1946: THE FIRST COMPUTER

1971: THE FIRST COMPUTER ON A CHIP

January 1975: THE ALTAIR COMPUTER KIT

June 1975: THE FIRST COMPUTER STORE

1976: PERSONAL COMPUTING CONFERENCES

April 1977: COMMODORE 'PET' COMPUTER

Aug. 1977: COMPUTERS FROM HEATHKIT, RADIO SHACK

Sept. 1977: THERE ARE 50,000 HOME COMPUTERS, 50 BRANDS, 500 STORES

Oct. 1977: PERSONAL COMPUTING TAKES OVER THE NEW YORK COLISEUM

and it's

THE HOME COMPUTER REVOLUTION

by Ted Nelson

Suppose someone had invented a magical robot which could do automatic typewriting, memorize any information you wanted it to, and juggle that information later — extracting any features you wanted, telling you how many words it contained, or whether these included the word "meatloaf;" or figuring out the average of numbers that happened to be in it. Let's say this instrument is called a Retupmoc.

A Retupmoc could help you keep your appointments, plan your time, keep your checkbook up. In your leisure time you could play all kinds of games with it. And you could hook it up to play music, work your household appliances, and show movies.

Would you like one?

Believe it or not, you can have one now.

But it's not called a Retupmoc.

It's spelled the other way around: C-O-M-P-U-T-E-R.

But mention computers, which are just Retupmocs under their more usual name, and people grow uneasy and fearful and resentful, and somebody mutters, "They're taking over the world!"

And so they have, in the worst way. Computer systems for record-keeping and accounting and control, as used by business and government agencies, have given the world a new bleakness and oppression.

It does not have to be.

"Everyone knows" that a computer is an implacable, dictatorial, unapproachable, incredibly complicated device.

What everyone *should* know is that computers are fun, friendly, helpful, exciting and cheap.

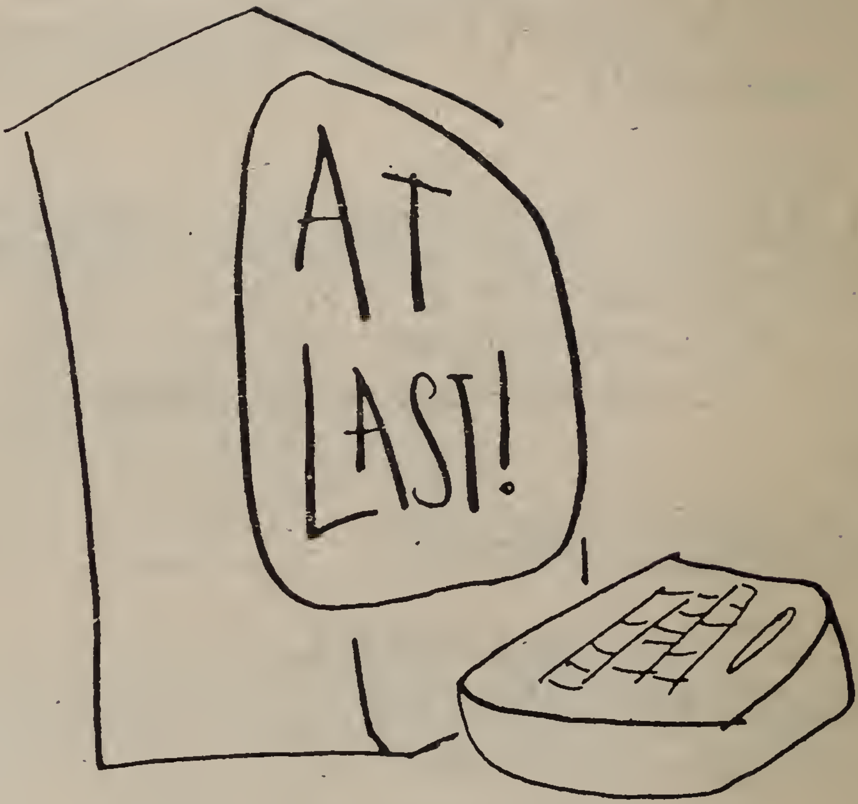
THE HOME COMPUTER REVOLUTION

Published by the author
Distributed by
The Distributors
702 South Michigan
South Bend, IN 46618

Additional copies: $2, twenty for $30, postpaid; check must accompany order. *Computer Lib*, by the same author, is $7 or ten for $50. Inquire as to wholesale discount.

My special thanks to Janie Hollinshead, for her help and encouragement on the first draft; to Debbie and Erik Nelson for their hospitality and grace under difficult circumstances; to Mark Miller for various forms of assistance; to Sheila McKenzie for her extraordinary generosity and effort in the closing phases of this project; to John Ouderkirk for the generous hospitality of his typesetting facilities; and to Linda Mallas and Nancy Tortorich for their great help in getting this finished.

Contents



Dedicated to those who saw it coming:

especially

Ivan Sutherland,

Doug Engelbart,

Calvin Mooers,

Claude Kagan,

Bob Albrecht,

and Ed Roberts;

and to Erik Nelson,

who has had to wait much too long.

HOME COMPUTERS!

The United States — indeed, the world — is about to be totally changed by a revolution few people have seen coming. The corporate world is unprepared for it. The public is unprepared for it. Governments are unprepared for it. And it will remake our world as drastically as the automobile, the telephone, or the atomic bomb.

It is the home computer revolution.

The computers we are talking about are not mere numerical calculators, nor do they have fixed, specific purposes. They are general-purpose computers, capable of being set up for any conceivable human purpose. They can do your taxes. They can play music. They can play games with you. They can help you run a business. They can help you write. They can help you keep track of things. They can turn on and off every appliance in your house, or show you a movie on your desk. Soon they will affect your work, your leisure, your eyes, your ears, your world.

In a world of rising prices, the price of computers is falling precipitously. Today, for three hundred dollars, you can get a computer that would have cost $50,000 or more ten years ago. The price of computers and their accessories will be less and less in the foreseeable future.

Already there is a thriving market in computers and accessories for home and amateur users, at prices from a few hundred to a few thousand dollars. Dozens of computers, hundreds of accessories are already available at these prices. Computer stores dot the land, and the hobby fanatics spend much of their time and their incomes in these dens of iniquity.

Some 50,000 home computers are estimated to be in place as of this writing. There will probably be ten million computers in homes by 1980, twenty million by 1982, perhaps a hundred million by 1987.

In a couple of years it will be no more unusúal to own a computer than to own an automobile. The home computer offers limitless possibilities and no prospect of market saturation. Home and personal uses of computers will dwarf the ordinary computer industry within ten years, perhaps five.

THE MYTH

Some very peculiar circumstances have governed the use of computers in the past two decades. Some very odd myths have shaped the public's mind on the subject of computers, and hang like storm clouds in the room when computers are mentioned.

Indeed, these myths enshroud the computer like clouds over King Kong's island, hiding it from the eyes of laymen.

All the separate myths fit together. The overmyth is both cohesive and coercive. To correct it, we have to peel back the misunderstandings and confusions like the petals of an artichoke.

One of these myths is the myth of the formidable, relentless and error-free computer: a mighty Moloch requiring the sacrifices and suffering of thousands of helpless serfs. This is an image which has been useful for some people.

Corresponding to this idea that computers are big and formidable, there naturally comes the idea that such things cannot possibly be used by individuals.

And then there is the myth of the computer priesthood, people whose work cannot be criticized, but only admired, for its incomprehensible hocus-pocus. They are beyond accountability, beyond comprehension, and not necessarily nice about it either.

So most people think that computers are unfriendly and unspeakably complicated; that they cost millions of dollars, require an army of technicians, and are unspeakably hard to learn to use.

All this is false. You came to have this idea for strange and complicated reasons, reasons involving oddities of our society and certain people that don't want the public to understand what they are doing.

In the past, employees (and clients) of large firms and government bureaus have been made to use computers. They have been ordered, or otherwise coerced, into using or adapting to computer systems that they were not consulted about and do not understand. If they complained, they have in general been told that the system they have to put up with is "the way the computer does it."

