nature films. They watched these films one hour a day for eight days and were then tested. In this experiment, the Sesame Street group performed significantly better than the adventure-film group on a number of mental tests. Thus we can conclude that the children do learn something (as opposed to nothing) from watching Sesame Street. At the very least they learn the specific skills that the program has been designed to teach, and they do so not only under controlled experimental conditions but also under normal viewing circumstances.

Do children learn any skills beyond those that are specifically taught by the TV program? Even if the answer to that question turns out to be yes, there will remain an inherent problem with learning from TV. This problem—the passive nature of the medium—is interesting in the context of this book, because it is a problem that is not inherent in computer-based learning systems such as video games.

One of the authors (GL) had some firsthand experience with the results of TV-bred students. One year, while teaching his statistics class at the University of Washington, GL noticed that the students seemed to be more passive than they had been in previous years. They didn’t ask questions; they didn’t respond as well to his jokes; in general, they just sat there listening to what he had to say, occasionally taking notes. At first, GL was baffled by this seemingly inexplicable change in his students’ behavior. One day, though, it dawned on him that this class was the first he had taught that had grown up watching Sesame Street. GL reasoned that this experience—hundreds of “TV as teacher” hours—may have substantially altered his students’ expectations about what the relationship between teacher and pupil was supposed to be.

With a passive teaching device, you can’t determine if or how well the child is responding. It isn’t possible to provide individual feedback. And you can’t easily tailor your teaching to the learner’s prior knowledge because you don’t have any idea how much prior knowledge the child has. Gavriel Salomon, among others, discovered that the levels of knowledge and skill that children bring with them to the viewing situation will affect if and how they benefit from television. But the television does not have this information about an individual viewer and thus, obviously, can’t be responsive to it.
Many different types of experiments confirm the importance of active involvement in the learning process. One classic study was done at MIT. In this study, two kittens from the same litter were placed in identical surroundings. One of them, the active kitten, could move around normally. Meanwhile, the active kitten’s brother, the passive kitten, was restrained in a gondola chair but was moved around in such a manner that its visual experiences were identical to those of the active kitten. When not being tested, both kittens spent their time in total darkness. After only ten days, the active kitten displayed normal visual behavior, while its passive sibling behaved as if it were blind. Eventually, after several days of normal experience, the passive kitten learned normal visually guided behavior. This experiment suggests that learning, in this case perceptual learning, depends a great deal on whether the animal has an opportunity to interact with its physical environment.

Computers as Interactive Devices

The idea that active involvement spurs learning goes back at least as far as the time of Socrates. Indeed, to interact with students is the reason teachers exist in the first place. However, there are many more learners around needing instruction than there are teachers to offer it.

Twenty or so years ago, educators began to realize that the computer, with its powerful interactive abilities, might be used to aid in instruction. Thus, the concept of computer-assisted instruction (or “CAI”) was born.


By the mid-1960s several CAI projects were underway. One of the most ambitious was the Stanford project, under the direction of Professors Patrick Suppes and Richard Atkinson. Thousands of elementary school students were involved, and a variety of scholastic subjects were taught. One, for example, was a drill-and-practice program in which the student would sit down at a teletype and type in a number and his or her first name. This information was transmitted via telephone lines directly to the computer at Stanford. The computer responded by typing the student’s last name and giving him or her an arithmetic drill for the day. When the lesson was completed, the computer typed the score, the elapsed time, and said goodbye, using the student’s first name (“Goodbye, Carol”). Students averaged from four to ten minutes per day at the teletype.

During the 1967–68 school year, almost 4,000 students at various elementary and junior high schools completed nearly 300,000 arithmetic lessons, covering such concepts as addition, fractions, and inequalities. Students in the program showed significant gains in the achievement of computational skills when compared to other groups of children who received traditional, noncomputerized methods of teaching. College students as well as elementary ones benefited from learning via CAI. In September 1967, thirty Stanford University students enrolled in a course of computer-assisted Russian, wherein they received instruction at computer-based terminals for fifty minutes a day, five days a week, throughout the entire academic year. Another group of students received regular classroom instruction. At the end of the first year, the computer students performed at a higher level. In addition, the motivational be-

MIND AT PLAY

For maintenance, the CAI system for the entire community came to a standstill. Also, there were resource problems. If the school system had a lot of resources, both financial and technical, that could be allocated for such an exotic enterprise, things were fine; otherwise, acquisition of a CAI system simply wasn’t possible. Thus an affluent, upper middle-class community such as Palo Alto, with a great deal of money to spend on education and brigades of Stanford scholars right next door interested in and working on CAI, found itself with computer instruction in abundance. But isolated, poor communities were out of luck.

Today, with the proliferation of cheap microcomputers, computer-based instructional programs have become much more available. Rather than being confined to large experimental projects, educational programs are being written and distributed by a large range of enterprises ranging from giant companies to small, basement entrepreneurs. Graphics are a central and universal feature of all computer systems, both large and small. And the distinction between games and educational programs has become much more blurred. This blurring is evident, for example, in Thomas Malone’s work, which, as we saw in chapter 2, was primarily concerned with how to make the educational experience enjoyable and intrinsically motivating for the learner.

We shall return to Malone’s work and discuss it specifically in the context of education in a later section of this chapter. For the moment, let us examine the work of an MIT researcher, Seymour Papert, who is currently one of the leading advocates of the computer as a tool for learning.

In a remarkable book called Mindstorms, Papert explicitly compares learning via television and learning via the computer. With television, the child is in the position of listening to

Learning from the Screen

to talk to the Turtle via computer commands, asking it to move forward or backward, right or left. The Turtle can be made to trace a square, a triangle, a rectangle, or a circle. Bill learned to use his own body to help decide how to move the Turtle. For example, when he wanted to make a circle, he would say to himself, “When you walk in a circle you take a step forward and you turn a little. And you keep doing it.” From this description, he was about to develop the necessary means to instruct the Turtle to move in a circle. Soon Bill was creating squares of different sizes, triangles embedded within circles, and complex architectural structures. By connection with movement and the navigational knowledge needed in his everyday life, Bill related to Turtle Geometry, as he did to his songs, far more than to multiplication tables. The mathematical knowledge that Bill had previously rejected finally found its way into his intellectual world.

For the most part, Papert’s demonstrations of the effectiveness of computer approaches to learning have been anecdotal. Furthermore, his research involves special kinds of computer games—those expressly designed for instructional purposes. It still remains to be shown that interacting with more ordinary computer games can teach people important things.

Video Games as Learning Devices

It would be comforting to know that the seemingly endless hours young people spend playing Defender and Pac-Man were really teaching them something useful. What, for example, about “eye-hand coordination,” which is becoming such a buzzword in video game magazines? Is the massive amount of practice in motor coordination going to be useful for anything
besides being a better video game player? As relevant research has not yet been performed, we can only speculate about direct educational benefits. But it seems safe to say that the games provide a number of interesting and important indirect benefits. We'll consider four such indirect benefits before turning to speculations about direct ones.

AN INTRODUCTION TO COMPUTERS

Virtually no one denies that a thorough and sophisticated knowledge of computers is going to be necessary for getting along in society from now on. This dictum applies a fortiori to young children, who are going to be dealing with computers all their lives.

Increasingly, courses on computers are being taught in school, particularly from the junior high level up. Suppose a school child is presented with a new object that he or she is supposed to learn about. What provides the motivation for such learning? One very strong motivating force accrues if the object already plays a prominent role in the child's life. Thus, we argue, children who play video games know that these games are computer-based, even if they have never seen an actual computer and only have a vague notion of what a computer is. However, when, sooner or later, they are presented an actual computer and are told that they have to learn about it, they don't do their learning in a vacuum. They know at least one thing about computers that is very important in their lives, namely, that with computers you can make video games. Thus, there is at least one thing about computers that interests the youngsters, and that provides the motivation to learn enough

---

6An example of the converse of this rule is seen in foreign language learning. American children typically have virtually no experience with speaking foreign languages. French, for example, is usually entirely unrelated to the rest of the children's lives, and they have rather little motivation to learn it. Partly for this reason, foreign language learning has become deemphasized in the U.S. over the years.

Learning from the Screen about how to program the computer to be able to create video games. It turns out, of course, that in the process of learning about how to program video games, they learn quite a bit about how the computer works in general and about how to write computer programs. This additional knowledge then provides the ability to write more general programs. Learning about computers is, in other words, a bootstrapping operation that involves an alternation of first finding out some interesting thing the computer can do and then acquiring the knowledge necessary to make the computer do it. This knowledge then provides the vision for more complex goals that require and motivate yet more knowledge, and expertise spirals upward. In any bootstrapping operation, however, there is always the problem of how to get it started in the first place. Often in educational settings, threats and cajoling are needed. In the case of computers, video games provide this start in a painless, indeed enjoyable, way.

We recently attended a party in a middle-class suburb of Seattle. At some point during the party, a group of grownups were in the basement playroom drinking and talking while, in the same room, a group of children were playing video games on a home computer. The adult discussion turned to the book we were writing, and the inevitable question was raised: Did the games that we were watching on the other side of the room teach anything useful? As it happened, we had spent that morning discussing the video-game-as-computer-introduction argument that we've just described. We presented the argument to our listeners, but it was received with a great deal of skepticism. In the middle of the discussion, twelve-year-old Scott, one of the video game players, came wandering over, software catalog in hand. We asked him why he liked video games and eventually inquired what he would choose if he could have anything in the catalog. Much to everyone's sur-
prise, he pointed not to a specific game but to the programming module. We asked him if he knew how to program, and he replied that he didn’t. “What is programming?” we asked, and he told us he wasn’t really sure. Did they have any computers in his school? No, but he’d study computers next year in junior high. Well, if computers weren’t a social focus in his school, if he didn’t know how to program computers and really didn’t know what programming was in the first place, why was the programming module what he most wanted? Scott replied that he did know that computer programs made video games work, and if he had the programming module, then he could learn how to program his own games. This reply provided such a perfect confirmation of our argument that the astonished adult onlookers refused to believe that we hadn’t set Scott up.

