Representation Versus Depiction

Chris Crawford

One of the oldest battles in the games business goes under the heading, “Graphics versus Text.” Back in the bad old days, graphics were hard to do. An 8-bit CPU can’t blit much imagery, 64K of RAM couldn’t store much imagery, and the lousy displays of those days couldn’t show much imagery. Game designers found that text was often a more expressive way to communicate what was happening in the game. For example, consider this little text fragment: “Stung by the ferocity of your swordthrust, the mighty dragon thrashed about, spraying you with its lifeblood as its screams reverberated against the mountains.” That consumes a grand total of 163 bytes. Now imagine showing the same thing with graphics and sound. To show this well in SVGA would consume tens of megabytes.

Thus, in the early days graphics were the luxury item, the element that differentiated a big expensive game from a cheap one. Consumers helped drive the trend along; they found the cosmetically rich games more enjoyable.

A History Lesson

The classic morality tale here is the sad saga of Infocom. Before Infocom, text adventures were the cheapest kind of software, and they were often cheesy. Infocom brought three new elements to the picture: first, they insisted on high production values, putting odd little tchotchkes into their boxes to improve the product feel. Second, they had the best parser in the industry. Third, they made a real effort to bring some creative quality to their designs. They weren’t just cute puzzles; they really tried to capture a flavor for each game.

But consumers wanted graphics, something Infocom refused to do. Things steadily spiraled downhill until, in the late 80s, Infocom relented and began including graphics in their games. But it was too late; the company was acquired by Activision and faded from the stage.

Meanwhile, Sierra and LucasArts were making graphics adventures just like Infocom's, except that their products were heavy on the graphics. These two companies prospered as Infocom failed. What better proof do we need? It doesn't take a lot of brainpower to learn the lesson here.

Moreover, the primary restrictions against the use of text have been eroded to the point of irrelevance. Here we are in 1995 with CD-ROM and 90 MHz Pentium processors and 8 megs of RAM standard and SVGA—it would certainly seem that the graphics versus text issue is a dead one.

I would argue that the issue has instead become more interesting. Previously the matter was driven by poverty. We couldn't afford snazzy graphics, which of course made snazzy graphics all the more valuable. Snazzy graphics was the factor that separated good games from bad games. But now that we live in a graphics-affluent environment, we can revisit the old issue with a completely new approach. There are higher and more interesting issues here; now we begin to think about those higher issues. Instead of arguing the old dead issue of graphics versus text, now we can argue the more interesting problem of depiction versus representation.

What's the difference between "graphics versus text" and "depiction versus representation"? It lies in the distinction between concept and delivery. "Graphics versus text" focuses our attention on the direct sensory experience of the user; "depiction versus representation" directs our attention to the interpretive thought-processes of the user. In other words, when we frame the question as "graphics versus text," the phrasing concentrates our discussion on the computer screen instead of the user's mind. When we instead shift to the phrasing "depiction versus representation," we start to think about the psychological processes going on inside the user's mind.

One Synaptic Cleft or Three?

At this point I'd like to shift gears for a moment and discuss the way in which we view the user/computer relationship. The tendency, I think, is to view the user and the computer as two separate entities with a gulf between them, analogous to the synaptic cleft in the nervous system. Our great task is to bridge that cleft. This is a sound way of approaching the problem, but allow me to present a different approach that could be enlightening. This new approach is more complex: we imagine four entities. First there is the computer, the repository of the ideas that we build into our designs. Separate from this is the output device, the monitor. Next comes the human input device, the eyeball. Last comes the human brain. Now, here is the important point in this approach: we're not trying to transfer information from the monitor to the eyeball, but rather from the computer to the brain.

This process involves not one but three translations. First, we must translate the raw data inside the computer to a form that can be displayed on a monitor. Oftentimes this is itself a major hurdle, requiring considerable talent on the part of a programmer. Next, we must be able to communicate that information across the gap between monitor and eyeball. This too is a difficult task requiring considerable skill. Indeed, there is an entire class of professionals who specialize in this problem; they are called graphic designers. But there is one more step in the translation process: the information must be smoothly communicated from the eyeball to the brain. Now, a graphic designer might leap up at this point to declare that this, too, falls under the purview of his or her specialty, and there is certainly much truth in that. Nonetheless, I would like to focus your attention on this step as distinct from the second translation step (monitor to eyeball). The crucial distinction revolves around the interpretive strengths of the receiving agent.