Some people would call this a useful explanation for the simpleminded. But it could as easily be called a damned lie.

In a previous book, *Computer Lib*, I discussed the myths of computers and what has kept them going. These myths are now, in the next couple of years, going to be exploded. And a lot of people, and companies, are going to have egg on their faces. (See "The End of the Myth," an appendix to this book.)

ANOTHER WORLD

There is an everyday world we all know. It involves houses and trees, children and dogs, cars and jobs and movies and taxes.

There is another world, unseen and unseeable: the world of information and its automatic handling and treatment — a new world that has accidentally been called that of "computers."

Millions of people deal daily with computers. Hundreds of thousands of people are currently employed as computer programmers. They work with questions and issues and problems that are reshaping that everyday world, silently, irrevocably.

To be able to share this world of information and its new gifts to your life, and know its impact on the regular world, you have to understand what computers are about. And if you are going to have a sensible opinion about tomorrow's world problems, which will be grave, you have to understand. Because tomorrow's world and political problems will concern systems and interconnections and compatibilities as much as they will involve resources and hunger and bombs.

It is very important for you to know about these things right away. The country and the world are changing fast. Records are being kept that were never kept before, making possible repression and dictatorship on a scale the world has never known; and yet today's little computers also make possible a greater freedom for individuals than has ever been known before, and wider possibilities of individual learning and understanding.

There are, too, bright new possibilities for our lives: the potentialities of interactive computer systems in aiding all forms of human activity — not just the technical stuff, not just the dry transaction. It may be that new forms of reading and writing, using computer screens, will make possible a new birth of intellectual freedom.

... OPENING UP

The press is largely mystified by these goings-on. But more and more, as 1977 wears on, radio and TV and magazines are doing pieces on personal computing. *The Wall Street Journal, The New York Times,* TIME, all have had their obligatory blurbs on how the home computing fad is growing. All the popular technical magazines for October have cover pieces on home computing. The weekend of October 28 through 30, with large Personal Computer conferences in both New York and Chicago, will thrust home computing into the public eye of the nation. From now on, home computing will be all over magazines and television — that is, until the phenomenon becomes a part of our everyday life. That means it will taper off in two to five years.

Unfortunately, the way you are going to be hearing about computers in the home will make it all sound like a gee-whiz kind of surprise. The writers will begin by saying how unlikely and startling it is, when it is only a strange fact of our culture that this has not happened until now.

There will be many shocks to the society. There will be total confusion in *both* the ordinary world *and* regular computer world as the little machines move in. The regular computer companies are many of them trying to sell, at huge prices, systems the equal of today's cheapie desktop machine. They will soon be in trouble. Even mighty IBM may collapse. Computer schools will have to face up to the total irrelevance of most of the things they've been doing. The little computer will also have a strong effect upon the ordinary schools. By bringing materials to kids that the schools would have held back to maintain the grade system, they may drastically alter the basis of education in our society.

Many jobs will be made foolish, that people have been carrying out faithfully and with great attention. But so many jobs are merely information-transfer jobs — made hard by being unsystematized, thus requiring judgment. Clerks, librarians and many others may find themselves in very awkward positions.

A BETTER DAY

The first computer age brought us the oppression and bureaucratic abrasion of unpleasant requirements and delays, bad forms and categories, incomprehensible directions, electronic excuses.

The second computer age will be the opposite. We can undo the oppressions of the first: eliminating paperwork, eliminating the bureaucratic functionaries who specialized in insulting you for not understanding the incomprehensible, and making things available, comprehensible, open.

But the tradition will die hard, though, and so it is up to the public to know what to demand.

A CONSUMER EXPLOSION

The little computer, costing from five hundred to five thousand dollars, will be the most explosive consumer product in human history, selling more units in less time, and having a more revoluntionary effect, than any other object ever sold.

Packaged computers are here from Commodore, Radio Shack, Bally and Sears; and others are probably forthcoming from such mass manufacturers as RCA, Fairchild, Atari, Texas Instruments and Timex. These portend a future in which small personal computers will figure in our lives by the million.

The question then is, how many million by when?

This book predicts ten million personal computers by 1980.

WHAT WE'VE NEEDED ALL ALONG

But there's more to it than that.

The essential thing to understand is not that something new has happened, to which the public is responding in astonishing ways, but rather than the public is responding to the late development of products that have been effectively suppressed until now. Computers would by now have had far more impact on the general public if various things had gone better This will be redressed; and their lowering price assures that they will become available for any conceivable human purpose.

When asked why the sudden demand for small computers, many computer people say that it is due to the new computer on a chip, or "microcomputer." This is false. The computers of the late 1960s, small and rugged, could have been used by individuals, professionals, businessmen throughout the country — and many of these people could have afforded them. (And American business could have been simplified by the computer, rather than entangled and complicated, long before now.) Thus the sudden demand for personal computers is actually the unleashed demand that has been previously held in check; and it is still reaching toward the levels it would have achieved in the 60s and early 70s had it not been for the strange monopoly structure of the computer market, and its domination by a certain style of system upheld by a distributed priesthood.

PRECEDENTS

It is not as though the unexpected had never caught on before. For each dramatically successful consumer product the number of people who knew it would catch on were always in the minority.

They laughed at the Emperor of Brazil when he brought the first telephone ever sold. Nobody imagined a worldwide net of wires, conversations, whispers or heavy breathing.

Everybody at the turn of the century could understand the *use* of the automobile, but its future could easily be doubted. After all, there were no paved roads to speak of.

Even Edison didn't favor building a projector when he invented movies a hundred years ago. The novelty would wear off too fast, he thought.

The first phonograph records broke too easily, because of their cylindrical shape.

The first Kodak camera had to be mailed back to Rochester for reloading.

Radio was far too much trouble at first. You had to poke around a little germanium button with the end of a wire until you heard something through the big tin-can headphones. (And two people couldn't listen at once.)

The first television kits came out in 1929. But it was uncomfortable to stare at a little orange square through holes in a spinning disk, and the disk motors were hard to synchronize. Two people could watch at once only cheek-to-cheek, with one eye closed.

The first tape recorders were large, but the big problem was the weight of the reels of tape — a thousand pounds for half an hour seemed a bit much.

And computers, everyone knows, are big and hard to use.

THE MOTIVATION

We 20th-century folk — Americans especially, perhaps, but all us coevals — are gadget-hungry. The inventions we have listed caught on because they involved, in varying degrees, privilege, style, leadership, power, freedom, convenience, and leisure entertainment.

The home computer will combine all of these.

ADVERTISED USES

Many different ideas are afloat for what to do with the home computer. If those invented earlier sounded fanciful, many are already being advertised by manufacturers.

The first Heathkit computer brochure, for instance, announces programs for Blackjack, Biorhythm, and Star Trek — as well as two miscellaneous game sets.

Here is word from a firm called Polymorphic: "Your entire family can use the Poly 88 to help systemize (sic) your household routines: store your appointment calendar, analyze your home energy use, set up a diet program, balance your budget. It introduces children to the world of computers — it can play games with them or help them with their schoolwork." And again: "It can keep track of your receivables, project future sales, evaluate investment opportunities, or collect data in the laboratory . . . Use the System 8813 to develop reports, analyze and store lists and schedules, or to teach others about computers. It is easily used by novices and experts alike."

Another firm tells us: "Sure, our computers play games — but they don't stop there. We can show you systems to do your paper work, guard your house, plan your menus, and help you shed a few pounds. With our help, you can predict your biorhythms, your taxes, your business future." That's the word from 2005AD, Inc. (formerly Mom's Wholesale Audio).

These are only a few applications already being advertised. Others include composing and performing music, drilling your child in arithmetic, and keeping track of your possessions.

NO LIMIT TO THE APPLICATIONS

But simply to list these uses does not really tell you how the computer works, and until you understand that, you probably won't be able to figure out how all this relates to you.

For the computer is limited only by your imagination. Once you understand how wide are the options the computer allows, you can make up your own computer applications, out of your own interests.