Will Scott go on to become an expert programmer? Of course, we have no way of telling. However, an example of a former video game fanatic who did is provided by another composite of several players whom we interviewed. Mark, as we shall call him, had spent three years and thousands of quarters playing video games in Los Angeles’s Westport Arcade. After his first year of playing, Mark became curious about how the games worked and started reading everything he could find on the topic. It became clear to him that what took place on the screen when he played the games didn’t happen by magic, but rather as a result of very sophisticated computer programming. His twenty-first birthday was coming up, and he begged his parents to buy him a computer so that he could try to create his own games. After working out a complex financial arrangement with his mother, he depleted his savings account to pay for half of an Apple II. Mark immediately learned to program his new toy by reading several books and soon had made himself a simple video game. In the process he had not only learned what a program is and how to write one, but he learned a great deal about how computer graphics are generated and how information is represented inside the computer.

One unexpected surprise for Mark’s parents was his desire to go back to school. He enrolled at a local junior college, majoring in computer science. To finance his education, Mark conducted a small window-washing business on the side, for which the computer came in handy. He wrote programs to keep track of his customers, the dates on which he had washed their windows, and whether their accounts were up to date. He wrote a program that sent out notices to his customers reminding them that it was time to have their windows washed again. His business flourished, even in a slow economy.

The experiences of Scott and Mark are being repeated throughout the country. Everywhere we turn, we seem to hear anecdotes of nervous, computer-shy parents being wheedled and cajoled by their children to buy computers. The children’s motivation is not always to be able to program video games, but it seems to be in a sizable number of cases.

Of course, a series of anecdotes doesn’t prove that video game playing leads to computer expertise. Perhaps only a few video game players will go on to be computer programmers. Or maybe the appropriate research will show us that many expert programmers never had anything but disdain for computer games. However, at the very least, video games are providing many people with a substantial introductory dose of computer technology.

EDUCATION-ORIENTED VIDEO GAMES

Educators have been trying to harness the astonishing motivational power of video games by designing educational computer games that resemble their arcade counterparts. Two pioneers in this area are Thomas Malone at the Xerox Palo Alto Research Center and John Frederiksen at the research firm of
Learning from the Screen

MIND AT PLAY

Bolt, Beranek and Newman in Cambridge, Massachusetts. Malone's work is primarily aimed at generating a theory of instruction based on the intrinsically motivating elements of video games, whereas Frederiksen has actually designed some games—in particular, some that will help people who have reading problems.

Malone, whose work was discussed in some detail in chapter 2, believes that, unlike the usual learning environment, computer games create "intrinsic motivation." Such intrinsic, or self-generated, motivation is due in part to the presence of goals, of challenge, and of fantasy.

A major ingredient of video games that makes them not only fun but also ideal for learning is the well-defined goal structure that they incorporate. Imagine, for example, a child who has to learn the locations of several major cities around the world. Typical schoolroom tests have, in the past, involved indicating the major cities on a specially provided map. Some students can perform well on the test because they can visualize the world and "see" where the cities belong—that is, by using a spatial strategy. Others have succeeded by memorizing the latitudes and longitudes of the cities in question and placing the cities accordingly—a more verbal strategy. But in neither case did students generally consider this task to be "fun." Like the old arithmetic drill and practice, the geography test was labeled work.

But now imagine a computer game in which children read about a child hero who is given information about latitudes and longitudes and must use this information to solve the problem of rescuing other children from evil aliens who are holding hostages at various points around the globe. The player roams around the computer world seeking these hostages and is rewarded for each one found. Instead of being tedious, unconnected facts, existing only to be memorized in school, city locations constitute vital information, and the ability to locate the cities becomes a necessary skill for achieving the interesting goal of rescuing hostages. Moreover, the goal is part of an intrinsic fantasy, involving a child hero with whom the child-learners can strongly identify.

As we noted in chapter 2, another ingredient of the intrinsically motivating instructional game is challenge, or uncertainty of outcome. With computer games, one way to achieve this challenge is to create games whose difficulty levels shift in accordance with how well the player is doing. Difficulty will depend on such factors as the amount of time needed to reach the goal, the memory capacity required, and the response speed that is necessary. With a computer's flexibility providing the foundation, these instructional games can vary in difficulty level so that they remain forever challenging.

Stanford psychologist Mark Lepper has pointed out, however, that simply inserting "challenge" into educational material is not without its pitfalls. Lepper remarks that the strategy of providing challenge can backfire if the children aren't aware of the structure of the learning material. Lepper writes, "Rather than enhancing motivation, such programs may often undermine it by leading children to perceive themselves as incapable of succeeding. As one child engaged in one of these programs commented, 'Every time I think I've got it, I just miss more of them.' " Educational material that is structured as a video game does much to remedy this difficulty. Instead of abstract problems that mysteriously increase in difficulty, some more-or-less true-to-life situation—such as being required to cut up a pizza into equal-size slices—is presented. It then

MIND AT PLAY

becomes quite clear from the situation that it's harder to divide
the pizza among seven people than among two people or eight
people.

Given that goals and challenges are appropriately provided,
why are they so captivating to people? Malone thinks it's be-
dause they engage a person's self-esteem. Success in the world
of learning, like success in any other sphere of life, can make
people feel better about themselves. The other side of the coin,
of course, as Lepper points out, is that failure lowers self-esteem
and, in so doing, can destroy a person's interest in learning.
The optimal instructional game is one that has a challenging
difficulty level but also minimizes the possibility of damage to
self-esteem. The challenge of a game must remain inviting
rather than discouraging. Computer games have the powerful
potential for providing this balance on an individual basis in a
way that the single teacher in a classroom full of students would
find practically impossible to do.

The final aspect of the computer game that makes it espe-
cially suited for drawing people into learning is fantasy. Child-
ren love fantasy and make no bones about it. Probably most
adults love fantasy, too, but they're often somewhat embar-
rassed to admit it. As a fantasy extreme, Malone points to
Disneyland. People love Disneyland, with all its surrealist,
characters and fantastic adventures. It would be natural to try
to identify ways of making the learning process more like going
to Disneyland. It is a challenge for educators to try to figure
out how to incorporate the fantasy elements so intimately
linked with the Disneyland-like experience into the world of
learning.

In computer games, fantasies abound. They are created by
all the “bells and whistles” of the game—the plot, the graphics,
and the sound effects. Fantasies are instructionally useful for
a number of reasons. For one thing, they help a learner apply

Learning from the Screen

old knowledge to understanding new things. For example,
when playing the game Darts, which teaches about fractions,
players can see right on the display screen that some objects
—arrows and balloons—are placed higher or lower than others.
If the connection is made between number size and position
on the number line, then learners will be able to use their old
knowledge about position to make inferences about the relative
sizes of unfamiliar fractions."

A second advantage of fantasies is that they provide or
provoke vivid images related to the material to be learned. This
provides an ideal situation, since numerous psychological ex-
periments have demonstrated that using imagery is one of the
best ways to learn new material. Finally, fantasies serve several
emotional functions, one of which is wish fulfillment. This
aspect of fantasies presents some unique problems for designers
of instructional games. It isn't an easy task to predict what
kinds of fantasies will be appealing to different people. Should
a single fantasy be selected that is likely to appeal to the entire
population? Should several fantasies be offered from which
students could select their favorite? Should some types of fan-
tasies be avoided if they could lead to special problems, for
example, fantasies that provoke feelings of aggression? In the-
ory, fantasies can be powerful ways of harnessing preexisting
emotional motivations and can be used to increase interest in
learning. Given these fantasies, it's easier for parents to buy
games for their children that both held the children's interest
and also taught them something.

Whereas Malone's work is general—it is primarily con-
cerned with an overall theory of instruction and motivation—
the work of John Frederiksen and his colleagues has been
aimed at a very specific educational problem, that of improving

8See, for example, A. Paivio, Imagery and Verbal Processes (New York: Holt,
1971).
reading skills. Frederiksen's past work, both theoretical and empirical, has been aimed at revealing some of the specific subskills that are involved in reading. Poor readers can be identified in terms of deficiencies in one or more of the subskills. Practice on the deficient subskills can then be prescribed.

As an example, some people are poor readers because they are slower (although not necessarily less accurate) than good readers at identifying specific letter groups such as cl, ple, th, and gen when these groups appear within words. Put another way, the good readers seem to be able to identify the letter groups more "automatically" than the poor readers. By "automatic" we refer to the ability to carry out some skill quickly and without paying conscious attention to it. An experienced driver can, for example, drive and hum a tune at the same time. Both of these skills—driving and humming—are automatic, and the operation of one does not interfere with the operation of the other.

Suppose that we wanted a reader, deficient in this letter-group recognition skill, to be able to practice it. To allow such practice to proceed in an enjoyable way, Frederiksen and his colleagues have devised a game called "Speed" that requires players to detect whether a target letter cluster is present within words that are presented in rapid succession. Is the cluster th present within hearth? Yes. Within church? No. It seems easy to do when you have plenty of time, but in the video game that the researchers developed, the players had to answer faster and faster without making too many errors.

Figure 5.1 shows the video screen as it appears at the start of a session. Notice that the letter cluster to be identified is ler. When the player is ready, words will be presented on the screen, one after the other, and the player's job is to hit one button indicating yes (ler is present) or another button indicating no (ler is not present) as quickly as possible after each word appears. A "speedometer" at the bottom of the screen indicates that, initially, the words are to be presented relatively slowly—about 80 words a minute, or just slightly more than one a second. But the player will be required to speed up, reaching a goal of 126 words a minute. Each time a correct response is made, the rate at which words are shown is increased, and the speedometer needle moves to indicate this increase. Figure 5.2 shows that the player is performing well, and words are now coming at a rate of 95 per minute. As the player gets faster and faster, he or she is rewarded with a smooth increase in speed, moving in the direction of the final goal.

Suppose the player makes an error. If so, an error light comes

---

Learning from the Screen

as representing miles per hour rather than words per minute. Although research on the benefits of these games is just now underway, preliminary reports are highly encouraging. Over the course of training, many players begin with goal speeds of 110 to 120 words per minute and reach speeds of over 150 words per minute by the third or fourth time they practice a particular cluster. After training, many players have shown a dramatic improvement in other skills related to reading ability.

LEARNING PHYSICAL LAWS

To some people, the laws of physics seem intuitively reasonable, but to others they don’t seem very obvious at all. Take the following hypothetical example: suppose a train were coming toward you at 60 miles per hour, and someone standing on top of the train threw a ball forward at 20 miles per hour relative to the train. How fast would the ball be going relative to you? If you answered 80 miles an hour, you would probably be among the intuitive group; if you answered 20 miles an hour, your intuitive grasp of physics might be faulty. People in the latter category, even when they learn to provide the correct answer, often seem unsure of themselves, as if the problem were beyond their real understanding.