For example, when we design a complex screen layout, we will take advantage of color to accentuate differences between various portions of the display. We do this because the human eye has color perception; it can perceive those color differences. Of course, if we are designing for a color-blind audience, then we must take this into account and refrain from using color. But if our audience is not color-blind, then we definitely want to take advantage of their perceptual capabilities. Failing to use color with such an audience is just stupid. Since the normal eye has the power to perceive color, we take advantage of this power to speed up translation of information.

Let us now apply the same reasoning to the third translation step, between the eye and the brain. Just as we take advantage of the color perception of the eye to increase the efficiency of our communication, so too do we wish to take advantage of the interpretive strengths of the human brain. The brain can squeeze information out of some types of images very efficiently; other images boggle the visual processing centers and cause us to stare in incomprehension.

Constructs in Communications

This is really just the inverse of a specialized display card or sound card. Consider that a souped-up display card will have more than just a bitmap for pixels. It might have sprite capability, for example. A sprite is a graphical construct, a way of organizing the visual field of pixels into logical groupings. In practice, there are many ways to logically organize a visual field: sprites, scrolling, straight lines, circles, rectangles, regions. These are all mental constructs that allow us to think about the display in more structured terms, rather than just a pile of pixels.

In the same manner, a sound board organizes sound output according to some set of constructs. A sound board is more than a DAC (a device that converts a digital value to an analogue value; this allows us to drive analog speakers with digital computers). A sound board includes additional circuitry that allows us to specify a wave of a certain frequency, or perhaps a musical instrument with a characteristic distribution of frequencies.

So here's the big idea: display boards and sound boards allow us to efficiently transfer information by resorting to mental constructs. So why not think about the reverse process? The human mind isn't an arbitrary bit processor: it has specialized regions that efficiently process certain types of information. If we take advantage of those special capabilities, then we can communicate more efficiently and effectively to our audience.

Our first task is to identify those types of images for which the human mind has special capabilities. Now, there are some capabilities built right into the retina: edge detection and motion sensitivity are two good examples. But these are low-level issues, a kind of visual pre-processing. I'm worried about the higher-level mental processing that allows us to digest complex constructs.

Facial Recognition

The first and foremost capability here is the facial recognition processor. Our brains have special circuitry for analyzing human facial expression. This is not merely a culturally learned capability; much research has demonstrated that infants can recognize and differentiate faces and facial expressions. Anthropologists have shown that the basic facial expressions are universal and independent of cultural differences. Both of these observations point to a wired-in capability.

Even more interesting is the research into "micro-expressions." These are quick expressions that flash across the face in a fraction of a second. They happen so fast that neither the expressioner nor the audience is consciously aware of their existence, yet careful experiments have demonstrated that the audience definitely, if subconsciously, perceives and recognizes the micro-expression. There are some micro-expressions that we are consciously aware of. For example, when a person lies to you, he has difficulty maintaining eye contact; his eyes will flash away briefly. We all know this and thus arises the demand, "Look me in the eye when you say that!"

Consider what this implies about the processing capabilities of the human brain in handling facial expressions. There's a great deal of processing involved in feature recognition: transforming the texture and shading information into cheeks, eyebrows, lips, and so forth. Next, that feature information must be mapped into emotional space (this expression is angry, that expression is happy). Yet, all this processing is carried out in a fraction of a second. Clearly, a great many neurons are dedicated to this processing.

So here we have our first construct. If we want to communicate efficiently with our audience, we want to use facial expressions as shorthand for emotional expression. Indeed, our colleagues in the other arts have known this for centuries. How many paintings do you see of people's knees, toes, backs, or elbows? That's ridiculous! Paintings are always about people's faces! What's the most famous painting you can think of? Right—the Mona Lisa. And what's the most talked-about aspect of that painting? Her enigmatic smile—her facial expression.

