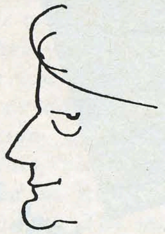
Missionary Position

THE ART OF THE COMPUTER SCREEN



by Theodor Nelson

The interactive computer screen will be mankind's new home.

The sooner we understand it, the better.

Your Interactive Graphic Screen

You can get graphical screens for your personal computer already. Most of the prebuilts are being offered with graphic screens; you can put them on the others as accessories.

People want them for games. People want them for practical uses. And people want them for sheer excitement. With this equipment—and suitable programs—you can make your own cartoons, your own interactive pictures, your own complete console for living.

But so far the programming to be seen on hobbyist screens has been rudimentary and difficult. (For instance, every time you see a Video Dazzler, you generally see Steve Dompier's same little picture of a champagne bottle pouring.) There are few interactive animations for these systems, as yet.

Just what are we talking about?

The Commodore PET offers a screen with text and certain picture capabilities. Short line segments, vertical and horizontal, can be combined into pictures or animations. Patterns of dots may also be put on the screen, but in certain very restricted arrangements.

The Radio Shack computer allows a certain pictorial capacity with little squares, 48 (vertical) by 128 (horizontal). Separate TV required.

The Merlin video board for S-100 machines allows graphics of 96 by 128 squares. Separate TV required.

The Video Dazzler from Cromemco offers color graphics of 64 by 64 squares in eight colors. This is also an S-100 system. The Super-Dazzler, still in the works, promises much higher resolution, but we don't know when. Separate TV required.

The Levine Board (available from the Itty Bitty Machine Co., Evanston, Illinois) is an S-100 board offering 256 x 192 squares of graphic animation. Unlike the Dazzler, it does not slow the computer down. Separate TV required.

The Compucolor machine, a prebuilt with color video included, offers graphics in color, 192 x 160 boxes. This has certain peculiarities, restricting the display to only two colors within small regions. But the machine is inexpensive at $3000, considering all it does.

These are only a few of the many fabulous pieces of equipment *now* on the amateur-computer market, offering different kinds of interactive pictorial capability. We won't even get into the programming problem. But we will talk about what it's for.

More and better will be available soon. The thing to do now is understand what you can *do* with the screens, understand what they portend, and prepare.

What's Coming

Perhaps what will matter most in the coming decade will be the design of interactive systems for people to use in their everyday lives. These will resemble nothing so much as video games; but they will be video games about real life and video games for the mind. Tomorrow's desk, tomorrow's automobile dashboard, tomorrow's control panel—all these will use the computer screen as a magic viewer and magic wand; a gateway to what we want to see or do.

How hard it is to write about this in a column! If you saw it in front of you you'd understand it immediately—the smallest child would. Five years from now you'll see it everywhere. But right now, at this instant, the brink of the new world, I have to fumble with words.

Earlier we saw how easily a computer can be made to behave interactively. The general principle is this: something appears on the screen, typed by the computer; you type something back (take your time); the machine replies at once with something new.

The Most Important Computer Program Ever Written

All the computer-screen systems of tomorrow were foreshadowed by one astonishing program created by an isolated genius in the early sixties.

A stern, thoughtful young man named Ivan Sutherland, then a graduate student at MIT, was given permission to use the special graphics computer at Lincoln Laboratory.

Lincoln Laboratory is a stern, thoughtful complex on the outside of Boston where they do electronics research associated with warfare. The special graphics computer was the TX-2, built especially for experimentation with pictures on computer screens. What did this have to do with war research? Only that the military finds out about new developments first, and so that is where computer screens got their first boost.

Ivan Sutherland, in any case, showed a rare vision in what he chose to do with the TX-2 computer—and how he did it.

He created a system that allowed you to draw on the screen. For this reason he called his program SKETCHPAD.

The SKETCHPAD program allowed you to draw on the computer screen as you might on paper—but with remarkable new capabilities.

You could draw a picture on the screen with the lightpen—and then file the picture away in the computer's memory. You could, indeed, save numerous pictures in this way.