The surprise and magic is that the computer has no nature at all. Intrinsically it does nothing; somebody decides what it is to do and what are to be the detailed steps for it to follow when it does it. (This is discussed more later.)

When the basic idea hits you full force, it can be quite an impact.

THE THRILL OF COMPUTERS

It may happen the first time you see graphic interaction on a computer screen: a picture draws itself, and the user makes a choice, and then zip! a new picture appears. To see such a system is to recognize, perhaps suddenly, what a computer can do. Or the realization can come from seeing a computer whose printer is flinging out page after page of writing, the pages of a book or successive letters to the customers of a company. The power, the speed, the awesome controlled variation, give one a sense of mighty forces that can be harnessed to human purposes.

Or it can come from playing at the screen with an interactive game and suddenly realizing that some feature of the game could easily be different. When you come to see that the program is only somebody's list of things to do, the computer becomes in your eyes not some independently threatening mechanical autocrat, but a gentle giant, trustingly obedient, with no judgment whatsoever.

All of these visions are correct, and one of them may have the emotional pull that grabs *you* — the emotion required to make you care about anything.

These are many ways that computers can become thrilling to you, depending on your personal obsessions and your style — or love of gimcrackery. Most computer people, at least the good ones, were once thrilled by computers; many can still remember it; some have not lost it.

The computer is a machine that brings out the kid in all of us.

Us kids get turned on to computers by the excitement of being a part of what's happening.

Us kids get turned on to computers by the sense of mastery, the same excitement of control that comes with learning to drive a car or swim. But the mastery when you control a computer is not in the feel of the splashy water, or the open road, but in a realm of dynamic abstraction, action abstraction, which is unlike anything in the natural or the everyday world.

Us kids get turned on to computers because we love to make things happen, and to have gadgets play out our ideas. Us kids get turned on to computers by succeeding at the challenge of making an intractible dumb object into a docile servant — rather like taming a horse.

And finally, us kids get turned on to computers by the excitement of controlling expensive and sophisticated equipment. (But this challenge will disappear, since computers will not be expensive any longer, nor will we consider them sophisticated, any more than a washing machine.)

When you realize what computers really are, and that they are really not what you thought they were, the experience energizes your imagination and electrifies your thought. Wow! you think. With a computer I could keep track of my record collection and memorize Latin names of flowers and create animated cartoons and send all those letters I should send and figure out the key to the universe!

At least that's how it feels. The liberating excitement at realizing that you now can have a super-robot, a magic librarian, a mental butler, is an experience that may change your life.

You may be half delirious for weeks or months. Your friends and family may be disturbed by your behavior. Your social life may suffer. You can stew in the day about how to set up one thing, and lie sleepless in the night contemplating the arrangement of another. You can become, to all intents and purposes, insane.

In time you will come out of it; in appearance, perhaps, unchanged. You will continue to work, play, eat and sleep. But there will be an important change. For now the computer will be an active part of your thinking life. That is what it means to become a computer person.

THE MAGIC OF THE INTERACTIVE COMPUTER

Without any warning — before you have time to get scared — let's sit you down at a computer terminal.

What's that?

A computer terminal is anything that lets you talk to a computer. Not out loud — that's not convenient but through a typewriter of some sort.

You sit at a keyboard, and type. The computer receives the electrical signal from each key you touch. When it is good and ready *it* types something back. It types on paper, or it types on a screen. Both of those can be built into terminals.

"What will I do?" you ask.

You'll see. We are going to show you magic.

AN INCREDIBLY, RIDICULOUSLY SIMPLE EXAMPLE

"We are going to play a game," I say. I turn the computer on.

The computer types:

I AM THINKING OF A NUMBER. TRY TO GUESS IT.

"What do I do now?" you ask fretfully.

"Guess a number," I say.

"Forty-three," you say.

"Don't tell *me*," I say. "Tell the computer."

"But how?" you ask.

Raising a jaded eyebrown, I gesture with my pipe. "The keyboard."

"You mean . . ."

"Just type it," I say, stifling a yawn.

4, you type painstakingly, 3.

"Now what?" you ask.

"It's the computer's turn, but it doesn't know that. You have to signal that your number is complete."

"How do I do that?"

"This button here," I say, "It's called the carriage return."

"Gee, this is exciting," you say. You hit the carriage return. The computer types instantly:

YOUR NUMBER IS TOO SMALL. TRY AGAIN.

You practically fly out of your chair. "My *God*, how did it do that?"

"Never mind," I say; "that comes later in the book. Why don't you try again?"

You type, 43333. Carriage return.

Instant reply:

YOUR NUMBER IS TOO LARGE. TRY AGAIN.

"Gee, this is exciting," you say again.

"Before we go on," I say, "do you find this program hard to use?"

"Hard to use?" you reply in amazement. "Why, it's easy as pie!"

Indeed so.

This is called interactive computing.

This is a very simple program. It is simple for you, the user, to understand, and it is simple for almost anyone to program. In this example, no training is required, whatever. No skills are required, just for you to type a number. If you make an error, you simply backspace, which obliterates the characters you have typed.

Then you type the corrected version.

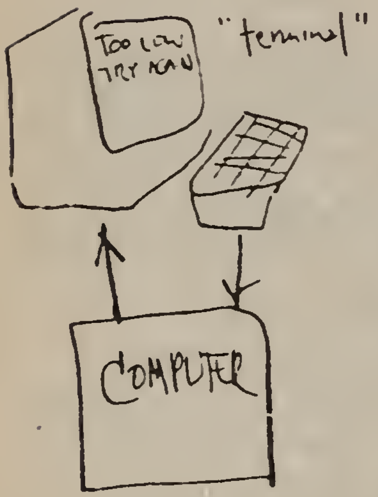
This is the basic idea behind interactive computer systems in general: that you, the user, can without difficulty or mishap employ the computer for some chosen purpose of your own — in this simple example, playing a game.

We will draw the curtains on your first experience at the keyboard. But you can imagine how it ends. Your guesses draw closer and closer to the secret number until it types, CONGRATULATIONS. YOU GUESSED THE NUMBER IN 14 TRIES.

Except the number of tries depends, of course, on how many it *did* take.

IS IT A "TERMINAL" OR IS IT A COMPUTER?

THE DISTINCTION IS LESS AND LESS CLEAR.



In the old days, the "terminal" just sent characters between a person and a computer.

These days, it costs almost nothing to include the computer itself inside the apparatus of keyboard and screen.

And maybe that's how COMPUTERS SHOULD HAVE BEEN MADE

ALL ALONG.

ANOTHER EXAMPLE THAT SHOULDN'T FRIGHTEN YOU EITHER

You are an executive secretary. "Here are some names and addresses," says your boss. "Kindly put them into the computer."

Yikes, you think.

"There's nothing to it," he explains. "Watch."

He turns on a little box on the table. It, too, has a keyboard. Also a big TV screen.

Your boss puts an ordinary looking cassette into an ordinary-looking tape recorder and hits a button.

The screen says:

NEXT NAME?

Your boss looks at you encouragingly. You look at the list. You type ABERNATHY, CLAUDE Q. (Carriage return.)

The screen says:

ADDRESS?

You type: 643 MAIN STREET (carriage return) FERNWOOD, OHIO (carriage return).

The machine says:

NEXT NAME?

"Why, this is *easy*!" you tell your boss.

He nods, smiling.

IT'S THAT EASY?

In both these examples, the game playing and the secretarial work, no training is required. The task and the game are simple. Now perhaps this is not your idea of intellectual challenge, but at the same time, you didn't break the computer, either.

Interactive systems can do much more. In the coming years they will help us with every conceivable human task — whether painting a picture, composing a sonnet, or trying to decide how to invest your money.

THE MYTH OF COMPUTERS BEING HARD TO USE

Many people seem to think that anyone who sits down to use a computer must be highly trained. "Oh dear, I'm afraid I'll break it," they say as you sit them down.

This has been true in the past. Computer systems were very easy to screw up, and most still are.