Taking another example, look at figure 5.3 and try to answer this question: Suppose a bomber is flying north at 600 miles an hour and drops a bomb at exactly the moment it is over point X. Where will the bomb land? Again, the less intuitive person is likely to reply that the bomb would land on point X, when in fact the bomb would land to the north of the point. If you answered this question incorrectly, you are not alone. Many highly intelligent, highly educated people have the same difficulty, which actually has been awarded its own label: “naive physics.”

Naive physics is a topic of a good deal of research (much of
it sponsored by the U.S. Department of Defense, which wants
the recruits who operate its artillery and its nuclear generators
to have a good understanding of physics, not a naive one).
Psychologist Lepper points out that video games could poten-
tially be used to provide an intuitive understanding of many
principles of physics. He calls such games “educational simu-
lations” in reference to the fact that physical events can be
simulated on the screen by inserting the appropriate physical
laws into the computer program. He refers to an example
developed in a physics-learning program at the University of
California, Irvine, in which lunar and solar eclipses are simu-
lated on the computer. “In this program,” Lepper notes, “stu-
dents studying the mechanics of eclipses can ‘experiment’ with
a wide variety of relative positions of the sun, earth, and moon,
receiving responsive feedback after each choice.” Lepper goes
on to speculate about how other laws can be studied in such

Learning from the Screen

a way that the consequences of the laws are quite obvious.
Violations of the laws could similarly be examined. Suppose a
gravity simulation were changed so that gravity were governed
by a simple inverse-distance relationship, rather than by an
inverse-square relationship. What would happen? How would
the universe be different?

While playing Sabotage one day, one of the authors (GL)
was struck with his own vision of how video games could be
used as physics instructors. GL noticed how perfectly realistic
the little helicopters looked when they exploded and fell out
of the sky. He marveled at how elegantly this perfection could
be programmed, since the laws of physics could be written as
equations and the equations simply inserted into the game
program. GL went on to fantasize about how he would use a
video game to provide a child with an intuitive understanding
of physics. We will describe this fantasy in some detail, because
it’s an instructive and very plausible exercise in how intuitive
learning of physical laws might occur.

Imagine a child (call him David) faced with the problem in
figure 5.3. David, like most people naive about physics, declares
that the bomb will fall on point X. David (who is assumed to
have some programming capability) is then assigned the task
of programming the problem on his home computer, in order
to see what the falling bomb would look like. He soon com-
pletes this relatively simple job. He runs the program—and
what is depicted in figure 5.4 takes place on the computer’s
screen. The bomb moves horizontally during the time it is still
attached to the bomber. Then, when released, the bomb exe-
cutes an abrupt right-angle turn and drops straight down to
point X. The instant David sees this happen on the screen, he
knows that something is amiss. Things just don’t look like that
when they fall from a moving object. Obvious failure. Back to
the drawing board.
It doesn't take David long to figure out a major component of his problem. In a flash of insight, it dawns on him that the bomb was originally in motion along with the bomber and, just because the two had separated, there is no reason for the bomb to abruptly cease this forward motion—although David has implicitly assumed, both in the original answer to the question and in the program he had written, that that's indeed what would happen. He quickly revises his program so that the bomb's forward motion continues after it parts company with the bomber. When David reruns the program, the bomb's descent is as depicted in figure 5.5: it is a straight line going obliquely down from the point of release. This looks much better than the original version, but it still isn't quite right.

Determined to make it look perfect, David spends several hours working on variations of the program. First he tries varying the ratio of the bomb's forward to downward speed.

This doesn't provide any direct help. The bomb's descent still takes the form of straight lines; they just go at different angles, as shown in figure 5.6. It doesn't take long (especially after dropping a few pennies from his moving hand) for David to pick up the general principle that straight lines are wrong. The bomb should curve downward; that is, its angle of descent should become progressively steeper. Looking at the various angles, it's quite evident that the steeper lines result from faster downward speeds. Thus David figured out that if the downward speed were to increase during the bomb's descent, then the bomb's path would progressively steepen and would look more the way that he knows it ought to.

How exactly to make the speed increase? The simplest thing David can think of is to increase it by a constant amount after each constant unit of time. The way David has things set up, it takes about two seconds for the bomb to drop. So (somewhat arbitrarily) David arranges his program so that the bomb's
downward speed increases by ten miles an hour after every half second. The bomb’s path then looks as depicted in figure 5.7. Better, but still not perfect. Falling things change their direction of motion smoothly, not abruptly. So David tries increasing the speed every tenth of a second, and finally every sixtieth of a second, which is as fast as is permitted by the computer. At that point, the path looks like the one depicted in figure 5.8, which is at least in accord with David’s vision of how it ought to look. A proud David has discovered in a day what it took Isaac Newton many years to figure out.

In the process of solving the bomb problem, David has reinvented Newton’s first law of motion—that an object moving in a direction of motion will continue in that direction unless acted on by an outside force. Moreover, he has rediscovered the law that the force of gravity causes accelerated motion—that speed increases at a constant rate as a function of time. Perhaps David couldn’t have stated these laws in this formal way—not yet, anyway. But, since he has, in a very real sense, invented them, he had a superb intuitive feeling for them. When David would eventually be introduced to the rules in a formal physics class, he would have very little difficulty understanding them because he had already internalized the concept.

In a famous Platonic dialogue, Socrates demonstrates that a naive learner actually has, inherent within him, the knowledge of the Pythagorean theorem. In a similar way, the preceding example indicates that David had the knowledge of physical law inherent within him—it only took a video game to extract it. By seeing the way things were hypothesized to happen, he was able to tell immediately whether the hypothesis was correct, and he was provided with clues about how to correct erroneous hypotheses.
Learning from the Screen

All the sophisticated and complex principles of reinforcement can be easily programmed into this kind of package. For instance, to get the student going on spelling (or verb conjugation, or proving geometry theorems) the program can provide continuous reinforcement—a video game following each correct answer. Gradually, then, the ratio of correct answers to video games can be increased until the student is on an efficient, partial reinforcement schedule. Later we’ll discuss the possibilities for actually using this kind of technique in a video parlor to increase the amount of useful education that takes place there.

Direct Educational Benefits

Probably the question that we are most asked in connection with video games is: Do the kids learn anything useful by playing those crazy games in the arcades? As we’ve noted, the answer to this question is still open. There’s one notable area in which research is being done and in which definite progress has been made. This is the area of video game therapy for various types of mental disorders suffered, for example, by people who have had strokes or automobile accidents.

Before getting into this area, however, we must inquire about educational benefits of video games for relatively untroubled people. Are there any? We asked a number of psychologists around the country what they thought—at least intuitively—about whether people learn from the games, and the answers ranged from “Maybe they’re learning a tiny amount” to “Probably they’re not really learning anything.” The general consensus was that if a game was not really designed to teach
MIND AT PLAY

a specific skill, there was little likelihood that it would just happen to teach something useful.

What about motor coordination? If you skim through the video game magazines, you get the impression that we are, through the miracle of video games, becoming a nation of unsurpassed eye-hand experts—that our youth is acquiring priceless skills that, at a moment’s notice, could be transferred to make them superb marksmen or air traffic controllers or athletes. But is this really true? Again, the answer is currently up for grabs, since the research has not been done. The psychologists we talked with were skeptical. “Practice in eye-hand coordination?” said one of our colleagues incredulously when we posed the question to him. “Every human being gets practice in eye-hand coordination every time he picks up a pencil or a pen! What additional benefit could playing video games have except maybe to make people better at playing other video games?”

Our own intuitions are a little more optimistic. Since the widespread introduction of television, we’ve been entering an age in which the way we do things—indeed the way we think—has shifted from a linear, logical mode to a more spatial, visual, global mode. (Marshall McLuhan, of course, pioneered this insight.) One very salient manifestation of this shift is the recent explosion in the use of computer graphics. Such graphics are becoming increasingly sophisticated and are finding their way into a wide variety of disciplines. An architect designing a new building, for example, can program a computer to display various versions of the building in different perspectives, thereby seeing in minutes a series of drawings that would have taken weeks in the precomputer days. Or an accountant, presenting a company’s financial state, can do so by showing a series of computer-generated graphs and charts rather than reciting a long, dreary string of numbers.

Learning from the Screen

The list of such examples is virtually endless. It is natural to ask, then, whether all the practice that today’s video game players receive as they watch and manipulate shapes on screens is likely to help them in their later lives when they’re using computer graphics daily. It’s tempting to speculate that it might. On the other hand, any such help might better be classified as the indirect type that we considered earlier. Perhaps playing the video games will do no more than grease the path, so to speak, for learning the computer programming that will directly serve the future needs of today’s young video game players.

It’s noteworthy, in this respect, that the U.S. Army is currently considering video games as a means of training its recruits for the very similar task of learning to operate artillery. The army itself is remaining fairly quiet about this project, perhaps out of doubt that it will really work, or perhaps out of embarrassment over the image such a program might create in the minds of the public. But others aren’t remaining quiet at all. Art Hoppe, a columnist for the San Francisco Chronicle, for example, has this to say:10

WORLD WAR VII

It was June 28, 1994. Aboard the U.S. Space Station Dreadnought, silently orbiting 248 miles over Washington, D.C., the grizzled colonel addressed the dozen young men who comprised the elite Special Zapping Force.

“Man your consoles, men,” ordered the colonel grimly. “Intelligence says the Russians are launching World War VII.”

It had all begun 12 years earlier with the fourth orbital flight of the Space Shuttle Columbia. For the first time, the shuttle’s payload and purpose were more military than scientific.

“After all,” said the general in charge of the Pentagon’s space

program, “space is the modern equivalent of the ‘high ground’ that military leaders have for centuries sought out and exploited to their advantage.”

So it was that the struggle commenced between American and Soviet forces to seize and hold the “high ground” of space.

Initially, the two sides were relatively equal and the battles nip and tuck. Each attempted to flood outer space with spy satellites, hunter-killer satellites which sought out and destroyed the spy satellites, and “shoot-back” satellites which could sense and explode approaching hunter-killer satellites.

World Wars III and IV ended in ties with space cluttered by lumps of metal and tangles of wire. But the military on both sides were happy planning and fighting their wars hundreds of miles above the surface of the planet. And the public was equally happy to keep them there.