The same thing applies with photography. Great photographic portraits are great because of the facial expressions they capture. Who can forget the photograph of the South African woman runner who was tripped and injured at the LA Olympics? As she lay on the track, the pain of her injury showed all too plainly on her face, but was secondary to the burning desire to get up and finish the race. It was all there in that face, and that's what made it such a great photograph.

The movies, too, take advantage of our natural brainpower in processing human facial expression. The sequence that best exemplifies this for me is from *Star Wars*. Luke et al. have escaped from the Death Star in the Millennium Falcon and are making their getaway, pursued by enemy fighters. There follows an intense action-packed sequence in which Luke and Han Solo shoot down the enemy fighters. What is most striking about this action sequence is its reliance on faces to communicate action and emotion. You would expect such a sequence to be all zooming spaceships, roaring turbolasers, and billowing explosions, but in fact such imagery occupies only half of the display time of the sequence. The other half is taken up by character faces: frightened, concentrating, worried, triumphant. George Lucas knew that special effects only get you halfway there. You need facial expression to cinch the communication.

It is important to recognize an important point here: the facial expression is not the communicated reality, but rather an indirect representation of the communicated reality. The important idea being communicated by facial expression is emotion, but the facial expression is not the same thing as the emotion; it is a representation of the emotion. In other words, facial expression doesn't depict emotion, it represents emotion. It is an indirect indicator of emotion, but the indirection of the expression is compensated for by the speed with which the brain can translate that indirect representation into an interpretation of emotion.

Well, that was impressive. What else can we unearth with this line of inquiry? As it happens, there is one other area of human brain function that powerfully translates intermediate constructs into mental ideas: language processing. Indeed, human language processing is the hands-down champion when it comes to efficient processing of constructs.

How to Design Good Constructs

Let's step back for a moment and talk about the engineering of construct transformation. Suppose, for example, that I'm building a video board and I decide to build into this video board the construct of a straight line. This allows the programmer to call a function on my board that will draw a straight line onto the screen. I have a bunch of transistors dedicated to the logic of straight line drawing. So the programmer simply sends a call to my board that reads something like DrawLine(BeginX, BeginY, EndX, EndY). In other words, he sends a verb command, plus four parameters.

Those five pieces of information represent a straight line. They do not depict a line, they represent one. Now, the actual depiction of that line would probably take a great many more bits than the representation does. This allows the programmer to get the job done faster. He shoots five numbers down to my card, and then my thousands of dedicated transistors assemble the depiction of the line much faster than his general-purpose CPU could.

The criterion of efficiency here is the ratio of bits calculated to bits transmitted. If I can send just ten bytes down the wire and get thousands of bytes calculated for the screen display, then I've got an efficient scheme. If on the other hand I need to transmit thousands of bytes and still get only thousands of bytes displayed, then it's not a very efficient construct. Blitting bitmaps is just such a case. On the other hand, compression schemes such as JPEG and MPEG are really constructs allowing us to communicate a complex image more efficiently.

Now for a fairly simple point: we always prefer to use the most efficient construct available. Sure, we don't need line drawing routines, circle drawing routines, region handling and all that complexity; we could do everything with straight bit-blitting. But good programmers use the constructs whenever possible. If they see a way to break an image down into components that match the constructs of the system, they'll do it every time. It's always faster and more efficient that way.

There's a general rule here: communications efficiency arises from the amount of technology dedicated to the processing unit. More transistors on the video card ICs means that we can have richer display constructs. More lines of code in the software interface code makes it easier to trigger vast display changes with just a few passed parameters.

Language Processing

So now let's return to the subject of language processing, and apply these concepts. First, consider that language processing is far and away the most efficient representation of ideas ever created. We can see this in several ways. First, there's the amount of negentropy dedicated to the processing unit. A considerable portion of our brain tissue is dedicated to language processing. In terms of raw processing power, that's a hell of a lot of hardware dedicated to construct interpretation. This in itself implies that language processing should be a particularly efficient form of information transfer.