Before now, most computer systems have not been set up with ordinary people's use in mind. A certain class of experienced user was anticipated and so only these people used the system. Something like trappers and explorers of old, who knew how to approach one another and ovserve a certain etiquette of the forest. In old-fashioned computer systems, you had to know a lot, and take actions slowly and deliberately at your keyboard. You had to think carefully before each act, and experiment systematically within very careful limits, testing each new concept in detail before going on.

But that's about to change. Interactive systems will start appearing on little computers for every purpose. And we are now going to see a new kind of user: slam bang, sloppy, impatient, and unwilling to wait for detailed instructions. Thus the systems must be set up to enable such people to blunder and muddle through, learning as they go, reversing those actions which they took by mistake. The new systems will be easy, and fun and powerful, and useful.

WHAT YOU CAN DO NOW

Today you can walk into a computer store in any region of the country — that's right, a *computer store* — and purchase an electronic digital computer capable of running programs with thousands of separate steps.

You can use it for cartooning, music making, record keeping, writing letters, and even arithmetic.

SCREENS

With various accessories, you and your computer can create animated cartoons and pictorial games on the screen — and their complexity and action are limited only by your own imagination, and the free time you can devote to their creation.

WRITING: THE ELECTRONIC TYPEWRITER

You can hook your computer up to your typewriter and have it send out incessant letters to all your friends, changing or adding paragraphs as you like, with the computer doing all the typing automatically. You compose the letters, of course.

MAGNETIC COMMUNIQUES

You can even skip the typewriter hookup and send letters to your computer hobbyist friends — but with letters stored on cassettes. When your friend plunks the cassette in his own system, he will read your letter — or watch the animated cartoon you have sent him.

TELEPHONES

But why bother with the mail? The telephones, too, can be hooked up, allowing you to dial out automatically by computer; and even to add music and sound effects to your telephone conversations.

With a special accessory, when the program demands it, your computer can actually make an outgoing telephone call by itself to any telephone number you designate. It can automatically detect when the line is busy, and keep trying the call until it gets through.

With this same special accessory, your computer can receive telephone calls automatically. So send your friend his letter directly, instantly, over the phone — a long, and instantaneous, telegram. (And in far less time than you'd take to say it all on the phone by word of mouth.)

You can also set up your computer at home to receive calls. While you are on the road, you can telephone back to your computer and send a written message which it will store away. This could be a salesman's orders, a reporter's story, or a letter to your aunt. Having completed this transaction, your receiving computer at home can turn around and telephone outward to someone else, all by itself, relaying whatever messages you want passed on to your friends, your family, your office, or other computer hobbyists.

INFORMATION STOREHOUSE

Using various types of recording device, including the cheapo cassette recorder, you can inexpensively store large quantities of information for your own future automatic use. For instance, you might want to store your checkbook record for balancing and analysis. This means you can run the computer through your own record of checks to find out how much you've spent on business-related activities, how much for entertainment, and so on.

GENERAL AUDIO CONTROL

Naturally, hooking your computer to the volume controls and tuner and record-playing controls of your hi-fi will allow you to program any part of the system as a clock-radio, to provide musical interludes, to provide background music for dinner and dancing music after a specific television show. All these arrangements and more could be listed in a schedule stored in your home computer for weeks and months ahead.

Thus your program can turn on particular stations, making them louder and softer at given times, or switch among speakers, depending on where you want the sound. Or using some kind of a master control hooked to your computer, such as a joystick, you can set the computer up to adjust sound levels all over the house by tilting the joystick in the direction you want it to be louder.

HOUSEHOLD CONTROL

Because the computer can be used to control many different electrical and electronic objects, it is naturally possible that your computer could be profitably hooked up to your toaster, your high fidelity, your television, your clock; to your air conditioner, your furnace, your refrigerator, and your cuisinart.

SAVING HEAT

You can use it to monitor temperatures inside the house and control the furnace for the least use of fuel. Thus the furnace can automatically lower itself at night, which is good for sleep. By means of a number of electronic thermometers stationed round the house, and an hourly schedule of family doings, your furnace system can selectively raise or abandon particular rooms of their house, according to their individual use.

Such a setup might offer the user considerable savings in energy, particularly in buildings designed for the saving of fuel. However, older buildings might benefit as well. This is a straightforward and practical use of the computer.

(A resident of New Hampshire uses his digital computer to control his *wood stove*, and reports 10 to 30 percent improvement. (*Mechanix Illustrated*, Oct 77, 142-3.))

PERSONAL PLANNING

Using a computer, you can keep track of your daily and weekly schedules, and have a list of things you intend to do. Not only is it always up to date, but it never gets cluttered with crossed-out entries.

Using a computer, you can type a tentative menu plan for a forthcoming week and have the computer tell you how much time you will have to spend in the kitchen each day, what the calorie or cholesterol count of each dish is, and what the costs will be.

Similarly, for financial planning there will be nothing like it. Or for space planning, especially when large households move.

SPEECH

Using a special accessory, you can hook up your computer to respond to your own spoken word.

Such speech input hookups are able to recognize some thirty-two words at a time, but you can program the computer to switch vocabulary to another 32 words instantaneously. You can, indeed, have as many vocabularies as you like — as long as at any one instant the computer need distinguish only among 32 possibilities.

You can have the computer talk back to you, through a special output accessory. Hooked up to this accessory, the computer can speak in recognizable English. (Its voice is understandable, if not lovely.)

MUSIC SYNTHESIS

You can hook it to music synthesizers, and have your own small orchestra of several voices play counterpoint — sounding either like real musical instruments, or sounding like nothing that has ever been heard before. Using a special accessory — a piano keyboard — it is possible for you to make of your home computer a musical instrument as versatile as a $5000 music synthesizer — and some in some ways better — for much less.

...IF YOU YOU HAVE THE HARDWARE

It's not necessarily cheap, but you can actually do all these things with accessories presently on the market.

...IF YOU HAVE THE PROGRAMS

Besides the hardware, you have to have programs tht go in the computer and make those gadgets do their stuff, and see to the interior juggling of information. (For an inkling of what this means, see section on Programming.)

ANYTHING, MOSTLY

These are some examples of what computers — computers actually now available for the home — can do.

There is virtually nothing which has *not* been proposed as a use for the home computer, but we will see what catches on. It's hard to tell now what applications will be found feasible and practical.

None of these ideas are good or bad; they depend on personal taste. Nothing is inappropriate for a computer; no appliction of computers is bad — unless it is either morally bad, or badly done.

YOUR OWN STYLE

The point of your own general machine is controlling all the others, or having it do your thing, in your own personal way.

You don't have to use a program unless it 's really your kind of program. There are lots of different ways to do each thing.

People will differ considerably in the styles of systematization they will adapt to. One man's easy system is another's regimentation. The other may devise a system that the first thinks is chaotic. To each his, or her, own.

Drawbacks . . .

1. FOR SOME OF THESE THINGS YOU'D BETTER KNOW ELECTRONICS

What we've described can indeed be done with available equipment. It is indeed possible to put all the electrical appliances and controls of your household under the control of your computer. But the connections must still be specially built, and there is no repairman you can call if such a thing fails to work. For the complicated hookups, then, you will need to know electronics. The reliability of such systems is something you would probably find important; for now, it is not as safe to do these things with a computer as without.

2. FRANKLY, IT COSTS A LOT

As of this writing, it costs at least $600 to get into home computing. (This is the cost of a Radio Shack setup, without the cassette recorder.) And to begin to do anything serious on it, like handle your business records or create programs to save, you need some form of information storage. This can be done on the cheap with cassette recorders, but there are far, far better ways — especially disk memories, which begin in price at about $500 and go up into the tens of thousands.

Similarly, a setup allowing you to make a real variety of electronic additions will cost around a thousand dollars for openers, and it's upward bound from there. A good complete sort of system with disk memory now tends to cost from $2500 to $5000, assembled. (And these disk prices may not go down quite as fast as the price of the computers themselves.)

STUFFINESS

It seems that many of the proposed applications for home computers seem a little on the stuffy side.

Menu planning is an example. Some people plan menus and some don't. Surely there is nothing wrong with planning menus way ahead if it's the way you operate, but it's not a style of operation you can necessarily push on your fellow users.