Then, in 1989, the Dreadnought was launched and manned by the Special Zapper Force, each highly skilled young member equipped with a viewing console and laser beam. The Russians countered with a similarly armed Salyut 8 space station.

But World War V was strictly no contest. When it came to shooting down enemy satellites, the Russians were simply no match for the Americans, most of whom had devoted up to ten hours a day since the age of seven training for just such a battle.

Typical of their ranks was 21-year-old Sergeant Dick Deadeye, who had graduated from the prestigious Institute of Astrophysics and Video Arcade on a Pentagon scholarship of $100,000 or, more accurately, 400,000 quarters.

While young Americans had been preparing for war, Russian youth had been wasting their time on chess, calculus and pushups. After their disastrous defeat in World War V, the Kremlin launched a crash catch-up program in every Russian community featuring such devices as “Tractor Command” and “Won Ton Invaders.”

But it was too late. In World War VI, they were beaten again with Sergeant Deadeye racking up a record 1376 enemy satellites in 27 minutes. Yet here they came again.

“Fire!” cried the colonel and the screens lit up with exploding satellites and missiles. Suddenly, they all went blank. Then on each appeared the same message:

Learning from the Screen

“This is the Intergalactic Council. Your species has been found guilty of littering, loitering and disturbing the peace. We have dispatched our Political Action Committee Man to deal with you.”

“It’s a trick!” shouted the colonel.

“No, sir,” said Deadeye. “Look out the port!”

Sure enough, from a corner of the universe emerged a gigantic, yellow, pie-shaped object with a single eye and a wedge-shaped mouth. Shifting jerkily this way and that, it one-by-one gobbled up all the satellites, Salyut 8 and approached the Dreadnought head on.

“Fire, Deadeye!” ordered the colonel. “Only you can save the human race from extinction.”

“It’s no good, sir,” said Deadeye. He then uttered the tragic words that had already brought gloom to the hearts of all red-blooded Americans for a generation: “I’m out of quarters.”

Something not so unlike Hoppe’s comic fantasy may soon be a reality. The concept of war in space is becoming more real with every passing day. Even on the ground, artillery is now operated using laser aimers whose data are depicted on a computer screen. At the very least, it would be to the army’s advantage to select as recruits young men and women who are proficient on the kinds of skills that are useful in such a high-tech environment. Indeed, actual video games are now being considered for just such a selection process. One example reports work done on behalf of the U.S. Navy Biodynamics Laboratory. After extensive testing with the games of Air Combat Maneuvering, Breakout, Surround, Race Car, and Slalom, the researchers conclude that “In terms of availability, equipment, reliability, expense, and other practical considerations, the video games have many advantages . . . video games have considerable promise for performance testing and other applied contexts.” In other words, those people who were good

on these games would be considered suitable for various, quite similar, military activities.

So far, we have been rather tentative about the direct educational benefits of video games. One area in which their use for learning has been quite real, however, is as training aids for certain cognitive and perceptual-motor disorders. These disorders can be found in patients who have been in accidents, have had strokes, or have simply been born that way. Such persons, it turns out, can often be trained so they can learn to lead more normal mental lives. Video games have recently been discovered to be superb tools for use in such training effort.

Let's first consider a simple example. Some people have disorders involving the muscles of their eyes. One of the authors (EL) was one such sufferer. The therapy required for this disorder was very simple: she had to work with a contraption that presented two dots to the eyes, and she was supposed to constantly stare at them. The dots could be moved farther and farther away to create the proper exercise for the eye muscles. At other times she had to simply look back and forth between two dots for hours on end. Although the therapy was effective in theory, it was so boring that she rarely could bring herself to actually do it. Many types of video games, however, require exactly the same kinds of eye movements but are orders of magnitude more interesting. Some modern sufferers are being treated in this far more interesting way; according to several, the process is far more enjoyable.

Video games are already in use for ailments that are substantially more serious than an inability to move the eyes together correctly. A good example is Jan, another of our composite respondents. Jan is a typical teenager who couldn't wait to get her driver's license. On the morning of her sixteenth birthday she took her driver's test, passing it by a mere two points. On weekends her father let her drive his car. Then, one Saturday, less than two months after her birthday, she ran the car into a redwood tree. Although Jan survived the accident, she did suffer some brain damage. Her major problem seemed to be a rather unexpected inability to spell, a malady known to specialists as spelling dyspraxia. Jan, depressed after her accident, was unable to complete the spelling drills that had been ordered as treatment. Fortunately, her doctor had another idea. At a nearby hospital, Jan was put on a new kind of therapy—video game therapy. Twice a week she went to the clinic for this novel treatment, mostly playing "Hangman," a game that is particularly enjoyable in its video format. In "Hangman," one or two people guess letters in a word. If a correct letter appears, there is no penalty; if an incorrect letter appears, another portion of the "hangman" is added on. After only two months of playing Hangman regularly, Jan's problem was substantially relieved. By Christmas it was no longer noticeable. Again we can contrast the motivational aspects of video game therapy with the more traditional methods such as old-fashioned spelling drills.

Even relatively simple games such as Pong and Breakout demand careful visual searching and tracking. Games like Concentration are useful for memory problems in general. Concentration is a measure of visual memory in which one or two players attempt to match objects revealed by choosing pairs of squares. In the game the memory targets are not specific to English-speaking persons; thus it can be used with non-English-speaking populations. Games like Adventure, in which a player must find a "hidden" object while avoiding dangerous pitfalls like dragons and enemies, are useful for rehabilitating
people who have problems with strategy and planning. And the sports games, like Basketball, Bowling, and Football, in which the premium is on quickness, accuracy, strategy, and alertness, are useful for people who have problems with eye-hand coordination, visual field, and tracking.

A number of therapists have opened centers throughout the U.S. that are dedicated to video game therapy. Therapists informally report that patients who are not motivated on many other types of retraining tasks are finding the games highly challenging and beneficial. However, there is not yet a great deal of published research in the area. One intriguing study involved twenty-five learning-disabled children between the ages of six and thirteen. Subjects were tested both before and after heavy playing of a number of video games. The game sessions lasted approximately thirty minutes and were carried out at weekly intervals for twelve weeks. Dramatic improvements occurred. Specifically, children improved in their ability to perform a line-tracing task that tapped into motor ability and in a task of spatial visualization in which they had to match a missing block with a like-shaped contour. Further research is now underway to answer such questions as whether it is best for the children to engage in "structured" or in "free" video game training.

One interesting ongoing rehabilitation project is that of William Lynch, Director of the Brain Injury Rehabilitation Unit, at the Palo Alto Veterans Administration Center. Lynch has begun collecting performance records on normal patients playing video games in order to develop standard score profiles, much like those used with personality or neuropsychological data. The performance records of brain-impaired patients can then be compared to these standards. With these

\[\text{Learning from the Screen}\]

profiles, it becomes very easy for the therapist to provide the patient with ongoing feedback through therapy. In addition to the feedback, Lynch is finding an important benefit of these games: patients don’t have to come to the clinic for therapy; rather, the therapist (that is, the game) goes to the patient’s home. Lynch hopes to see not only new video game programs and more varieties of games, but especially more sophisticated home video game units, which will permit user modification or development of programs.

Lynch has found that many potentially useful games are too difficult to comprehend or physically difficult to manipulate for certain patients at the early stages of recovery. He sees a need for more choices in game difficulty or format to accommodate people with cognitive disabilities. The use of adaptive controllers, for example, could make video games available to quadriplegics or multiple amputees. The quality of future video displays will be considerably improved, and soon we will see training programs that can evaluate skills requiring three-dimensional perception in a more lifelike manner. With these advanced techniques, an entire simulated environment such as a room or a building can be brought into the clinic or the home. Lynch believes that efficient and economical treatment dictates a change in approach from the traditional one on one to a combination of individual, group, and automated techniques.

\[\text{What Will the Future Bring?}\]

There are two obvious future directions for video games as learning devices. The first is to endow games specifically designed for learning with the ultra-motivating character of the
MIND AT PLAY

arcade games. As we have seen, this is already occurring to some degree, as research projects move briskly along on several fronts. On a theoretical front, work like that of Malone is aimed at identifying the characteristics of video games that make them motivating to begin with. And on an empirical front, work like that of Fredriksen is aimed at actually creating video game–like computer programs that will help promote specific cognitive skills.

The second direction is to somehow make arcade games more specifically educational. One step would be to have arcade games keep track of the performance of individual players and use this information to start players off at appropriate difficulty levels. From the players’ point of view, this would save an expert the boredom of moving through the elementary levels before reaching the level that presents him or her with a reasonable degree of challenge. In addition, points could be presented both cumulatively and as an average number per game. From the manufacturer’s point of view, games that don’t present any challenge are bad because they eat up the game’s playing time—it would be much more profitable to start a good player off at a more difficult level, thereby decreasing the total amount of time played for that quarter. And from the point of view of educators and parents, keeping an individual’s history is a very helpful step in the process of making the games educational.

A potentially more difficult problem will be to persuade game makers to insert educationally beneficial elements into the popular arcade games. Perhaps the same civic groups that are trying to have video parlors outlawed could instead try to arrange community subsidies for arcade operators or game manufacturers, the amount being determined by the degree to which designated educational games are played. Or perhaps tax credits could be granted for companies developing demonstra-

Learning from the Screen

bly educational games. But what if educational games simply cannot be made as inherently compelling as the more traditional video games? In that case, good performance on an educational game could be rewarded by a free turn on any game in the arcade. Community funding of such a scheme would probably be minuscule compared to the education budget as a whole.

The Research Window

We have painted a rather rosy picture of computer games as potentially powerful educational devices. Not everyone is so optimistic. Mark Lepper, who seems to be generally in favor of the games, is nonetheless careful to enumerate some of the differing opinions about a variety of computer-related issues. He writes:

To proponents [of the games], the addition of motivational features [such as fantasy, plots, graphics, and sound effects] is expected to enhance attention and produce superior learning. To critics, the addition of these extraneous features seems more likely to prove distracting and impair initial learning. At the very least, from this second perspective, the use of such features should make learning significantly less efficient per unit of time invested.15

Lepper goes on to list similarly opposing opinions in other educational domains such as motivation and long-term retention.