You could then combine the pictures, pulling out copies from memory and putting them amongst one another.

For example, you could make a picture of a rabbit and a picture of a rocket, and then put little rabbits all over a large rocket. Or, little rockets all over a large rabbit.

The screen on which the picture appeared did not necessarily show all the details; the important thing was that the details were *in* the computer; when you magnified a picture sufficiently, they would come into view.

You could magnify and shrink the picture to a spectacular degree. You could fill a rocket picture with rabbit pictures, then shrink that until all that was visible was a tiny rodent; then you could make copies of *that,* and dot them all over a large copy of the rabbit picture. So when you expanded the big rabbit till only a small part showed (so it would be the size of a house, if the screen were large enough), then the foot-long rockets on the screen would each have rabbits the size of a dime.

Finally, if you changed the master picture—say, by putting a third ear on the big rabbit—all the copies would change correspondingly.

The drawing operation in SKETCHPAD was very special. The user would point with the lightpen at a starting-point on the screen, and draw a line from that starting-point to any other position. A line would extend from that position to the tip of the lightpen, and when the lightpen moved, so would the line, stretching like a rubberband from its starting-point. This was called a "rubberband line;" it allowed the user to try out different positions without erasing.

Then, when the user wanted to join two lines, there was a way of *attaching* them: two lines that were attached remained attached, even when the user decided to move one of them.

One of the most important aspects of SKETCHPAD was this: working on a screen, you could try out things you couldn't try out as a draftsman on paper. You were concerning yourself with an abstract version of the drafting problem: you didn't have to sharpen any pencils, or prepare a sheet to draw on, or use a T-square or an eraser. All these functions were built into the program in ways that you could use through the flick of a switch or the pointing of the lightpen. And the drawing itself existed in an abstracted version, that could be freely changed around with no loss of detail.

Thus SKETCHPAD let you try things out before deciding. Instead of making you position a line in one specific way, it was set up to allow you to try a number of different positions and arrangements, with the ease of moving cut-outs around on a table.

It allowed room for human vagueness and judgment. Instead of forcing the user to divide things into sharp categories, or requiring the data to be precise from the beginning—all those stiff restrictions people say "the computer requires"—it let you slide things around to your heart's content. You could rearrange till you got what you wanted, no matter for what reason you wanted it.

There had been lightpens and graphical computer screens before, used in the military. But SKETCHPAD was historic in its simplicity—a simplicity, it must be added, that had been deliberately crafted by a cunning intellect—and its lack of involvement with any particular field. Indeed, it lacked any complications normally tangled with what people actually do. It was, in short, an innocent program, showing how easy human work could be if a computer were set up to be really helpful.

As described here, this may not seem very useful, and that has been part of the problem. SKETCHPAD was a very imaginative, novel program, in which Sutherland invented a lot of new techniques; and it takes imaginative people to see its meaning.

Admittedly the rabbits and rockets are a frivolous example, suited only to a science-fiction convention at Easter. But many other applications are obvious: this would do so much for blueprints, or electronic diagrams, or all the other areas where large and precise drafting is needed. Not that drawings of rabbits, or even drawings of transistors, mean the millennium; but that a new way of working and seeing was possible.

The techniques of the computer screen are general and applicable to *every*thing—but only if you can adapt your mind to thinking in terms of computer screens.

It should be obvious that you can use the techniques of computer screens to do bookkeeping, writing, design, architecture; to plan how to move your furniture, to catalog your goldfish. Whatever your field, whatever the kind of data, you can use the computer screen to store, retrieve, choose, draw, rearrange, correct, adjust; to see instantly the results of an idea, and change the idea accordingly; to enact your work, and see it whole, rather than guess at its consequences and work with little pieces.

This is, of course, completely the opposite of "the computer" that so many people think of: cold-blooded, demanding, and requiring everything people tell it to be set up in difficult codes.