It is important to point out that this may not suit some peqle's style of planning at all. A particular computer system and program may not do what you want it to, in a style suited to your life.

UNNATURAL?

Many people feel, correctly, that life has gotten very far from any natural conditions; that we need to get closer to each other, and to the earth.

And based on these reasonable ides, they also feel that computers can only be an additional complication in life, a further estrangement from the good and true.

This is not so.

Tomorrow's easy computers will help us get to the heart of what we do, rather than diddle with false complictions.

The computer screen, about which you will read more later, offers unprecedented promise for the betterment of our lives.

Call it a paradox if you will, but this step that now awaits us — using the computer screen as a magic slate to keep track of things, get our lives under control, and bring us knowledge — will be a simplifying, clarifying step like none before.

And because it will make understandable what has been tangled, it will make our lives "natural" as they have not been for decades. Or perhaps centuries.

The little computer will make our lives *less* complicated.

HOW WE GOT HERE

In 1946, just after World War II, the first electronic computer — the ENLAC — was turned on in Philadelphia, and the legend has it that for a moment the lights in half of Philadelphia went dim.

The people who developed these machines had their minds fixed on only one purpose, the working through of very large arithmetic problems. But it happens that the instrument they created, the electronic computer, has uses far more sweeping; the search for faster numerical operations accidentally produced a totally novel and fundamental device, with no particular purpose at all. But it took thirty years before the computer was really understood, and society is just about to act upon this understanding.

The remarkable thing is not that computers are useful for people as individuals. The remarkable thing is that it has taken us so long to discover that computers are useful for individuals, a fact that should have been obvious from the outset.

The purpose of this chapter, then, will be to focus on how the misunderstanding grew and was perpetuated.

This is the obligatory "history of computers" chapter. Ordinarily such a chapter tells you how more and more uses were found for the computer, until — lo and behold! We arrive at the present state of enlightenment, whatever that may be as of the date of publication.

In some of the latest books and articles, the history section tells you that the development of personal computers is a wonderful surprise, based on the fact that circuits have gotten smaller and smaller.

Rather the contrary. Personal computing is a postponed development, hindered till now by the peculiarities of the computing industry. Our accumulated need for personal computers is enormous; there would have been a market, though not a great one, for them long ago, at any price. At the prices of the late sixties it would have been considerable. This pent-up demand will produce, right now, in 1977 and 1978, the most explosive growth of any consumer product in world history. The question is not, why have home computers come? The question is, what has held them back? This chapter hints at some of the answers.

Eckert and Mauchly, who invented the computer, first took their invention to the International Business Machines Corporation, but they were dismissed by its president, who thought that the calculating machines they already had were fast enough.

At one time Eckert and Mauchly did find a backer who made racetrack signs. Unfortunately he was killed in an airplane crash. It is interesting to speculate on this quirk of fate: the first use of computers might have been entirely different, and everybody might have understood them from the start as presentational machines, had it not been for this misfortune.

But they found backers at Remington Rand, a large office equipment conglomerate. Remington Rand introduced its computer under the trade name UNIVAC.

The early computers, like most that were to come, took far longer to develop that their designers anticipated, and a new problem had to be faced whose meaning and proportions became clear only gradually: the problem of "programming" — the art of deciding what things the computer should do and in what order.

By the early 1950's the electronic computer had become adapted as a beast of burden, and was doing bookkeeping work for many large corporations. And computers spread throughout the business world doing billing, ordering, inventory and other record-keeping, through systems that all involved punch cards. By the end of the fifties they were everywhere.

But computerization didn't happen nicely. A lot of people suffered. A lot of people lost their jobs. A lot of people had to adapt to terrible systems that were forced on them unnecessarily.

The Univac had not been alone for long. RCA, not to be outdone, created a giant computer called the BIZMAC. Burroughs, General Electric, Philco and offers began to compete in this strange new field.

IBM came back in, of course. And as the world knows, IBM was the winner.

IBM's most fundamental product was originally the punch card. The computer tied in with it well. Computers could use up punch cards faster than any of their previous machines. Quite naturally, the punch card became a cornerstone of IBM's computer systems.

While its leadership has in technical matters been only moderate, IBM has always been first and foremost a sales organization. Its autocratic and pompous first president, Thomas J. Watson, created a unique style of hard-driving sanctimoniousness. Its suave and energetic personnel, from the keen-eyed salesman to the self-confident and well-dressed repairman — called a customer engineer — have always had a great appeal to the hearts and minds of businessmen. In addition, the determination and resources of its agressive sales force have created a self-fulfilling and self-perpetuating myth of invincibility. Like the United States Marines or the Roman Legion, the very prospect of their onslaught demoralizes the opposition and hypnotizes innocent populations into submitting.

THE IBM RELIGION

Many business people do not merely respect IBM. They *believe in* IBM. This leads one to suspect that perhaps IBM should not be seen as a company at all. It is much more like a religion. IBM employees and customers, to an astonishing extent, believe in the manifest destiny of one company to control the computer field. The competition is beneath contempt: "*We're* in it for the *long* haul," you hear them say. Meaning that in return for IBM's stewardship of this blessed invention, the computer, in what they think is its truest and most divine manifestation, no reward is too great.

The mystical relation between IBM and the upper corporate management of its customers also deserves examination. The seeming efficiency and strength speak to something emotionally deeper then mere profit and loss. And it is corporate management, not technical people, who choose big computers.

THE EFFECT OF THEIR PRODUCTS

In the 1950s, IBM defined in its product line what the socially accepted view of a computer generally was to be. But not quite realizing it.

Computers can be small or big. They can come simple or with a lot of encumbrances. They can be convenient for one user at a time, or set up to be run by a bureaucracy that stands in the way of the user.

As IBM increasingly sold computers, IBM increasingly defined the way computers were made by their competitors as well. Other computer companies came to base their products in part on the design of IBM's computers. The computers they created had the same thinking behind them: some computers for business, some computers for science. Furthermore, it was important, apparently, to provide products resembling those of IBM. In this way the customer could recognize the relation between the competitor's products and IBM's products.

Against IBM's heavy salesmanship, few companies could stand up. Philco, General Electric, Westinghouse and others gave up in the sixties. (RCA and Xerox were to follow.) But in the early nineteen-sixties, there were several firms that genuinely challenged IBM.

THE INNOVATORS

Actually three companies flouted IBM's definition of the computer, making really significant innovations.

1. BURROUGHS

The first was Burroughs, a venerable old accounting machine company. Unaccountably, Burroughs listened to the people who programmed the computers in deciding what kind of computers they would make and sell.

Spearheaded by Bob Barton, The Burroughs Corporation created the prog rammer's dream computer, the Burroughs 5000. The machine Burroughs announced in the early sixties was spectacular. It was built for the convenience of programmers, with a special eye for the Algol language (see chapter on languages), but Fortran and Cobol were also made available.

Most importantly, its master control program, MCP, was an elegant and simple system for shepherding the other programs in and out.

It was the best big computer made to that date.

2. DIGITAL EQUIPMENT CORPORATION, or DIGITAL, or DEC

In 1960, a group of engineers around MIT, in business selling computer circuits, found they had enough extra computer circuits left over to build a computer. It was put together in the garage of Kenneth Olsen. They decided to sell it, but did not think they should call it a computer: even then, "everyone knew" that computers cost millions of dollars. So Olsen and his associates called it the Programmed Data Processor 1, and dubbed it a "minicomputer." In the next three years they brought out the PDP 3, 4, and 5.

This PDP-5 was quite an innovation. It was only the size of a closet. It cost under $30,000.

Then came a big PDP: a time-sharing machine designed with loving care by programmers, for programmers.

Olsen and his paternalistic little company, housed in a listing old 19th Century factory in the boonies of Boston, had chosen the only sales strategy which could provide them a unique toe-hold against IBM. It also showed an immense respect for the customers' technical employees. The computers they made were designed quite precisely to suit the needs and convenience of programmers, engineers and scientists. They built, in other words, machines directly suited to the research requirements and industrial needs of the technical elite.