Lepper raises other sobering issues as well about possible

social consequences of the games. We've noted that boys vastly outnumber girls in the video parlors and that computer games may well provide an easy lead-in to computer literacy. Like the Pittsburgh researchers cited in chapter 4, Lepper is concerned about the possibility of girls growing up to be second-class citizens where computer usage is concerned. And, he asks, what about class and socioeconomic differences? Will they be exacerbated due to the rich having greater access to computers than the poor? Or, conversely, will the video parlor prove to be a social equalizer in this domain?

These are serious questions, and, as Lepper points out, we don't have much time to answer them—we have a "research window" of perhaps ten years. To do the required research properly, it's important to be able to compare children who have had computer experience with other children who have not. But in the near future, it will be impossible to find research subjects who are computer-naive. That happened with TV. There was about a five-year period in the early 1950s when it would have been possible to match groups of TV-watching children and non-TV-watching children. Comparison of such groups would have allowed powerful conclusions about effects of TV on a variety of social and educational behaviors. But this opportunity slipped away. Before the social scientists realized what had happened, the vast majority of homes in the U.S. had a television set, and the opportunity to do the research properly had vanished forever. We hope, along with Lepper, that this missed opportunity will not repeat itself with computers because computers will probably have effects on our society more profound than those of television.

We hope that some of the readers of this book will follow the path from video games to computer technology, and it seems appropriate to end by sketching out for computer novices how computers operate, what the major trends are in the technology, and how computers broadly relate to the games.

The Computer as a System

To be a computer, a device must have two characteristics. First, it must be capable of solving problems by manipulating information according to systematic rules. You mimic the principal operations of a computer when, for example, you multiply two three-digit numbers to find their product. The process is
a sequence of logical, systematic steps such as elementary multiplication, carrying, and addition. Second, the device must be capable (at least in theory) of solving any problem that has a logical solution.

In chapter 3 we discussed a system as an integrated collection of components, all working in concert toward the accomplishment of some goal. Like a stereo system or a cognitive system, a modern computer is a system with at least three components: memory, a central processing unit (or CPU), and input/output (or I/O).

MEMORY

To store information, a computer has a memory made up of a long string of what are called words. Each word is actually a number. How many words are in a computer’s memory depends on the computer, but memory typically comes in chunks of 1,024 words. When applied to memory size, this number—1,024—is, in computerese, called a thousand, or “one K.” Thus a computer with 64K of memory would actually contain $64 \times 1,024$ or 65,536 words.

How does the computer keep track of what’s where in its memory? Each location in memory has an address. The address of the first word is 0, and, in the case of a 64K computer, the address of the last word is 65,535 (that is, the 65,536 words are numbered 0 through 65,535.) The computer is able to access any of the words by its address. It can detect (or “read”) the word at any given address and it can, in some cases, change (or “write”) the word at a given address as well.

Computer memory is characterized as either “read-only memory” (ROM) or “random-access memory” (RAM). Read-only means just what it sounds like—the contents of memory are determined by the computer’s makers when the computer is manufactured. The computer can read ROM—that is, it can determine what a word at some address in ROM is—but a ROM word cannot be changed.

Words that are in random-access memory, or RAM, can, in contrast, be both read and changed by the computer. Thus the essential difference between ROM and RAM is that the computer can change the contents of the latter but not of the former. Later we’ll elaborate on the circumstances in which one type of memory versus the other tends to be used.

The numbers that constitute the words in a computer’s memory can be used to represent information. This information can be of any sort, but two of the most common uses of words in memory are the representation of letters or other alphanumeric (keyboard) characters and the representation of video screen locations.

The numbers in memory represent letters when the computer is used as a “word processor” or computerized typewriter. For example, we are writing this book using a word processor. If one of us types the word “Information” on the computer’s keyboard, each letter in the word is represented by a number, and as they are typed, these numbers are written into successive locations (addresses) in the computer’s memory. Which numbers are used to represent which letters is arbitrary, but this correspondence was decided upon by an international committee some years ago, and it is called ASCII code. Table 6.1 shows the numbers that represent the letters in the word “Information.” As one of us sits typing this manuscript, a long sequence of numbers goes into the computer’s RAM, each number representing a letter (or some other keyboard character), and the sequence of numbers in the computer’s memory thereby represents the manuscript we’re writing. Eventually the numbers in all the memory locations that contain the manuscript will be sent, in sequence, to a printer. The printer is wired up so that when it is sent the number 73, it will print
MIND AT PLAY

out "I," when it is sent the number 110, it will print out "n," and so on.

A second common type of information that the numbers in a computer's memory are used to represent are locations on a video screen. A video screen (of which a TV screen is a common example) is a tube, behind which (inside the TV) is a gun that shoots electrons. By aiming the gun at a particular place on the screen, that part of the screen can be displayed, or made to light up. In computer lingo, the location is "painted" on the screen. When used in conjunction with a computer, a typical video screen can be conceptualized as a grid, say 100 locations horizontal by 50 locations vertical. Each point in the grid (there are $100 \times 50$ or 5,000 of them in this example) is called a pixel, and the location of each pixel is specified by a horizontal and a vertical coordinate. For example $(50,25)$ represents the pixel occupying the junction of the fiftieth column and the twenty-fifth row. (This would be the center of the screen, in our example.) An entire picture is painted on the screen by specifying a series of such coordinates and then aiming the electron gun so as to paint the corresponding pixels on the screen. Figure 6.1 shows how a flying saucer might be displayed. If you examined the figure carefully, you would find that 49 of the 5,000 pixels are painted—those at coordinates $(51,24), (52,24)$, and so on.

The use of information in memory to represent locations on a video screen is, of course, critical in the construction of video games. It's the way Pac-Man, spaceships, and anything else appearing on the screen gets made. Soon we will expand this example to see in a little more detail how the computer might be programmed to display such a flying saucer and how the saucer might be programmed to do interesting things. Before we do this, however, a few words on computer programs in general.

**TABLE 6.1**

<table>
<thead>
<tr>
<th>Letter</th>
<th>ASCII code</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>73</td>
</tr>
<tr>
<td>n</td>
<td>110</td>
</tr>
<tr>
<td>f</td>
<td>102</td>
</tr>
<tr>
<td>o</td>
<td>111</td>
</tr>
<tr>
<td>r</td>
<td>114</td>
</tr>
<tr>
<td>m</td>
<td>109</td>
</tr>
<tr>
<td>a</td>
<td>97</td>
</tr>
<tr>
<td>t</td>
<td>116</td>
</tr>
<tr>
<td>i</td>
<td>105</td>
</tr>
<tr>
<td>o</td>
<td>111</td>
</tr>
<tr>
<td>n</td>
<td>110</td>
</tr>
</tbody>
</table>

**FIGURE 6.1**

Close encounters of the computer kind: how a flying saucer might be displayed.
Earlier we pointed out that a necessary characteristic of a computer is flexibility—a computer must, at least in principle, be able to solve any solvable problem, and it must be able to manipulate information in any logically possible way. This flexibility is accomplished by a computer program, which is a list of instructions telling the computer what to do.

Computer programs reside in the computer's memory: just as the words in memory can be used to represent letters of the alphabet or locations on a screen, they can also be used to represent instructions in a computer program. For example, suppose you wanted to write some video game–related program. Starting simply, the first program you might want is one that would display the flying saucer depicted in figure 6.1. Figure 6.2, a schematic map of your computer's memory, gives the general idea of how the program works. The words at memory addresses 101 to 198 correspond to the coordinates of the pixels that are to be painted on the screen. The odd addresses (101, 103 ... 197) contain the X (horizontal) coordinates, whereas the even addresses (102, 104 ... 198) contain the corresponding Y (vertical) coordinates. Thus the words at a pair of addresses such as 101 and 102 contain the horizontal and vertical coordinates corresponding to one screen location that is to be displayed. For reasons that will become clear in a moment, location 199 in memory contains the number 255, which couldn't possibly represent an X or a Y coordinate (since X and Y coordinates have maxima of 100 and 50, respectively, in the hypothetical computer you are using).

Thus the information corresponding to the to-be-displayed screen locations are in addresses 101 to 198 of the computer's memory. How does the computer know what to do with them? Starting at address 1000 in memory, the words in memory represent instructions that tell the computer what to do with the screen locations. Successive program instructions are put into successive memory locations. As it turns out, there is not one instruction per word—rather, each instruction uses more than one word. Exactly how words are assigned to instructions is beyond the scope of this book. All you need realize is that, starting at address 1000, the contents of memory represent instructions in a computer program.

This program has seven instructions. Instruction 1 says, "set some variable, V, equal to 101." Instruction 2 says, "Set a

<table>
<thead>
<tr>
<th>Address</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>51</td>
</tr>
<tr>
<td>102</td>
<td>24</td>
</tr>
<tr>
<td>103</td>
<td>52</td>
</tr>
<tr>
<td>104</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>197</td>
<td>53</td>
</tr>
<tr>
<td>198</td>
<td>28</td>
</tr>
<tr>
<td>199</td>
<td>255</td>
</tr>
<tr>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

Program

1. Set V equal to 101.
2. Set X equal to word at address V.
3. Set Y equal to word at address V + 1.
4. Light up pixel at (X,Y).
5. Increment V by 2.
6. Is the word at V equal to 255?
   Yes: Go to Instruction 7.
   No: Go to Instruction 2.
7. Stop.

FIGURE 6.2
The flying saucer program.
second variable, \( X \), equal to the word at address \( V \).” Instruction 3 says, “Set a third variable, \( Y \), equal to the word at address \( V + 1 \).” Instruction 4 says, “Display the screen location corresponding to \( X \) and \( Y \).” Instruction 5 says, “Increment \( V \) by 2.” Instruction 6 says, “Is the value of the word at address \( V \) equal to 255? If so, go on to Instruction 7; if not, go back to Instruction 2.” Finally, Instruction 7 says “Stop.” You can now see the significance of the 255 at address 199. When the program finds this number, it will “know” that it has displayed everything to be displayed and that it is therefore finished.

This little program has demonstrated two major features of all computer programs. The first is a loop. The program loops between Instruction 6 and Instruction 2. Each time it goes through the loop, it reads another pair of \( X \) and \( Y \) screen coordinates from memory and paints the appropriate screen location. The second feature is conditional branching. This feature is embodied in Instruction 6, which says, “If one thing is true, then do something, whereas if another thing is true, do something else.”