The Failure To See

In the fifteen years since SKETCHPAD, no initiatives worth discussing have been taken by the computer industry to bring us closer to a world of computer screens for everyone. It was not in IBM's economic interest to make computers easy to use, but to sell complication and make it sound necessary. The computer companies, mostly following like goslings after IBM, have simply brought out smaller computers and cheaper terminals. (Screens have finally appeared, but merely because it has become cheaper to put out a terminal with a screen than a terminal that prints; but most screens show no pictures.)

The brainlessness of the ordinary computer companies has now become plain, however; for personal computing has arrived with a bang, and with it the certainty, for all to recognize, of a computer-screen future.

Most people have not seen SKETCHPAD, or the movies of it, and nobody was motivated to tell them. Even many people in the computer field, technically-minded and preoccupied with their own areas, have failed to see the revolutionary implications of these developments. Indeed, many see computer graphics as worthless frivolity, rather than what it is: the beginning of a new world.

In the meantime, the hundreds of young people who have seen what would soon be possible with computer screens have retreated to the universities, or elsewhere, to wait out the situation.

And of course the public has hardly heard of it at all.

Of course most people are not yet prepared to think in terms of computer screens. There is some wrench, some about-face required, much like that of learning to live with the printing press, or the telephone. But for many it will only take five minutes of real interaction to see what's coming, and start thinking about what *they* want.

Some Important Screen Systems

The computer screen is something new on earth. That few people have seen how to use them, or seen how immense will be their impact on society, should perhaps be forgiven. People didn't know what they had on their hands when movies were first invented, either. (I've discussed this in "Getting It Out of Our System," in *Critique of Information Retrieval,* edited by Schechter and published by Thompson Books, 1968.)

But a few dazzling examples have begun to show us how computer screens should be used.

SKETCHPAD showed us what could be done at the screen with pictures. Another system, NLS, has shown what can be done with text.

Douglas Engelbart's "NLS" system, created at Stanford Research Institute, allows a user to read from screens and write on screens, instantly pulling to the screen whatever he wants from large quantities of stored text—or putting new things away.

The many users of Engelbart's system can share the writings that are stored in it, and even make marginal comments on each other's work—all stored electronically.

The only drawback of NLS—aside from its presently high cost—is that it is not for beginning users. To learn its use takes ten days, not ten minutes. The kind of performance it offers is terrific; later systems of this kind will have to be simpler for most people to use. But Englebart has shown the way.

The third spectacular example is Alan Kay's "Dynabook" at Xerox Palo Alto Research Center. The Dynabook is simply a small computer with screen, keyboard, and SMALLTALK language. But the dazzling screen manipulations—pictures, animations, fancy text—are exciting to everyone.

A fourth example is PLATO. The PLATO system, created by Donald Bitzer at the University of Illinois (and now being sold by Control Data Corporation), allows a thousand users, all over the country, to have highly interactive computing and graphics on super-looking graphic screens. (See "PLATO Makes Learning Mickey Mouse" by Elisabeth R. Lyman in *ROM,* September, 1977.)

PLATO costs far too much, and is in its present form a dead end, since it uses an expensive central computer instead of little private computers, like Dynabook; but it remains the most publicly visible system for the human use of interactive computers.

The Anatomy of the Computer Screen

The computer screen is something new on earth, and so we are just discovering—and inventing—its nature.

What to use it *for* is obvious: everything. But how to design overall systems is another question. It can be very hard to do well.

However, the different things people have been putting on the screen can be described and categorized, together with their uses so far.

A *cursor* is a movable marker on the screen. When you control a cursor, it serves to tell the computer program what you are pointing at. When the program controls a cursor, it is a way of showing you what you should be looking at, or where the next thing you type will appear. (The Latin root of "cursor" means runner, and the cursor does indeed run around the screen for you.)

A *menu* is a list on the screen of things the computer is ready to do for you; and if you point at one of the items on the menu, the computer then does it. If there is a dot of light to point at, that is called a *lightbutton.* If the menu is composed of symbols or pictures to point at, it is a *symbol menu.*

A *menuplex* is the complex of menus a user may weave through.