Digital's computers could be easily hooked up to anything, taken apart, rearranged. Digital went out and proved that they could do without IBM-type salesmanship by proving that they could do with no salesmanship. Not only did Digital products sell themselves, they had to.

3. CONTROL DATA

A colorful individual named Seymour Cray, working at home in Minnesota, and with a lot of capital behind him, created the fastest and biggest computer in the world. It was brought out by Control Data Corporation.

It was a "scientific" computer, meaning it didn't bother with decimal arithmetic.

It cost millions of dollars. It was liquid-cooled.

It was ready in 1964.

THE CRUSHER

At the beginning of the sixties, IBM had everything sewed up. They made very good computers, and offered extremely good service. Their 7090 was the standard "scientific" computer, their 1401 family was the standard "business" computer.

(Sophisticated users always questioned this division: the "scientific" computers were merely *better,* and the "business" computers were kind of silly, with their decimal arithmetic.)

But in 1964 IBM did an abrupt turnaround. They announced a line of computers for "all" purposes. This was the System 360.

The 360 resembled Digital's big computer in its internal structure. It resembled Burroughs' big computer in that everything had to be done through a master control program, shepherd to the other programs.

And it did not resemble the great big Control Data machine at all. IBM said that nobody would need a computer like that.

IBM had locked out the competition.

It is fairly clear now that the System 360 was to lock out other manufacturers. By convincing a firm's top brass that no other computer was necessary, it prevented other machines from getting a toehold, or more than a nominal share of the computer market — even if they were superior, like the machines we have mentioned.

THE RESULTS

Of the innovators, Digital prospered most. They could be locked out of the front office, but not out of the lab.

The basic mistake that Burroughs had made, little Digital had avoided. While Burroughs had dramatically improved the technical basis of computing, it had tried to sell the resulting system for business purposes, and so could not stand a chance against the psychological and conceptual warfare of IBM and the IBM religion. While their products' superiority from the programmer's point of view was widely known, the occupational hazards of trying to recommend Burroughs' machines to one's employers, or the difficulty of doing so even when it seemed safe, tended to make programmers' recommendations ineffectual. It was not enough for the computer people to believe it was the best business computer. They could not reach the businessmen who chose.

But the market chosen by Digital was not an area where IBM salesmanship, merely as a political skill, could penetrate. Digital products were rugged and extremely well thought out, and were built by technical people for technical people — for engineering and laboratory use, with the convenience and needs of the technical user in mind. They were good for programming, but further down than Burroughs, at the very bottom level.

This was an area — technical appropriateness — in which the president and vice-president and comptroller do not interfere with the purchasing decision. And so laboratory people and engineers were free to buy what they preferred.

It is now fairly clear that the 360 was a trick to lock out other big computers.

What has not gotten attention, however, is the effect the 360 had in postponing small computers and interactive computers.

The inconvenience and difficulty of the 360 is truly astonishing. Where the Burroughs 5000 had offered a control program of great simplicity and clarity, somehow the control program of the IBM 360 was tangled and complex to an almost unbelievable degree.

Yet we cannot help but note how this helped IBM. This complexity placed in every customer firm a group of programmers trained by IBM and loyal to IBM, comparatively unaware of the compeition — a fact that effectively rendered a firm's connection to IBM permanent, almost like a Trojan horse within the company.

And the computer departments, in their own self-interest — out of jealousy if not fear — were motivated to prevent the purchase of small computers elsewhere in the firm.

Now, if computers were just a form of mighty industrial machine, as some think, and some would have you think, then a monopoly on computers is of no more interest than a monopoly on crucibles or drop forges — just a form of business blood sport.

There are three schools of thought about the IBM 360 line and its successor of today, the 370. The faithful believe it to be the finest computer available, the state of the art, and are often blissfully unaware of superior features of competitive machines and systems. Some say they are indifferent, "it's just a big computer." No political hot water that way. But others, often very sophisticated computer people, consider the 360, with its attendant complications and Mickey Mouse, to be an abomination of the vilest kind, an insult to the needs of people who care about computers, a colossal waste of time.

But one thing is sure. With this computer line, IBM made certain that the only people who would program its systems would be those willing to fight their way through the tangle. A measure of determination — of obeisance, of devotion — a pound of flesh — had to be paid in full. It wasn't like Dartmouth, where they built the computer system to be wide open to the rankest beginner. It wasn't like the Burroughs systems or the PDPs. It wasn't like the software that comes with hobby computers, intended to get the ten-year-olds whooping it up in no time. No.

It is hard to see the 360 as anything other than a system to be run only by the faithful, where those with casual interest are unwelcome. You must pay your dues to use the 360; learning its mumbo-jumbo "JCL" language is getting your union card. After which you, as an insider, may also come to feel that nobody else should program unless they, too, have to go through the same initiation.

But what if computers are really important to human mental life?

What if they are something human life and culture and education have really needed? What would we say about someone who had kept them out of people's hands?

These are emotional issues. They will become more emotional. The more people come to love their little interactive computers, the more they will demand to know why they took so long to get here.

But wait.

It is easy to get paranoid if you care too much.

IBM has responded to its continuously shifting business environment like the "rational monopolist" we hear about in economics courses. Whatever effect their technology and marketing have had on our culture has probably not been intentional, except that the image they promoted for computers helped them sell more computers.

IBM is in many respects an elightened and charitable firm, anxious to take the initiative in equal opportunity programs, a sponsor of cultural events.

If the corporate vision of IBM can be turned to the recognition of what personal computing can become, should become, and will become, the human and intellectual resources they can turn to the improvement of the world are without parallel.

Let us all hope that inspiration occurs.

WHY AMERICAN BUSINESS IS SO CONFUSED

American business has come to believe that a computer is what IBM says it is: that it is enormous, horrendously expensive, and requires the services of a band of devoted priests. (The latest inexpensive IBM computers are "less than $1000 a month".)

American business believes this because IBM defined the nature of the computer to suit its marketing purposes. Now, confronted by lots of little computers — and most computers should have been little all along — businessmen are stumbling in circles, wondering how the rules of the game changed. Can their big computers have become obsolete overnight? No, it happened more slowly, but obsolete they are indeed becoming.

And in the next few months, we may expect to see a lot of frantic scrambling by electronics manufacturers, toy manufacturers and miscellaneous others to try to figure out what they should make next. Especially since, if they want to copy their competitors, the copied machines will be obsolete. Consultants and soothsayers should have a field day. And it may still be months and years before most firms recognize and understand what computers can do in the home, for individuals, for art, for leisure, for literature.

THE MISSED OPPORTUNITIES

Has personal computing been held back by the collective lack of vision of American business, or by the corporate villainy of IBM?

Probably more of the former.

IBM is culpable of having made its systems difficult and cabalistic and centralized, and thereby discouraging dispersed, imaginative and popular uses. But possibly not on purpose.

IBM did some research on interactive systems, like Janice Lourie's fabric design system and the hypertext work IBM sponsored at Brown with van Dam and Nelson, but since this work had to use 360 computers it was basically a doomed line of endeavor. (360s cannot be highly interactive — see "How Computers Should Always Be Used," later — and the typical 360 environments are harassed by notoriously chronic system crashes.)

What was needed in the nineteen-sixties was innovation in interactive computing with little machines.

Anytime from 1963 on — the date that Digital's computers got down to closet size — people could have created highly interactive computer systems for individuals, for writers, for graphic artists, for film-makers. Why didn't they?

Most corporations were locked into IBM, and their computer people would, or could, have none of it.

But some people did innovate: non-profit corporations like Rand and Lincoln Laboratory. Universities, especially MIT and Stanford and the University of Utah. The National Film Board of Canada. Bell Laboratories.

The late sixties especially were a time of high ferment and much financial investment in computers. Why didn't this bring computer systems for individuals?

Different firms were going for what they thought would be strong market opportunities for interactive systems in education and medicine, which did not pan out.

Nobody tried to offer commercial systems for writers, artists or home use. Not even the richest homes, which probably would have been an excellent beginning.