But we need not rely on such indirect means. Just look at the language itself. Consider that I can store an entire textbook on quantum mechanics in just a few hundred thousand bytes. That's as much space as it takes for a single full-screen, 8-bit image! In other words, in the same amount of information that it takes to communicate a single full-screen image, I can communicate the theory of quantum mechanics! That's efficiency!

Now, this comparison is misleading because it counts bytes rather than information as it is transferred from the computer to the brain. A full-screen image can be taken in by the eye in a fraction of a second, whereas a book on quantum mechanics will take much longer to read. So raw byte count isn't a fair assessment of the efficiency of information transfer, because we're not transferring bytes, we're transferring ideas.

Now at this point we get onto tricky ground. How do we measure the size of ideas, if not in bytes? The sad fact is, we don't have the tools for such measurements. Because we can't be quantitative here, we'll just have to rely on more hand-waving approaches.

When to Depict, When to Represent?

Clearly some things are better represented by direct depiction. In general, I think it's safe to say that most spatial problems are better communicated with depictional images than with representational words. For example, imagine yourself at a particularly tight moment in *Doom*, with monsters all over the place, spitting fireballs and shooting and clawing at you. An image or two can communicate this desperate situation with perfect clarity, but a text description of the situation would be tedious and slow to digest. Indeed, could you imagine *Doom* played in real time with a pure text display? "The door opens to reveal four imps, two cacodemons, and a hell knight. The first imp is 12 meters away and 28 degrees to the left of your sightline; the second imp is 15 meters way and 22 degrees to the left of your sightline...." Before you finish reading this tedious description, you'd be dead!

But this does not justify leaping to the conclusion that depiction is always superior to representation. Try depicting depreciation, or libertarianism, or overload, or indigestion. Sure, you can probably come up with some long-winded, tedious sequence of icons or images to get the idea across, but will they really communicate the notion faster than the word itself? I think not.

Indeed, even a testosterone-soaked game like *Doom* relies on textual representation for a good deal of its communications. When you pick up an item, it doesn't stop the display to show an animation of you bending down and picking it up; it just beeps and flashes a text message at the top of the screen. What could be faster or more efficient? Indeed, the status bar at the bottom of the screen is highly representational in style. The weapons indicator, for example, doesn't show little pictures of different weapons; that would be a waste of screen space. Indeed, it jumps TWO levels of representation: it presents numbers that index to weapons. The designers knew when to depict and when to represent.

The obvious and eminently reasonable conclusion is that there are some ideas better communicated with depiction and some better communicated with representation. But I would caution against using this conclusion to justify whatever conclusions one desires to reach. There are qualifying considerations that make application of this dictum a matter requiring much judgment.

The first qualifying consideration concerns the spatiality of the idea being communicated. The spatial position of monsters in Doom is a vitally important consideration, so the best way to communicate that information is with direct depiction. On the other hand, the list of weapons available has no spatial content, so there is no need to use depiction in the display of the list; representation works better.

The second qualifying consideration is the desirability of taking advantage of the brain's strengths. We want to use images that the mind can most quickly interpret, so we want to take maximal advantage of the two areas of image processing that the mind is particularly adept with: facial expressions and language.

A brief digression here: the real strength of the mind does not lie in reading, but in language comprehension. Language is a human talent that developed hundreds of thousands of years ago and is built into our brains. Writing and reading are far more recent adaptations of language, and require more explicit training. Thus, the interpretation of text is really a two-step process: reading, which is still rather slow and cumbersome, and language interpretation, which is lightning fast and powerful. It's as if we had a video display board with immensely powerful display constructs, but it uses weird voltage levels for its signals, requiring us to send our commands through a cumbersome interface board before we can tap into the awesome power of this video board.

However, there is a way around this problem with language and reading. Our appreciation of spoken language is unhindered by the clumsiness of reading; it is for this reason that I have identified text-to-speech conversion as the single most important technology for game design. We need this technology desperately; when we get it, it will transform game design far more than the CD-ROM did.