Often a screen will be divided into sections having different functions or activities going on. These are called *panels* or *windows.* A place set aside with no borders is simply an *area.*

If advice appears as to what you may do next, it is called a *prompt.* If an area is set aside for prompting, it is the *prompt area.*

Some systems expect you to type whole commands in, and leave an empty line for the purpose at the top or bottom. This is the *command line.*

Sometimes a symbol on the screen will indicate what is going on; when something else begins, it changes to another symbol. This is a *ding-dong.* (If a cursor changes shape depending on what's happening, this is a *ding-dong cursor.*)

Pop-ins are symbols that appear out of nowhere under certain conditions.

A *peekaboo* is something that appears on the screen if you touch a smaller symbol (the *doorbell*).

These names, of course, give no flavor as to what you can do with them.

Just for an example, let's invent a console for a musician: someone who gives live performances, and plays a piano-type keyboard. Let's call him Irving. We'll call the system SAM, or System for Augmented Music.

Very well: a piano-like keyboard, for input.

The keyboard connects to a central small computer, which actually generates the sounds. Probably there are several computer chips; one to handle all the timing and switching and screen-work, several more to create the tones. (Making tones by computer chip is now becoming cheaper and simpler than having a whole music synthesizer, which has to be wired up specially.)

There are loudspeakers: let's be generous and say eight.

And there is the screen, just above the keyboard. A lightpen dangles before it, ready to be pointed more specifically. Irving will press a footswitch when he wants to tell the computer to act on what he is pointing at.

Irving the musician sits down at his instrument. On the screen, in the main panel, is a menu of voices he may want to play in, like organ-stops. Besides the usual names, like FLUTE and DIAPASON, he also has voices called BAUTANT, TWEEDLE, GRUNDOON, and SNAZ—voices he created through the screen.

With lightpen Irving now selects the name of the voice he wants to play in, BAUTANT. That name now appears on a top reminder line, saying that this is the voice he is playing in.

But more: at the bottom of the screen appear some pop-ins, a miniature map of the loudspeakers. Aiming his lightpen between the speakers on the map, he tells the machine where he wants the sound to appear to be coming from: in this case, the center of the room.

And he plays for awhile.

Now he decides to change the sound. Pausing for a moment, he touches a doorbell next to the word BAUTANT in the main panel. A diagram of the sound appears; swiftly he modifies that diagram. He lets it go, releasing his foot on the pedal; the diagram disappears, but he is playing now in the newly modified sound.

(Note that this part of the facility actually exists in Alan Kay's office at Xerox.)

Now suppose Irving wants to play an orchestral piece with himself (like Mike Oldfield's "Tubular Bells").

Basically it works like this.

As Irving plays on the keyboard, SAM "notes" the timing and pressure of each key-pressing. The timing is noted to the thousandth of a second, the touch about as subtly. Thus an accurate recording is made of what keys were pressed when; this is recorded by the computer as a list of symbols.

This list can be used to replay music just as if it were coming in live. Irving merely touches a lightbutton labeled, "Play It Again, SAM."

And as the computer replays each voice, Irving adds yet another "instrument" to the swelling orchestra—chosen from the voices listed on the screen.

Naturally, each of these instrumental contributions can be modified later if he doesn't like it.

Note that this is not exactly a canned recording. Each of the separate instrumental contributions can be left out, and Irving can replace it with a live performance.

This is something like having many synchronized tape recordings: except that each one can be modified, changed in its sound, or changed in its apparent location—all through the screen.

This is just an example. We could design panels, menus, symbols in great detail, but there's no point right now. These machine functions were just chosen off the cuff; any other things you might want a machine to do can be handled as easily. (But note that a number of computer musicians are building systems for themselves that are rather like this one—including Carl Helmers, the editor of *BYTE* magazine.)

Today, screen-facilities like these are so expensive and esoteric as to be available only to our air traffic controllers, utility companies, and war-control centers. But as the costs go down (and the programming becomes easier), we will have graphical computer consoles for everything.