Digital Equipment Corporation missed a golden chance; but they were busy consolidating their product line. (Digital brought out graphical displays as part of its product line from the very beginning; but they did not press the advantage, and it waned.)

But another thing that has hindered this development is the lack of suitable marketing and suitable conceptual understanding. The potential customers had no idea of the possibilities, and the small computer companies, and their salesmen, had no idea either. They chose to market the computers to organizations for purposes they already un derstood, rather than try to begin a marketing effort in areas that seemed to them nebulous, the area of personal computing.

There should have been an attempt to develop personal luxury computer systems for the very rich. As a snob item, these could have led the development of computers for the masses. But nobody dared.

Where was the realization that highly interactive systems hold promise for individuals in all activities and walks of life?

It was dispersed.

Someone here, someone there, knew it was coming; but most of them saw no point in arguing.

A personal note. In my own speeches from 1965 on, I have rarely failed to point out that the real and true market for computers was going to be in the home.

People were skeptical.

THE TECHNICAL SIDE

We have spoken of the politics of marketing. But we must also consider the technical changes that have made home computers so cheap so suddenly. Without chip computers, personal computers would now be spreading at moderate speed; instead we have the explosion that is now beginning.

Let us briefly consider the inner parts of electronic devices, old and new.

INNARDS

Electronic circuits are arrangements of wire and a few basic parts that can in principle be hooked up in an extraordinary variety of ways: in one arrangement they become a musical instrument, in another arrangement a radio, or even an adding machine. There are only about a dozen fundamental electronic devices; elementary electronics is concerned with the way these tie together to produce zillions of different effects. Their hookups are called circuits, and different circuits can do any number of different things.

Electronics has come a long way in its fifty-odd years. Faster and cheaper, smaller and more reliable, smaller and cheaper, more reliable, and faster, faster, faster.

Radios were made with vacuum tubes till about 1950. Then they were made with transistors till about 1970. Now they are made with integrated circuits.

Computers were made with vacuum tubes until about 1960, and with transistors until about 1965. And now they are made with integrated circuits.

The vacuum tube, invented early in the century, used to be the fundamental amplifying and switching device. But the transistor, invented in the forties, has replaced it almost completely — and now single transistors have in turn been combined into integrated circuits.

The integrated circuit is to all appearances a rectangular piece of plastic with a lot of metal legs. Only its serial number hints at what it does. You look up in a book what it will do for you: what signals, into what legs, will cause it to send back what you want, as a signal on some other leg.

SILICON GULCH

The companies that produce these integrated circuits are an extraordinary lot. Most are nestled in a suburb just south of San Francisco, known informally throughout the world as Silicon Gulch. An apt nickname, for it fuses the main ingredient of the integrated circuit, silicon — also the main ingredient of sand and of glass — with the hint of western outlawry and cowboy style that really exists in this cutthroat, high-rolling, high-technical world.

The semiconductor industry, as it is more properly called, is a crazy and unprecendented growth industry based on the idea that there is an unlimited market for faster and smaller electronics. It has also been based on the immense governmental subsidy stemming from the arms race. No circuit is too good for our rockets. Thus we now have a standard of production quality for these tiny integrated circuits that makes the watch craftsman of Switzerland look like bumpkin stonecutters.

The integrated circuit business is a terrific gamble. First a circuit must be chosen for shrinking — some circuit that is expensive and in demand. Or some circuit that might be in demand if it were cheap.

This is a guessing game. The goal is to be the only supplier of a certain circuit for a period of time. So the problem is to figure out a good circuit for shrinking that your competitors won't have as well.

But as much as the cost of tooling up is enormous, the cost of making the individual units is negligible. "The first one costs a hundred thousand dollars, the second one costs a nickel." But the markups can be enormous. In the first months of sale of a chip, before the competition of the other chip companies begins to bring the price down, enormous markups are possible — say, ten dollars a chip for what costs a nickel to manufacture.

Once chosen, the circuit is studied and adapted for smallness. On wall-size maps, they work out little tracks and mazes for the current to run through, for all the circuit's purposes. The finished design is visual madness: Aztec street-maps with strange overlapping mazes and islands, visually indecipherable. These maze-pictures, photographed and shrunk to incredible smallness, become the patterns by which the tiny circuits are laid down. Now on cookies of silicon the maps are copied, their lines become smaller than fingerprint lines. The silicon cookie with its little maps is subjected to metal gases, electron beams, etching baths. Tiny transistors are not attached but actually grown in place by inoculating the surface with certain elements. Little runways of gold and silver run up, down, and around each other on several levels, like the subways and tunnels and sewers beneath bustling streets.

COMPUTER CHIPS

The first company to put a computer on a chip was Intel. (The computer chip really needs more parts to make it go — memories, a power supply, a box to put it in, connectors. But the central program-following functions are right there, on the one chip.)

They were originally intended for cash registers, small vending machines and other gimmickry that had to carry out tricky functions. The intricate computations required by your modern cigarette machine are by no means simple or trivial. The machine must make change, flash signs to indicate when change or particular brands are not available, and study the monies that are in the slot, and activate mechanisms that deliver the package and the change.

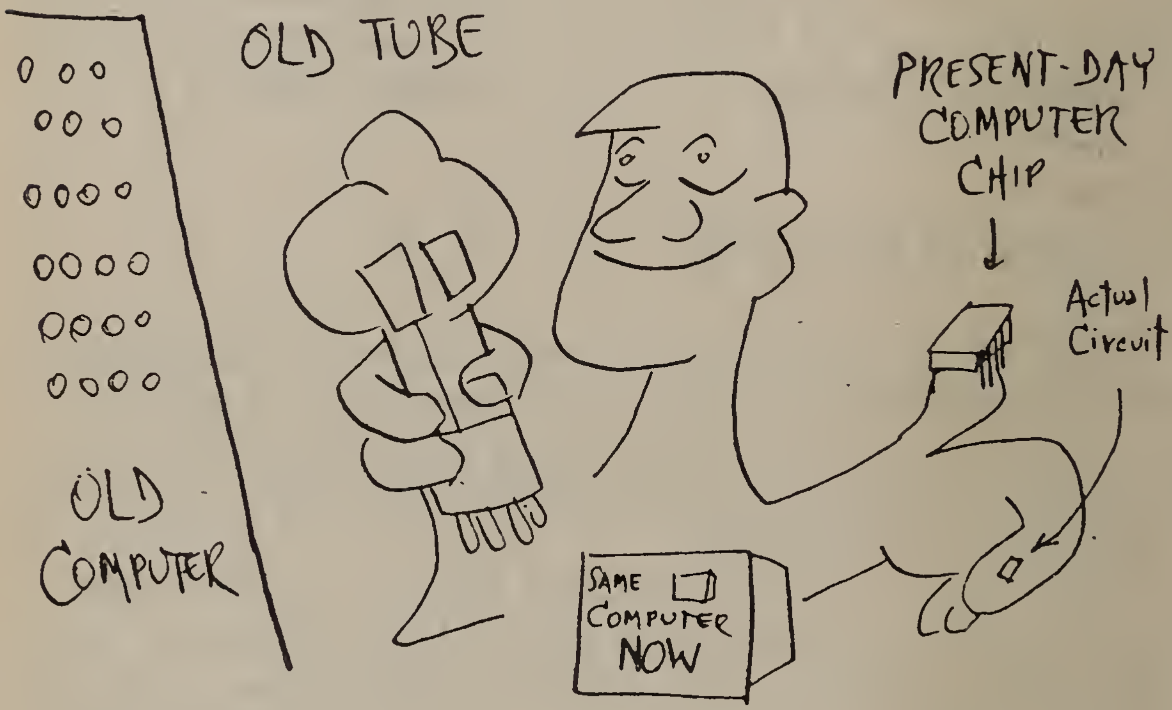
First came a couple of lesser computer chips, but the one that really changed the world was the Intel 8080. Intel's 8080 chip, it now seems, was probably as important to the development of our commerce as the cotton gin or the Colt revolver. Where the previous chips were fragmentary, and deeply restricted, the 8080 was a computer of significant power. Comparable computers, such as the BIT-8, had sold in the 1960s for $10,000. Yet here, in the chip-set sold by Intel for a few hundred, was a processing power just as great.