Returning to the main theme, then, we conclude that we definitely want to use language whenever appropriate. For the moment, the value of language is compromised by our reliance on written text as opposed to spoken language, but the advent of practical text-to-speech technology will change matters dramatically.

Directness Versus Indirectness

There is another major consideration, though, in the consideration of the use of representation versus depiction. It has to do with the nature of directness versus indirectness. In many ways, this consideration cuts very close to the heart of the issues.

Representation is intrinsically indirect, while depiction is just as intrinsically direct. That is, a depiction of something shows you the thing itself, while a representation of the thing is not the thing, it is a pointer to that thing. While a representation might mean the same thing as a depiction, there is always a step of interpretation between the representation and the thing itself. This step of interpretation distances the audience from the thing. By contrast, the direct depiction is as close to the audience as it can be.

Directness yields power. If you and I are having a heated disagreement, I could pass you a note saying, "I am angry." This would communicate the message indirectly. But if I were to bare my teeth and snarl, the directness of my expression would have greater impact upon you. This is why movies are generally more popular than novels, and television is more popular than radio.

But directness is a two-edged sword; it has advantages and disadvantages. Most designers are acutely aware of the advantages of directness, but ignorant of its disadvantages, or the corresponding advantages of indirectness.

Consider this example of the detrimental effects of over-specified depiction: suppose that you wish to communicate a simple idea about home safety—that you shouldn't leave electrical wires lying about on the floor, especially across high-traffic areas. Now, you could communicate this with representational text like so: "It is unsafe to place electrical cords across walkways; people might trip on them." That's pretty good. But golly gee, wouldn't it be more impressive to make a video depicting the problem? Here's old grandpa shuffling down the dimly lit hallway. His eyes aren't so good. The camera at floor level shows his foot catching on the electrical cord. Next, a slow-motion shot of his face as he loses his balance, and then a full shot of his body crashing into the floor. Then we follow up with normal-speed sounds of voices crying out, "It's grandpa! He's hurt! Call an ambulance!"

This is certainly dramatic video. But consider its message. Does it say that electrical cords are dangerous to everybody, or only to old people? Perhaps a viewer might tell himself, "Because there are no old people in my home, I don't need to worry about this problem." Or perhaps the viewer will draw the conclusion that the problem lay in the poor lighting in the hallway. Here is the problem: the video doesn't really address your situation or my situation; it addresses a single case. We are expected to generalize from that single case to a variety of cases, but that process of generalization is fraught with confusion. It is entirely too easy to generalize incorrectly. Is the tale of grandpa's fall a warning about the frailty of old people, or the dangers of poor lighting, or electrical cords? The tale never specifies which.

Note that the textual representation has no such problems. It clearly specifies the scope and nature of the point being made. It is a more precise communication; it has clearer focus; it gets its point across better than the video does.

There's another advantage to representation: it is not only more precise, it also offers more expressive possibilities. Consider these lines from a Bob Dylan song: "And take me disappearing through the smoke rings of my mind, down the foggy ruins of time, far past the frozen leaves, the haunted, frightened trees, far past the twisted reach of crazy sorrow." Consider the expressive richness of these words. Their power springs from their indirectness, from the power of the combinations that the words permit. Consider, too, the futility of trying to communicate these phrases with depiction. Just what would "the foggy ruins of time" look like? I suppose that you could come up with an image that does the job, but could any actual depiction have the suggestive majesty of representational phrase? And then there's "the smoke rings of my mind"—even further beyond the reach of depiction. The indirectness of representation makes possible an expressive range that completely outstrips depiction. Keep in mind that this is an advantage of representation over depiction, not necessarily text over graphics.

Indirection in Programming

At this point I'd like to draw an analogy with programming practice. Programmers use indirection heavily, and so have developed a clear understanding of its value. Allow me to walk you through some of the lessons that programmers have learned about indirection.

Let's talk about numbers. You can refer to numbers in a program in many ways. The simplest and most straightforward is to explicitly name the number. For example, suppose we wish to determine if a variable has grown too large. So we have an IF-statement like so:

```
IF MyVariable = 25 THEN...
```

In this case, I have decided that 25 is the maximum value acceptable for MyVariable.