Consoles for writing, for making music, for communications switchboards, for executives making telephone calls; consoles for artists (that's right), moviemakers, newsmen; for darkroom work, pottery, origami, woodcarving.

Basically they will all have computer, keyboard, screen, disk memory. The interconnections to the outside world will vary, and hence the cost.

But they will use menus and panels and the other things we have mentioned. No systematic study has ever been made of the art of such layout, the menus and symbols and their relation to what you want to do. The closest book so far is James Martin's *The Design of Man-Machine Dialogues,* which treats this study as a form of engineering, not an art.

Views

If something is in a computer system, there must be a good way to view it on a computer screen. There may, indeed, be some new and special way.

Since programs can be created to zip through stored data and analyze it in various ways, someone who is concerned with a particular form of data naturally has an interest in creating viewing-programs specially suited to those concerns.

For instance, text.

Someone interested in text naturally wants to run it forward and back on the screen, meaning up and down, at great speed; to be able to see all the headings, and from the list of headings to jump to the text beneath any one of them, just by pointing.

(Sophisticated users will probably need text systems with a much more elaborate structure, however; see *Computer Lib.*)

If you are interested in such things as census data—complicated boxes of numbers—the computer can be programmed to analyze it into all kinds of statistical breakdowns: numerical tables highlighting various aspects.

But wait! Why be satisfied with numerical tables? The graphical screen can be easily programmed to give you bar charts, pie diagrams, diagrams in proportional shades of grey. Or even new kinds of diagrams that can be rotated in multiple dimensions, presenting to the eye things you could never see before.

Then consider maps.

When the computer stores maps, it can store them in new forms. Through the screen you can magnify the map from the entire nation down to an individual street, if the information is there; no, down to the fine print on a chewing-gum wrapper in the gutter, if *that* information is there.

Map data is two-dimensional. But the computer can also hold information allowing it to present three-dimensional scenes.

Some screen-systems show a three-dimensional object as a system of lines—as in *Star Wars,* where the map of the Death Star, in three dimensions, is brought to the good guys just in time by Artoo Detoo. The three-dimensional line-drawn map in the briefing was in fact created on just such a system, on *our* planet.

Such three-dimensional mapping will become of increasing importance, especially in architecture, research, and teaching.

But once you have three-dimensional data—that is, information precisely describing the coordinates of spatial objects—it need not be viewed as lines only. Certain very expensive viewing-systems permit you to see it as a *colored photograph,* showing exactly how such scenes or objects would appear to a living viewer. And this offers the advantage that you need not build the object physically to visualize it, or view it, or photograph it. You need only create the data structure that represents it in the computer system.

NASA has used this approach very successfully, to make "photographs" of what certain complex space equipment would look like if they built it. This way both Congressmen and engineers can be sure they're talking about the same thing.

Soon, it will be possible to do trick visual effects like the big ones of *Star Wars*—great rockets, planets, monsters, scenery, what have you—without having either models or made-up actors. It will only be necessary to create a computer representation of the desired stuff, and the computer will make the movie or the visual insert, frame by frame.

Finally, one clever engineer thinks he can put this all in your home or school. The big fancy systems for fake photography, the kind you'd use for *Star Wars,* cost a great deal of money, like a million dollars. But Ron Swallow of HUMRRO, a research organization in Alexandria, Virginia, believes he can put it all in a box with a color TV. So instead of your home computer screen merely showing *regular* interactive graphics (and two-dimensional pictures), you can travel through whole worlds—cities and canyons and planets and playgrounds—that look almost *real.* He says the terminal will cost $5000 in a couple of years.

All these different kinds of views will become important. And all will increasingly appear, and become familiar, in different panels of our control screens.

The Frontier: Clarity

Many people seem to think that bigger and better complications mean progress in computers.

They are totally wrong.

Beyond the Computer Screen

Anything you want to do with information can be done at a screen; soon it can probably be done better there.

For instance, if your screen is connected to a good text system and sufficient memory, you can certainly do better writing there than is possible with a typewriter. (Unfortunately, there are as yet few good text systems—but there will be more soon.)