Once this program is written, and displaying the saucer, it would be relatively easy to extend the program so that the saucer appears to move. The extended program shown in figure 6.3, for example, would cause the saucer to appear to move to the right. Three instructions have been added in this program. Instructions 7 to 10 have the effect of incrementing all the \( X \) coordinates by 1. The program now loops from Instruction 10 back to Instruction 1. Every time it passes through the loop, all the \( X \) coordinates of the saucer are incremented by 1, so the saucer will be successively displayed one pixel to the right relative to where it had been displayed previously. The visual system will perceive a continuously moving saucer from these successive displays, just as it perceives continuous motion from the rapidly presented still pictures in a movie.

The program in figure 6.3 is still very primitive. Notice, for example, that the saucer will eventually disappear off the right-hand side of the screen. To see it again, you would have to restart the program. But most novice programmers would find it highly reinforcing to see even this much action occur as a result of a program they have written. If you understand the two simple programs we’ve provided so far, you can probably see ways to extend and improve them. For example, you might insert an instruction sequence that would detect whether the saucer was moving off the screen, and, if it were, to act accordingly—to move the saucer back to the left or, perhaps, as is so popular in video games, have it reappear somewhere else on the screen. By manipulating the vertical coordinates as well as the horizontal ones, you could have the saucer moving wherever you wanted on the screen. Extending the program still further, you might insert instructions to detect when something is

---

**FIGURE 6.3**

The flying saucer program extended.

1. Set \( V \) equal to 101.
2. Set \( X \) equal to word at address \( V \).
3. Set \( Y \) equal to word at address \( V + 1 \).
4. Light up pixel at \( (X,Y) \).
5. Increment \( V \) by 2.
6. Is word at \( V \) equal to 255?
   Yes: Go to Instruction 7.
   No: Go to Instruction 2.
7. Set \( A \) equal to 101.
8. Increment word at address \( A \) by 1.
9. Increment \( A \) by 2.
10. Is word at \( A \) equal to 255?
    Yes: Go to Instruction 1.
    No: Go to Instruction 8.
MIND AT PLAY

typed in from the keyboard and have the saucer’s behavior be contingent on what was typed in. Even with the simple sorts of instructions we’ve used, the possibilities are limitless.

In these examples, we’ve acted as if the program itself were humanlike in its ability to follow a set of instructions. The component of the computer that is actually responsible for carrying out the instructions is the central processing unit, or CPU. The CPU is the heart of the computer, and it has three major functions. The first is dealing with memory locations, which consists of both reading and writing (that is, changing the value of a word) at some memory address. The second function is communicating with the outside world. The third function is keeping track of the order in which program instructions should be executed. Generally, the CPU does this by assuming that it should execute instructions in the order that they appear in memory unless some instruction indicates otherwise. This kind of exception occurs, for example, in Instruction 6 in the program of figure 6.2. This instruction tells the CPU that, unless a particular thing is true (the value of V is 255), the next instruction to be executed is not Instruction 7 but, rather, Instruction 2.

INPUT/OUTPUT

So far we’ve focused on events that take place inside the computer. If a computer is to be useful, however, it must be able to communicate with the outside world. Such communication is called input/output, or I/O for short. Communication from the world to the computer is input; communication from the computer to the world is output. So, for example, displaying the Pac-Man board on a screen would be an output event—sending the locations stored somewhere in memory out to the screen. Detecting that the Pac-Man joystick has been moved would be an input event, in the sense that information from the outside world (the stick movement) would enter into the computer’s CPU (which, in accordance with the Pac-Man program, would presumably then act in an appropriate way, that is, send output to the screen corresponding to the direction in which Pac-Man ought to move).

In the beginning days of computers—the late 1940s and early 50s—communication between the computer and the outside world was astonishingly primitive. Computers were used mainly as gigantic calculating machines, and computer programs were written primarily to solve tedious equations. At that time the only purpose of person-computer communication was to get programs into the computer and to read numbers that were the result of the computer’s computations. These early computers had a row of lights on the console with a toggle switch under each light. Entering information into the computers was accomplished by setting the toggle switches in the appropriate configuration, and reading information from the computer was accomplished by reading the configuration of the lights.2

And so, by flipping toggle switches and reading lights, the computer pioneers would tediously enter data into the computer’s memory, enter computer programs, and read the results of computations. It didn’t take them long to tire of all this busy work, and soon the teletype, a remnant of the telegraph era, began to be used to enter data. The teletype looked like a typewriter with a long roll of paper feeding into it. The operator would type information into the teletype and, in addition to being printed out on the paper, this information would be

---

2The numbers within a computer’s memory are actually represented in binary. In binary, each “digit” is either a one or a zero; thus a one was indicated by “switch up” or “light on” and a zero was indicated by “switch down” or “light off.”
related electronically to the computer's CPU. Similarly, output from the computer would be transmitted from the CPU to the teletype, where it would be printed out.

The teletype was the major input/output device until about ten years ago. Even then it was clear that printing things out was the slow link in the system. Printing out took a long time, and it was unnervingly permanent. If you made an error, you couldn't erase it; you had to start again or somehow mark the error as such.

Today the almost universal form of basic human-computer communication is the video display/keyboard. The keyboard, like the teletype keyboard of yesteryear, is still used to enter information into the computer; however, the information is displayed not on a slow, permanent typewriter but on a speedy, easily erasable, and easily modifiable video screen. Whereas characters can be printed on a screen at the rate of hundreds, or even thousands, per second, even a very fast typist can type only about six or seven characters a second. Thus the slow link in the communications system has switched from output to input. The most likely new development in this area is that computers will soon be able to understand spoken speech, thereby improving things on the input side.

**Mass Storage**

We've noted that the primary virtue of a computer is its flexibility and that this flexibility is accomplished by the use of computer programs. More precisely, it is because any program can be run on any computer that this flexibility is available. In order to take advantage of the flexibility, we have to be able to run more than one program on any given computer. However, computer programs reside in computer memory, and computer memory, as we've seen, is limited. There's just so much information that can be stuffed into memory before you run out of room. This means that when we've finished with one computer program and want to use another, we'd like to have some place where we can easily store the program that we're done with. This is accomplished with mass-storage devices.

Mass storage was first done using IBM cards and paper tape. Most people are familiar with IBM cards. Essentially, they store information by having holes punched in certain places; the particular configuration of hole locations in the card represents the information. Paper tape worked similarly: holes were punched on a long strip of paper that looked like a roll of masking tape. When a person was finished using a computer program, the output from the program and, if necessary, the program itself were transferred via a computer-controlled punch onto tape or cards. When the information was to be fed back into the computer again, it was done via a card or tape reader.

These mass-storage techniques worked reasonably well, but they had two disadvantages. First, both punching the holes and reading the punched holes were slow processes. Second, piles of cards or long rolls of tape were inconvenient—they were big and bulky, and could easily get torn or, in the case of cards, dropped. This latter event was the nightmare of anyone who worked with computers, since it could take hours to reassemble a large deck of cards that had been accidentally dropped and scattered.

To rectify these problems, two other mass-storage devices were developed. The first is magnetic tape. Magnetic tape is similar to video or audio recording tape, consisting of a long roll on which information can be stored in the form of magnetic impulses. Magnetic tape was convenient to use, but reading it and writing onto it was still a slow process. To get to wherever you want to be on the tape, you have to fast-forward or rewind it, which typically takes a long time.
The other device is the magnetic disk. A disk looks like a phonograph record, but its surface is covered with the same kind of magnetic material used on magnetic tape. Thus the physical storage of the information is the same on tape and on disk. The difference is that the disk is read or written on by the computer via a disk drive, which spins the disk very fast. The magnetic heads responsible for reading and writing can thus access any part of the disk—and any information on the disk—very quickly.

Computer Trends: Small and Fast

So much for an overview of what computers are designed to do and how they do it. We’d like now to take you on a more detailed tour of computer history. There are two major trends: computers are getting smaller, and they’re doing things faster. These trends continue today; understanding them is very important if we are to be able to speculate about what the future has in store for video game technology.

CONCEPTUAL BIRTH

The idea of a computer came into being a surprisingly long time ago, courtesy of an eccentric but brilliant British mathematician named Charles Babbage. Babbage, who lived in the early 1800s, developed a fanatical dream to build a machine capable of solving polynomial equations. His principal motivation for this seemingly arcane cause was the same as every schoolboy’s—he hated the tedium of performing this routine, boring task by hand.

Babbage managed to persuade Parliament to grant him £1,500—a princely sum—to develop what he termed a differ-
You can imagine Babbage's intense frustration. He felt, quite correctly, that if he could just get the machine to work, it would vastly diminish the burden of menial mental labor, just as the myriad inventions of the Industrial Revolution were vastly diminishing the burden of menial physical labor.

Babbage was a great visionary, and he must have died a disappointed man. But his ingenious design lay hidden like a land mine, waiting for the right combination of politics and technology to set it off a century later. It's a shame, really, that Babbage can't be brought back today to see the fruits of his ideas. Farsighted though he was, he would probably still be amazed to see his beloved machine shrunken to the size of a typewriter and responsible for everything from men landing on the moon to little, pielike humanoids zipping around mazes on mysterious screens in places where you drink your ale and buy your food.

THE ELECTRONICS EXPLOSION

Babbage's computer wouldn't work because it was fundamentally mechanical. Information was represented by moving gears, and the "CPU" was a similarly complex configuration of gears and cams. Whenever metal parts have to move in complicated patterns, extreme precision is needed. The whole thing was just too cumbersome.

By the middle of the twentieth century, more than a hundred years after Babbage's efforts, the electronic age was in full swing. Electricity provided the potential for incorporating Babbage's ideas into a workable device, since by using electrical components to represent information, the need for moving, mechanical parts could be vastly reduced. Additionally, manipulation of information could proceed very quickly when the manipulation consisted of on/off switching of electrical components rather than the movement of gears. In short, comput-

ers and electricity formed a potentially compatible partnership.

The arrival of World War II provided the final impetus for the creation of electronic computers. This war, like all others, spawned a variety of needs for sophisticated information-processing devices. One, for example, was the need to crack enemy communication codes. Additionally, new weapons were rapidly appearing on the scene, and complex computations were required to determine their performance characteristics. These needs led to a crash program in computer development, and in the 1940s, the first two computers—one, called the Mark I, built at Harvard, and the other, called ENIAC, built at the University of Pennsylvania—were turned on and actually worked.

These first working computers were huge, grotesque monsters, filling entire rooms. They were so big because primary informational components were vacuum tubes. Each tube is not that large—a few inches high and about an inch in diameter. But when you have thousands, each constituting only one tiny part of a computer's memory, the whole conglomeration takes up a lot of space.