But programmers know that this way of expressing it is often undesirable. Suppose that I have three or four places in my program where I repeat the same test on MyVariable. This is trouble waiting to happen, because the odds are high that if I later need to change the 25 to, say, 26, then I have to hunt down every case of IF MyVariable = 25 THEN... All I need is to miss just one case and I've got myself a messy, ugly bug.

Of course, every beginning programmer knows the solution to this problem: at the beginning of the program you define a constant called, say, TestValue, and you set it equal to 25. Then your code should read:

```
IF MyVariable = TestValue THEN...
```

The big advantage of this approach is that, if I choose to change TestValue from 25 to, say, 26, then I change it at a single place in the program. This greatly cuts down on stupid bugs.

This is all pedestrian programming, but there's an enormously important point here: the solution involves recourse to a higher level of indirection in the representation of the number. The old, dumb way used the direct value, but the solution used a representation of the value (TestValue) instead of the value itself. In other words, the line:

```
IF MyVariable = TestValue THEN...
```

tells the computer to go back and look at whatever value was assigned to TestValue. It doesn't present the value itself, it instead represents the number with a name.

But this is only the first level of indirection. The clever programmer might wish to change the value of TestValue during execution. At first, its value is 25, but later on, he wants it to be 26. To do this, he makes TestValue a variable. This is a more powerful approach; you can do all sorts of snazzy tricks once you make TestValue a variable. You can insure that the IF-statement will trigger under different conditions. But there's a bit more work involved in making the variable work. You have to declare the variable, specifying what kind of variable it is (byte, word, or longword). Then you have to initialize it. Then you have to modify it during program execution.

Moreover, your workload as a programmer is now greater. When TestValue was a constant, you could always check its value statically by simply looking up its definition in the program listing. But as a variable it's much trickier to check up on. You have to halt the program in mid-flight and examine its value with a debugger. Granted, this is easy work, but it's still more work than simply looking up the constant declaration.

Note the drift: in moving from constant to variable, we increased our programming power and gave ourselves interesting new capabilities. But at the same time we made the program harder to understand and increased our workload.

But it doesn't stop there. We can take it another level of indirection higher by replacing the variable with a pointer to a variable. This pointer is really an address that contains the value in question. It's easier to think in terms of an index into an array. Now, why would we want to do this, you might wonder. The advantage is that the pointer or the index can be easily and simply altered to point to completely different values. Thus, an index of 1 might point to a value of 25, while an index of 2 might point to a value of 57, and an index of 3 might point to a value of 19. The big difference here is that now we are contemplating changing the value of TestValue, doing so frequently, and using a large variety of numbers that bounce all over the map. We are now thinking in broader terms about TestValue. It could be almost anything, and it will be many different things at different times.

There are many other advantages of pointers. For example, a linked list is a data structure presuming the use of pointers. Its advantage is that it can be of any size, so we don't have to worry about reserving large amounts of RAM that might not be used. Moreover, the linked list has the additional advantage that editing the list is much faster than with a conventional list: you insert or delete an entry somewhere in the middle of the list, and simply modify two pointers and poof! the list has been edited. With a conventional list, you must move massive amounts of data around to either make enough space for a new entry in the middle of the list, or to close up space left by a deleted entry. But all of this is impossible without the indirection permitted by a pointer.

The same thing happens with certain types of sorts or searches. Rummaging through huge amounts of data can be made much faster by using pointers. Entire books have been written on such techniques. My point is that the central idea behind many of these powerful methods is the use of the indirection of a pointer. Indeed, some of the most powerful methods involve double indirection, that is, pointers to pointers (or handles, as they are called). Such methods can be truly mind-boggling, requiring a great mental exertion to decipher. Once you figure them out though, they are truly impressive and elegant.

Indirection also costs computer time. When you use a constant, it goes straight into the object code and runs very fast. When you use a variable, the computer has to load the value in from RAM, a slower process. When you use a pointer, the computer must first fetch the pointer and then deference it to obtain the value, an even slower process. And when you use a handle with double indirection, the computer must go through two dereferencing processes before it can finally fetch the desired value. Obviously, indirection slows down the computer in much the same way that it boggles the mind.