Outside Control Diagrams

Yes, for handling information the computer screen is tops. But it has a more portentous capability still.

You will recall that computers can be hooked up to any other machine that can be controlled electronically. Thus a computer program can control a gas pump, a rotisserie, an oil well.

But in turn, *you,* at a computer screen, can direct the computer to take action in the outside world, making it turn on an eggbeater, or a drawbridge, or a stereo. By adjusting a picture to what you want.

A diagram that controls events—in the computer itself, or in the outside world—is a *control diagram.* If the diagram controls things outside the computer, it is an *outside control diagram.*

Control diagrams can be used, as we have seen, to control the operation of your computer itself. Whatever you want to do with a computer can ultimately be done most easily with control diagrams. But control diagrams are a powerful way to work with the outside world as well.

A practical application of outside control diagrams: there are now oil refineries where nobody goes around turning valves by hand any more, when the petroleum is supposed to take a new route.

Instead, an operator studies a map of the refinery on the screen. Selecting an area of the refinery where he wants to reset a valve, he touches that part of the screen with his lightpen; that area expands to fill the screen. He keeps expanding the map, and more details come into place, until he sees the valve he wants—the magnification is now sufficient to show it. With the lightpen he touches the valve's symbol, and a changing number shows the changing percent of flow.

Satisfied with that one, he changes a dozen more; all in less than a minute.

It's all going to be that way.

There will be setups run by control diagrams for editing movies, for running factories, for opening and shutting down public buildings, for lighting cities.

(You could probably drive your car with a lightpen on a control diagram—but your state Department of Transportation might not think it was safe.)

You should note one difficulty with controlling objects in the world by computer: *it's expensive.* The centralized hookup between the outside and the computer is the hard part, especially if it has to be reliable. The computer itself, and even the program for it, is negligible in cost by comparison.

Clarity and the Design of Objects

Let us briefly digress from the subject of computers, and talk in general about machines that are sold for human use.

Industry persists in turning out badly-thought-out objects that nobody can understand.

The technical things that consumers buy, like tape recorders, have always been badly designed. Designers have come out with a chaotic variety of confusing objects, differing widely. Most tape recorders are difficult to use, some ridiculously difficult. Yet tape recorders only do a few simple things; it's their bad design that makes them complicated.

Recent laws have made it mandatory for all contracts involving consumers to be written in simple English. What we need is a corresponding rule for the design of objects and systems for consumers. Just as the criterion for consumer contracts is that they must be readable by the average high school graduate, a corresponding rule for things sold to consumers ought to be that they have to be understandable in less than ten minutes of instruction. This ten-minute rule should be tatooed on everyone who designs consumer products.

Many engineers and technicians have claimed that this can't be done. Balderdash! It is merely difficult. Moreover, it takes intense dedication to clarity, and repeated revision and rethinking. You have to try over and over until a thing gets simple enough, just as you have to try over and over to make writing clear, and just as you have to rearrange over and over to edit a movie just right.

Another reason that technicians do not like the ten-minute rule is that it deemphasizes what they like to do, and minimizes their achievements in their favorite area of operations. Technical people like to think about technical things; that is why they are technical people. (One engineer has confided in me that he is never really happy unless he is *feeling those chips with his fingers.* This is a very poignant admission.) They think that designing a tape recorder, or a computer program for people to use, is a technical matter. It isn't.

Designing an object to be simple and clear takes at least twice as long as the usual way. It requires concentration at the outset on *how a clear and simple system would work,* followed by the steps required to make it come out that way—steps which are often much harder and more complex than the ordinary ones. It also requires relentless pursuit of that simplicity even when obstacles appear which would seem to stand in the way of that simplicity.

Much has to be reconsidered, of course, when it turns out that the simple-and-clear design is not feasible in its premeditated form; after many changes and reconsiderations, it is the brave designer who wins simplicity and clarity out of the tangle of different pressures.

This is not a column about tape recorders; suffice it to say that I have only seen one tape recorder I considered well designed. This was the Sony TC-50. It is no longer available. People think they want a lot of buttons. ▼