These early, room-sized computers were useful for big, well-financed organizations such as the military, the census bureau, and a few large companies. But their use was limited—you could hardly put a bunch of them in a video parlor, for example. In the late 1950s, a major technological revolution occurred, one that rivaled in importance the invention of electricity itself. This was the invention of the transistor by a team of scientists at the Bell Telephone Laboratories. A transistor performs much the same duty as a vacuum tube, but it has three major advantages relative to a tube. First, it uses substantially less power. Second, it can be made arbitrarily small. And third, it operates a lot faster. The new generation of computers that used transistors in place of tubes arrived in the 1960s. They had
shrank from the size of a room to the size of a piano, and they operated orders of magnitude faster.

A third revolution, which occurred in the 1970s, brings us up to the present. To get a flavor for this new development, it’s important to realize that the first transistors, convenient though they were, still had to be manufactured individually and wired together in fantastically complex ways. One of us worked for a computer company in the mid-1960s and has vivid memories of platoons of middle-age women, sitting all day patiently threading wires together, looking as if they were all knitting a gigantic and incredibly intricate macrame plant holder.

This manufacturing bottleneck created problems as far as the finished product was concerned. First, another size limitation was soon reached, since the electrical wire that was used to string the transistors together couldn’t get smaller than a certain minimum size, and, moreover, the individual components had to be large enough for the workers to manipulate by hand. Second, the workers themselves were expensive to employ, which meant that the resulting computers were similarly expensive. Third, since human workers—particularly those doing long, repetitious, boring jobs—make mistakes, the quality of the computers suffered.

All these problems ended with the development of the integrated circuit. An integrated circuit is a large number of transistors, miniaturized to a fantastic degree, all prewired and created in an instant on a single piece of silicon. The beauty of these integrated circuits, or “chips,” is that once a master chip is designed and made to work, the creation of an indefinite number of copies is so cheap and easy as to be trivial—much as the making of an indefinite number of lithographs is trivial once the original plate has been made. Moreover, these chips can, in theory anyway, be made almost arbitrarily small. An entire computer CPU can now be placed on a silicon square the size of a fingernail, and they’re getting smaller all the time. Indeed, if you’ve ever peeked inside a computer—or even a pocket calculator—you were probably startled to discover that it consisted mostly of air. That’s because the actual electronic components are very small compared to the components such as keyboards, switches, and so on that are needed so that humans can operate the devices with their big, clumsy fingers.4

Special-purpose Computers

Thus far we have described the workings of what are referred to as general-purpose computers. The major characteristic of a general-purpose computer is flexibility, in the sense that any program can run on it. This means that all or most of the computer’s memory is RAM, which, you’ll recall, has the capability of being changed, or written into, as well as the capability of being read.

Increasingly, however, special-purpose computers are beginning to enter the world. As the name implies, a special-purpose computer is designed to do only one job. For example, you may have seen automobile advertisements featuring a “computerized fuel-injection system.” Such a system involves a small computer that senses the state of a car’s engine at each instant

4We are currently seeing a “multiple-use wristwatch” craze. Computer components are sufficiently small that tiny computers can be put into watches, which can then act as calculators, play popular tunes, and even display video games in addition to a myriad of timekeeping activities. But there is a serious limitation on these gadgets, which is the number of buttons, pressable by human fingers, that can be attached to them.
in time, computes the most efficient fuel/air ratio, given the temperature, air pressure, engine power, and so on at that instant, and then adjusts the fuel flow such that the appropriate ratio is obtained. As we shall see, arcade video games are special-purpose computers.

Since a special-purpose computer does only one job, only one program is required for it. This means that the program is written by the computer designers and put into read-only memory, or ROM, which, you’ll recall, can’t be modified. What’s the advantage of this technique? The program can’t be changed, either unintentionally or by mischievous hands. More important, it turns out that when a computer is turned off, the contents of any RAM disappears. Thus, with a general-purpose computer, some program has to be entered into the computer each time the computer is turned on, which requires a mass-storage device where the program can be stored when the computer is turned off. Such a requirement would be an unnecessary nuisance for a special-purpose computer, such as an arcade video game. This is why the program is stored in ROM, which is immune to the computer’s being turned off. The program is part of the computer’s basic electrical configuration, or hardware.

Does this mean that a special-purpose computer isn’t really a computer, since we’ve defined flexibility—the ability to handle multiple programs—to be a necessary characteristic of a computer? The answer is yes and no. One can’t easily change the program in a special-purpose computer. However, the ROM where the program resides is all placed on a single tiny chip that plugs into the computer proper. It’s perfectly possible for the computer manufacturer (or anyone else) to write a new program, put it on a similar ROM chip, and replace the old chip. Thus even special-purpose computers are flexible enough to handle multiple programs.

VIDEO GAME COMPUTERS

Fundamentally, a video game consists of a computer program that somebody has written. Any particular video game can generally be run either on a special-purpose or a general-purpose computer. The video games that you see in video arcades or in other public places are almost invariably special-purpose computers. The most expensive and elaborate part of the arcade game is the video screen. The computer that runs the game is very small and relatively cheap.

Many video games also are made in versions that run on general-purpose computers, usually home computers. The video game programs for these general-purpose computers come on some mass-storage medium. For most inexpensive home computers, the mass-storage medium is a cassette of magnetic tape; thus the programs that you buy at your neighborhood department store are in the form of cassettes. More sophisticated (and expensive) home computers often have disk drives. Hence video games can also be stored on disks and sold as such. As with any general-purpose computer, the game program resides in RAM memory and must be read into the computer from the mass-storage medium whenever the game is to be played. The differences between special-purpose, arcade video games and those played on general-purpose home computers are interesting from a psychological point of view.

Home computers have a number of obvious advantages over arcade games. First, once you’ve purchased the computer (and the games themselves), the rest is free. But there’s a more important advantage: if you’re willing to learn how to program, you can create your own games or modify existing ones that you’ve bought. For many people, programming computers is an enjoyable, sometimes even addictive pursuit. However, learning to program is often a scary undertaking for the unin-
tiated, and strong motivation is necessary to get past the initial act of sitting down to write one’s very first program. Programming a video game often provides this requisite motivation because the games are fun. In addition, it’s fairly easy to write a very simple little program—often following some example given in a manual—and thereby to see instant results of your efforts and to have instant success.

However, if you play video games primarily because you like them, then, at least at present, arcade games are clearly superior to those designed for home computers. The designers of arcade games know that the games that are going to attract the greatest number of quarters are the games that are the most fun. So a lot of effort goes into making them fun. The designer of a general-purpose, home computer, in contrast, has different goals, and the games that are produced for these computers are incidental—they’re just one of a number of products that are designed for the computers. The computer manufacturers have to spread their energies—they have to design their computers to be fast and general; the computers must be number manipulators, word processors, and a variety of other things. Since home computers are so much more varied, there’s not quite the incentive to focus on making the game aspects of them as much fun as they possibly can be.

From a technical standpoint, there’s a second advantage to arcade games, which is that each machine can be tailored to a specific game. Consider, for example, the differences between the home computer and arcade versions of Pac-Man. The arcade version is a wonder to behold. The gaudy design on its exterior is a delight (for video players, anyway). The joystick is the perfect device for controlling Pac-Man’s motions. The screen is high quality and Pac-Man moves smoothly around on it. Most home versions are quite different. First, many home computers don’t have joysticks associated with them, so Pac-

Man’s motions have to be controlled with keys on the keyboard instead. This arrangement turns out to be considerably less satisfying. Second, the computer screen used by a home computer usually consists of the family TV set. While a TV may be adequate as a computer video screen, it is by no means ideal. You usually find that the image jiggles a bit, the focus isn’t quite as sharp as you’d like it to be, and the color isn’t quite right.

The Evolving Game

Computers are evolving astonishingly quickly. In The Micro Millennium, Christopher Evans provides the following analogy: if the automobile had made as much progress over the last thirty years as the computer industry, it would now be possible to buy a Rolls-Royce for roughly $2.75; it would get nearly 3 million miles to the gallon; and it would deliver enough power to tow an aircraft carrier.

Computer games, riding this wave of computer evolution, offer some tremendous advantages over their predecessors. By summarizing some of these advantages and by extrapolating the evolutionary process, we can speculate about where these games might be going.

ELECTRONICS

Computer games are devices that are almost completely electronic. There are few, if any, moving mechanical parts, which means that the games are easy to manufacture (and therefore relatively cheap). They are reliable; moving parts wear out, whereas electronic parts generally don’t (or at least they wear out much more slowly). Therefore, it’s much easier
and cheaper for arcade operators to maintain them. Further, electronic parts operate extremely quickly, which means that games based on them can operate at very fast speeds.

PROGRAM FLEXIBILITY

But by far the greatest advantage of computer games is the flexibility inherent in the computer programs that go into them. To see the advantages of this flexibility, imagine that you wanted to make a realistic but noncomputerized space game. You would design what amounted to a small stage set, in which space-related objects such as asteroids, spaceships, and extragalactic creatures would appear from the wings, probably suspended by thin wires and propelled by some kind of motor-operated pulley system. Such an arrangement would be very difficult to manufacture properly and would also be extremely limited in terms of what it could do, how modifiable it was, and how realistic it would appear to be.

Such potential drawbacks, however, have apparently not deterred all game designers. On July 7, 1982, the Associated Press released an interesting news item that described a new type of game being unveiled at the Knoxville World's Fair. In this game, mechanical robots marched jerkily about in a boxing ring-like arena, shooting each other with "laser guns" and otherwise creating the same kind of havoc that ordinarily takes place on a video screen. The movements of the robots, along with the firing of their formidable weapons, are controlled by the game's players (who themselves remain safely on the sidelines, manipulating joysticks that "look just like a fighter pilot's"). According to its promoters, this game represents the next logical advancement over video games—a claim apparently based on the realistic, three-dimensional quality of the game. The promoters conceded that the game was expensive—around $200,000—but seemed undaunted.

Unfortunately for the promoters, their dreams are probably unrealistic, because this game represents a step backward from the computer technology that has made video games so successful. Instead of being electronic, the game is primarily mechanical, which means that there are the same slow, troublesome, error-prone moving parts that video games have managed to eliminate. The game's flexibility is woefully limited. To simply change the setting or the nature of the robots, for example, would be enormously costly and time-consuming. The three-dimensionality of the game is obviously an advantage, but it is really the only advantage.