Note this also: our mental image of TestValue has shifted. When we were low on the scale of indirection, it was easy to think about what TestValue represented: it was a single number, 25. Now, a single value like 25 is something you can wrap your fingers around, something clear and almost tangible. But as we have moved up the scale of indirection, our mental image of TestValue has grown fuzzier. First it was a number, 25. Then it was a name (TestValue) representing a number, 25. Then it was a name, TestValue, representing a variable whose value was initially 25, but might change later. Later on, it became a pointer or an index to a value in a list of values. Are you starting to become befuddled? Is all this indirection making you think too hard to keep up with what is intended? If so, then you are experiencing the other half of the representation/depiction tradeoff. As I have already shown, higher levels of indirection permit more power and expressive range, but they also require greater amounts of interpretive labor.

Other Lessons from Programming

Have you ever noticed that programming languages themselves are intensely representational in style? That is, they're always done in pure text. There have been a few experiments in graphically oriented programming languages, but these have been little more than fascinating failures. When it comes to speed and power, the representational approach is so clearly superior that programmers have no problem choosing it. Indeed, the notion that programming might be carried out in a more depictional style is so alien that not many people take is seriously.

Consider also user interfaces. One might think that the steady shift towards GUIs might suggest a preference for depictional approaches over representational strategies, but I see it more as a shift away from overly representational styles. That is, DOS is utterly representational in style, too much so. *Windows* or the Mac interface are certainly more depictional, but I see them as less intensely representational. An icon, after all, is not a depiction of an item but rather a graphical representation. I suspect that user interfaces will continue to evolve away from representation and towards depiction, but I don't see this as a process without limits. At some point, the shift will stop, because representation offers decided advantages over depiction. That shift will stop when the balance between representation and depiction is less lopsided than it now is.

"But Depiction is the wave of the future!"

I must now address a common misconception. The belief here is that we are now raising the Video Generation, a bunch of kids who've been spoiled by so much video that they will have no patience with the tedious task of reading. If you want to reach the Video Generation, pundits pontificate, you'll have to do so with video. They'll never read text. All this high-falutin theory will crumble against the brick wall of audience requirements. Sure, these arguments might work for the Old Generation, but the Video Generation is different.

Balderdash. Reading is not a dying skill, and never will be. Text is a permanent part of civilization.

First, I will note that the claims about the new generation are based on a biased comparison. If you compare the average kid today with the adults around you, you will readily note that the adults are much heavier readers than the kids. Ergo, all adults are heavier readers than all kids, and as the kids grow up, reliance on text will diminish.

The flaw in this reasoning is that the samples are biased. The population of kids is fairly heterogeneous, but the population of adults is quite strongly sorted by wealth. Your third-grade kid could well be sharing a classroom with kids from all sorts of backgrounds. That third-grade class is a pretty fair cross-section of America. But how many of your friends are homeless? How many are drug addicts? How many are hopeless losers? You're probably upper middle class, and your friends and acquaintances are all highly educated, motivated people who read all the time. If you compare your adult circle with a third grade class, you're bound to find the third graders less interested in reading. But if you were to wander through the apartments in a public housing project, sampling the reading tastes of adults there, the odds are high that those adults will have lower propensity for reading than the average third-grader.

Let me put it to you another way: consider the reading habits of the brightest and most successful people you know. Now compare these with the reading habits of the least successful people you know. Quite a difference, isn't there? Smart people read—how do you think they got so smart in the first place?

So here we have a society that rewards reading with financial success. With such a clear relationship between reading and wealth, do you really think that people will stop reading? I think not. Sure, there will always be plenty of losers staring slack-jawed at videos put together for them by slicksters who go home at night to read a book. But the people with the disposable income will always be good readers.

This doesn't mean, of course, that reading is the recreation of choice of the wealthy. The only real point here is that reading is a major factor in financial success, and most of that critical reading is in some way job-related. It is entirely conceivable that some future generation of high-rolling executives will spend their days scanning reports, writing memos, and sending faxes, then come home to trashy video to unwind. But the key observation is that the skill is not going away. For many practical reasons, text remains the quickest and cheapest way to convey large amounts of information.

Which brings me to another point: video will always be reserved for lowest common denominator information, while text will be the only way to convey less common types of information. Consider this journal (*Interactive Entertainment Design*), for example. Hey, I could prepare it as a multimedia extravaganza, with little animated dolphins leaping over the waves and dancing clowns and all sorts of other nifty stuff. I'd probably be able to communicate the esoteric points of this essay more effectively. But doing so would be immensely more expensive, and with only 250 subscribers, I couldn't afford it. Text is the ideal format for distributing large amounts of specialized information.

Here's another way of looking at it. I just got another Barnes & Noble catalog in the mail. There are nine books featured on the cover—presumably the nine titles that would appeal to the broadest audiences. They include an autobiography of Norman Schwarzkopf, a biography of Tsar Nicholas II, a book on winemaking, a novel about medieval England, the Physician' Desk Reference, another book on the Turin Shroud, a photo book about cats, a book about comics, and an atlas of Great Britain. Now I ask you, when was the last time you had available anything comparable to these books on video? Yes, perhaps you saw the TV series on Nicholas and Alexandra—but come now, that was light fare; it wasn't anywhere near as detailed as a 462-page biography. Yes, you've probably seen some good shows about cats on PBS, and a documentary on Norman Schwarzkopf, perhaps even a show on winemaking—but are any of these videos you've seen as thorough or complete as the books offered in this catalog? In my library I have 23 books by or about Desiderius Erasmus—have you ever seen a single video about the man? I have 59 books about Arthuriana (King Arthur stuff)—when was the last time you saw a video on Arthur? And let's not just talk about stock retellings of the basic legend; when was the last time you saw a video on the Roman invasion of Britain (one of my books) or the revolt of Boudicca (another one) or the swordmaking techniques of Celtic smiths (yes, I have that too) or all the Irish bronze swords extant (yep)? Let's face it, text will always be the best way to address special needs—and in a world as big as ours, everybody's needs are special.

Ontogeny and Phylogeny—Oh No, Not Again!

The point is made especially clear when we consider the ontogeny and the phylogeny of writing systems. Books for very young children are mostly pictures with very few words. For older children there are still pictures, but more words. When we get to teenagers, pictures have receded to a minor place. And when we start talking about my kind of books, we're talking gobs and gobs of words and very few pictures.

The phylogeny of writing mirrors this sequence. The earliest writing systems of Stone Age peoples were simple depictions of hunters, prey, and natural phenomenon. These images later evolved into pictographs, images that were still essentially depictive but added some symbolic functions. Pictographs evolved into glyphs and hieroglyphs, more complex writing systems in which the representational component of the image took on even larger meaning. The big leap came with the introduction of the alphabet, which completely discarded the depictional aspect of the image in order to obtain greater representational power. It is significant, however, that while alphabetic writing systems make no attempt at depictional verisimilitude, they still attempt to preserve aural verisimilitude—phonetics. The word "cat" may not look like a cat, nor sound like a cat, but the pronunciation of the individual letters does match the pronunciation of the word (with the inevitable exceptions such as 'enough', 'women', and so forth).

My conclusion is that communications systems naturally evolve away from depiction and towards representation. To suggest that the dawning of the video age will reverse millennia of human experience and billions of individual case histories is silly.

Video is here to stay, and will always be the mass-market medium. But video will not put bookstores out of business. Reading is not an endangered skill. We don't need video to communicate with the new generation.

Conclusions

Where does this leave us? I hate to say this, but few simple conclusions can be drawn from this discussion regarding the application of representation versus that of depiction. We face trade-offs in choosing between the two. Representation is the more powerful of the two, and it permits a range of expressions that depiction can never match. But depiction is more direct, more immediate, and often more compelling. The task of the skilled designer, then, is not to cram as much depiction into the product as possible. The real task is to determine where to use more representational approaches and where to use more depictional approaches. It's a matter of balance.