With a computer/video system, in contrast, there is no foreseeable limit of how complex and how sophisticated such a game could be. Suppose, for example, that you wanted to simulate a player's disappearance into hyperspace, followed by the player's reappearance some time later in an entirely different region of space. Such a feat wouldn't have been possible with a mechanical game, but it isn't difficult to program on a computer. Such a program could be easily designed so that the player would appear to be in one region until he or she pressed the hyperspace button, at which time the screen would become filled with breathtaking special effects, corresponding to the programmer's vision of what being in hyperspace is like. Then a completely different region would be made to appear on the screen. In general, the rule is: Anything that can be imagined can be programmed. And, surely, to be limited only by imagination is the ultimate in game design flexibility.

The greatest benefit of this flexibility is clearly the enjoyment the player can potentially get out of playing the game. But there are other, less obvious benefits that accrue to the arcade operator who is primarily interested in making money. It turns out that, with many arcade games, the program that runs the game is also designed to do convenient things for the

176
operator. For instance, have you ever noticed what a game does when no one is playing it? Sometimes it just sits there doing nothing, whereas other times it mimics what happens when the game is actually being played. An idle Pac-Man machine, for example, sometimes has just the Pac-Man logo on it; other times Pac-Man is moving aimlessly back and forth; and still other times, what appears to be an actual game is going on. What determines when a machine does what? The game is programmed to do any one of a wide variety of things when it’s not being played. Which thing it actually does is under the control of the operator. By changing the position of a switch inside the machine (accessible only to the operator), the program will be changed so as to do one thing or another. Thus the operator can perform little experiments to find out which of these “idle-time” activities will provide the most effective come-on for clientele. And this idle-time option isn’t the only thing that the operator has under his or her control. Most people realize that the games (actually the programs) keep track of what the high scores are for that game and display the high scorers’ initials. What’s not so well known is that a game can keep track of many other things as well: how many quarters have been inserted into it, the average length of playing time per quarter, the number of times the various levels of difficulty have been achieved, and anything else the operator might want to know about. All this information is available to the operator and can be displayed on the screen whenever he or she wants to see it.

Moreover, many games have variable difficulty. By changing the position of another switch, a game can be made more easy or more difficult. This provides the operator with a defense if the arcade starts to become frequented by especially expert players.

Let’s see how difficulty might be manipulated with Space Invaders. In this game, aliens drift across the upper part of the screen, and during each successive short period of time (say every second) an alien will drop a bomb with some probability. Suppose, for example, that the per-second probability of dropping a bomb is to be one in five. This means that every second, there’s a one-fifth probability that a bomb will be dropped or, on the average, a bomb will be dropped once every five seconds.

How does the program accomplish this, that is, what determines the bomb-dropping probability? Part of the computer program that constitutes the game of Space Invaders is what’s called a random-number generator, which, as the term implies, generates a random number each time one is requested by the program. Suppose the random numbers generated by the generator range from one to ten. This means that if you choose a random number, the probability is one-fifth that the number selected will be two or less (that is, that it will be a one or a two). Somewhere in the program will be a sequence of instructions that goes:

1. Set some variable, \( X \), to 2.
2. Choose a random number (between one and ten).
3. If the number is \( X \) or less, drop a bomb. Otherwise don’t drop a bomb.

This little sequence of instructions (which is called a subroutine) will be executed every second.

Suppose you want to make the game more difficult. One way would be to boost the per-second probability of dropping a bomb from one-fifth to one-half (which would mean that, on the average, a bomb would be dropped every two seconds instead of every five). To make this change in the program, all you have to do is change the “2” in Instruction 1 to “5.” The same general principle holds for other changes in video game
difficulty. It is in the nature of computer programs that making a tiny change in the program can have a profound effect on what the program does.

THE TECHNOLOGICAL FREE RIDE

The third major advantage enjoyed by computer games is that they are yoked to computer technology: many of the new developments stemming from computers have direct spinoffs in video game technology. As computers become faster, cheaper, smaller, and more convenient, so will the associated games. As video screen technology becomes more advanced (incorporating, for example, holographic or highly realistic 3D effects), these features can be immediately incorporated into video games.

To provide just one concrete example, consider the fabulous special effects that are now being used in filmmaking (for example, in the Star Wars series). These effects are now almost exclusively generated by computers, and an awful lot of money has been spent to develop them. A game manufacturer may not have the resources to develop these effects, but since they’ve already been developed by moviemakers, it’s not difficult for the game designers to use the technology and transfer it over to games.

Video games are evolving very fast. The primary reason for this is that the computer technology that underlies them can be easily modified. In the old days, if a game didn’t seem to be quite right, that was too bad; the cost of changing it was too great to be worthwhile. To change a computer-based game, in contrast, all one has to do is to modify the program.

A second reason for the rapid evolution of the games is that technological innovations are happening very quickly. Graphic quality, for instance, is of primary importance to video game addicts. Zaxxon is not considered to be particularly good as a game per se, but it’s nonetheless popular because of its superb graphics. Computer technology is rapidly advancing in graphic resolution and graphic realism. Resolution refers to the inherent quality of the image. Contrast, for example, a photograph in a magazine like *National Geographic* with a photograph that appears in a newspaper. Most images—both those that are printed and those that are displayed on video screens—are made up of a large number of small dots (which, you’ll recall, are called pixels when they are locations on a video screen). The smaller and more densely packed the dots, the better looking is the image. In a newspaper photograph, the dots are large enough to be seen with the naked eye, whereas in *National Geographic*, the dots are microscopic.

High resolution is desirable in video games. First, a higher-resolution image fundamentally looks better. Second, the higher the resolution, the more lifelike the characters can be; with low resolution, the best you can do is to make stick-figure-like characters. And, finally, with high resolution, the appearance of motion is better. With low resolution, motion looks jerky and unrealistic; with high resolution, it looks smoother and more realistic.

Currently, most video screens (particularly home video screens) are more akin to newspapers than high-quality magazines. As you look at the screen, you can actually see the dots that comprise the image. But this is changing. Very high resolution graphics already exist, and as their price comes down, we will see them used more and more in video games. We won’t take the time here to go into a detailed description of the technology behind high resolution. Suffice it to say that it is highly dependent on computer speed; in general, the faster a computer can operate, the more points it can paint on a screen within a given period of time and the higher the resolution can be.
MIND AT PLAY

Graphic realism refers to the degree to which the image looks realistic. In large part realism is determined by the degree to which the image is made to look three-dimensional rather than two-dimensional. Pac-Man, for example, is a two-dimensional game; it has a "flat" board with "flat" Pac-Men. In Zaxxon, on the other hand, objects are depicted in three dimensions—a three-dimensional aircraft swiftly weaves its way through a gauntlet of three-dimensional obstacles—and it is this graphics feature that has been lauded by the video players.

If three-dimensional graphics are so much better than two-dimensional ones, why aren't all games three-dimensional? The reason is simple: it's much harder to write computer programs that depict things in three dimensions. Two-dimensional graphics, in contrast, are (quite literally) child's play. However, this situation is rapidly changing as well. Computer programming ("software development") is a subject of intense research these days, and one of the results of this research is better knowledge of how to write three-dimensional graphics. Like high-resolution hardware, three-dimensional software is beginning to trickle down to the domain of video games.

A Final Note

On November 9, 1982, the U.S. Surgeon General, Dr. C. Everett Koop, delivered a speech in Pittsburgh during which he declared that video games were evil entities that produced "aberrations in childhood behavior." Koop went on to urge that video games not be played.

Not since King Canute ordered the tides to recede has an edict been issued that is destined to have less effect on its audience. Because of psychological principles that are well understood, video games are likely to last. Further, as we've seen, the games can rapidly adapt to popular moods and fads and interests. They aren't going to disappear just because the surgeon general or anyone else has commanded them to.

Moreover, video games are actually in a position to provide society with very substantial benefits. As we noted in chapter 5, video games could be implemented for educational purposes in at least three different ways: specially designed games could be run on the computers that already exist in many schools, educational games could be marketed for home computer, and video game systems and games in arcades could be modified to include educational features.

Surgeon General Koop is not alone in his antipathy toward video games. We've seen parents and other authorities exhibit enormous fear and loathing toward video games and especially video game arcades. This antipathy has resulted in individual attempts to ban offspring from playing games and in collective attempts to ban video games from communities. And yet, as we've seen, the games have enormous potential as teaching devices. We've touched on only a few of the possibilities in this book; we've only scratched the surface. At a recent (May 1983) Harvard conference, "Video Games and Human Development," for example, researchers presented positive findings about effects of video games on everything from medical rehabilitation to cognitive and problem-solving skills to social behavior. One participant reported that video game players, contrary to popular views, are relatively high academic achievers. It should be noted that there were no critics of the games at this conference, and, of course, in any very new area of study, results should be interpreted cautiously. Nevertheless, rather than expending vast amounts of energy on prohibitive mea-
MIND AT PLAY

sures that seem doomed from the start—kids are going to play video games one way or another just as they will read comic books or furtively flip through *Playboy*—it seems much more sensible to expend the energy in harnessing the educational potential. That’s really our only strong editorial stance about video games.

| Ability, 43 | Asteroids, 53, 54, 101n, 140; mental rotation, 59 |
| Acceptance of video games, reasons for, 7–8 | Astro Blaster, 54 |
| “Addictive behavior,” 18; see also Video game addiction | Atari, 96 |
| “Adults only” games, 107 | Atkinson, Richard, 117 |
| Adventure, as video game therapy, 147–48 | Attention, 46–50; selective, 46, 50 |
| Aggression, 41, 98, 129 | Auditory information, coping with, 56 |
| Air Combat Maneuvering, 145 | Auditory modality, 47 |
| Alternate reality, 31 | Babbage, Charles, 166–68 |
| Alternative world phenomenon, 31–33 | Ban, on video games, 110, 183 |
| American Graffiti (movie), 85 | Behavior, 110; collective, 92–97; healthy, 89–92; norms of society, 92; role of reinforcement in, 14; video games as model for, 97–105; visually guided, 116 |
| Amis, Martin, 110 | Bell Telephone Laboratories, 169 |
| Analytical Engine, 167 | Berzerk, 109 |
| Anxiety, 90 | Blackjack, 72 |
| Apple II, 124 | |