Like the blind men and the elephant, with dozens of different goals, different firms throughout American pounced on this product with different things in mind.

CHIP COMPUTERS

As we said already, only a little more circuitry is needed to make these chips into complete computers.

A few hobbyists started building their own computers. But the main companies did not see this, its greatest potential.



"DINKIES"

Computer chips are often called "microprocessors." Functioning computers made from them — chip computers — are often called "microcomputers."

But there is something objectionable about this. Telephones and radios are made with integrated circuits, yet we do not call them micro-telephones and micro-radios. We name equipment for what it does, not for what is inside it; a computer is a computer, just smaller these days.

The term "dinky computer," in the opinion of the author, best expresses the size change. There is no change in the computer's character.

HOBBY COMPUTING

The new chip computers had, and are still having, a devastinging effect upon the entire American technical community. While the first applications were quite obvious, what to do with them *next* was not at all obvious to most people. Consider the accidental way that the 8080 was first boxed as a home computer.

The scene is Albuquerque, New Mexico, 1974. A small electronics company called MITS — short for Micro Instrumentation and Telemetry Systems — is not doing terribly well. Its president, Ed Roberts, has resorted to painful cost-cutting methods to keep the company afloat — including putting all employees on reduced salaries.

And then Ed Roberts got his Idea. We don't know exactly how it came, and there are different stories circulating. The 8080 could be built up with additional circuitry to make a full functioning computer.

Roberts looked at this and, being a man of vision, saw the possibilities. In an act with vast, historical consequences, he sent an engineer into a back room to work out a computer design that would permit the easy addition of accessories and memories.

On a shoestring — indeed, on a wing and a prayer — MITS brought out their Altair computer. We may date the real beginning of hobby and personal computing from the Altair announcement, which appeared just before New Years. The January 1975 issue of *Popular Electronics,* which actually hit the stands in December of 1974, proclaimed on the cover, "A computer you can build yourself for only $420.00."

The response was overwhelming.

Roberts had estimated that 200 orders over the period of the coming year would have been a profitable number. 200 orders were received the first day the mail came in after that issue had appeared. A torrent of checks began. Roberts took out a beautiful full-page ad in the *Scientific American.* He played his hand with great daring.

While the engineering design had been substantially done, actual production of the Altair kit was not underway, and the company had insufficient resources to build all the kits and accessories that were announced. People did not receive their kits for long periods of time. And when these kits came, they did not always have a full set of functioning parts. One customer, Gary Mitchell, took an extremely interesting tactic. His kit did not come when expected and so he drove to Albuquerque and Ed Roberts placated him. Still the kit did not come and so he drove to Albuquerque again and was again placated by Ed Roberts. Now at last came in the mail his kit.

Surely, however, Roberts may be seen, however much by accident, as one of the great heroes of computing. For however much he may have annoyed people, and stretched the resources of his company to the disadvantage of some of his customers, nevertheless the Altair computer singlehandedly created an entire home computing industry. The Altair computer, too, managed to create a standard of technical interconnection which has made possible today's huge cavalcade of interchangeable accessories. To the jaded observer of the previous computer market, this seems too wonderful even to dream of.

On account of this, Ed Roberts has gotten his just rewards; MITS was acquired by a larger computer firm, which has the capital and management to help MITS make good on all its promises.

The Altair had, at first, no impact in the conventional computer world. The computer straights who saw the Altair ads, if any did, probably saw no reason to even relate to what was happening. This equipment was insanely below the usual prices. And wouldn't the fad bottom out any minute?

But others, unhampered by loyalty to the way things used to be, saw the possibilities for themselves — indeed, of selling accessories for the Altair.

Bob Marsh, a keen-eyed smiling engineer with a very kempt Pancho Villa mustache, came out with some accessory boards and started selling them out of his suitcase at hobby computer clubs. He became Processor Technology, Incorporated. Some graduate students at Stanford, who lived in a dormitory called Crothers Memorial — Cromem — started an accessory company called Cromemco.

The people who responded to the situation by choosing to make Altair accessories were people who *believed.* Either you believed in the Altair or you didn't. You had to believe that something entirely new had been added to human existence — the Personal Computer. And you had to believe it was going to become one of the biggest things around. And you had to believe the enormous demand was going to make profits possible even with small markups.

Some went further. Some even believed it was a new step forward for civilization.

THE FIRST TWO CRAZY YEARS

The two years that followed saw an unprecedented growth in the amateur computing field based largely on word of mouth and a new brand of fanaticism.

The chip computers started a strange phenomenon: the computer club. Even before the Altair, a peculiar breed of hobbyist started turning out to figure how to build their own computers from computer chips and to talk about, and show off, little computers they had constructed. Boston and Francisco were the two hotbeds.

Crazy California! The Southern California Computer Society, in Los Angeles, soon grew to 3000 members. (The author visited to speak one Saturday and found himself facing 700 people. They apologized for the small turnout.)

In mid-1976, an engineering firm called Imsai — a firm that had developed some very strange computer equipment that wasn't being sold — *copied the Altair.*

Now, there was nothing about the Altair that could be patented. The actual work inside the computer was done by the 8080 Computer chip; the Altair was simply a system of interconnections and a power supply for that 8080, set up in a particular way that almost any engineer could have designed. Intel was glad to have everybody know about the operation of the 8080, and so there was no secret. Thus anyone was free to copy the Altair.

But any electronics buff could recognize the faults of the Altair design. Imsai, however, was experienced with high-quality engineering around the same computer chip, and so they could make an *improved copy* that would be *interchangeable* with the Altair and its accessories.

Shortly thereafter there appeared the Polymorphic computer, another copy of the Altair, but somewhat smaller, and with a new feature — graphics.

A number of companies brought out computer kits that endeavored to buck the Altair-style trend, refusing to be compatible. But they were only hurting themselves by choosing a smaller market.

Byte magazine appeared. Subtitling itself "The Small Systems Journal," this was a magazine basically about and for the computer hobby, with a strong hardware orientation. In it were circuit designs for new interconnections and gadgetry, such as music synthesizers that could hook to your Altair. The demand for this magazine astounded the publishers, and within a few weeks the first issue had become a collector's item.

In June of 1976, Dick Heiser, a young economist at the Rand Corporation in Los Angeles, started the world's first computer store. Business was astonishing. He and his wife Lois both left Rand.

A computer store!

And it worked!

(Heiser is an economist. This helped, too, in the precarious world of small business.)

Suddenly there were dozens of computer stores, soon hundreds, all over the country.

A chain appeared: The Byte Shop opened franchised stores across the country.

By the end of 1976 there were at least 300 computer stores in operation, and the circulation of Byte Magazine was pushing 100,000. This number is astonishing both in itself, as the growing constituency of a wholly new pastime, but also as a suddenly emerging magazine of great readership power and advertising influence. *Byte* took a survey of its readership toward the end of 1976, in which it was revealed that nearly a third of its readers had PhD's and a third had their own computers. That meant 30,000 personal computers in place, two years out.

The first year of personal computing, 1975, had been one of confusion, actual success, and a number of small starts.

The second year of personal computing saw hundreds of new companies begin, while the first few that had made a strong start consolidated their position. The third year, 1977, is two thirds over as this is being written, and the big companies are about to move in force.

THE MASS MARKET MACHINES

In 1977, big manufacturers have started to roll:

COMMODORE

In April, Commodore, an adding machine company with millions of sales to its credit, introduced a unit they call the "PET Computer" — a designation recalling the Pet Rock of a few years before. This was a complete computer with keyboard and screen and cassette drive, all housed in a modernistic housing for $650.00.

HEATHKIT

It was not until August of 1977 that the long-expected announcement came from Heathkit, the venerable maker of electronic kits, that they would indeed be selling a computer. (This had been an open secret within the field for over a year.) Heath offered computers based upon both the

RADIO SHACK

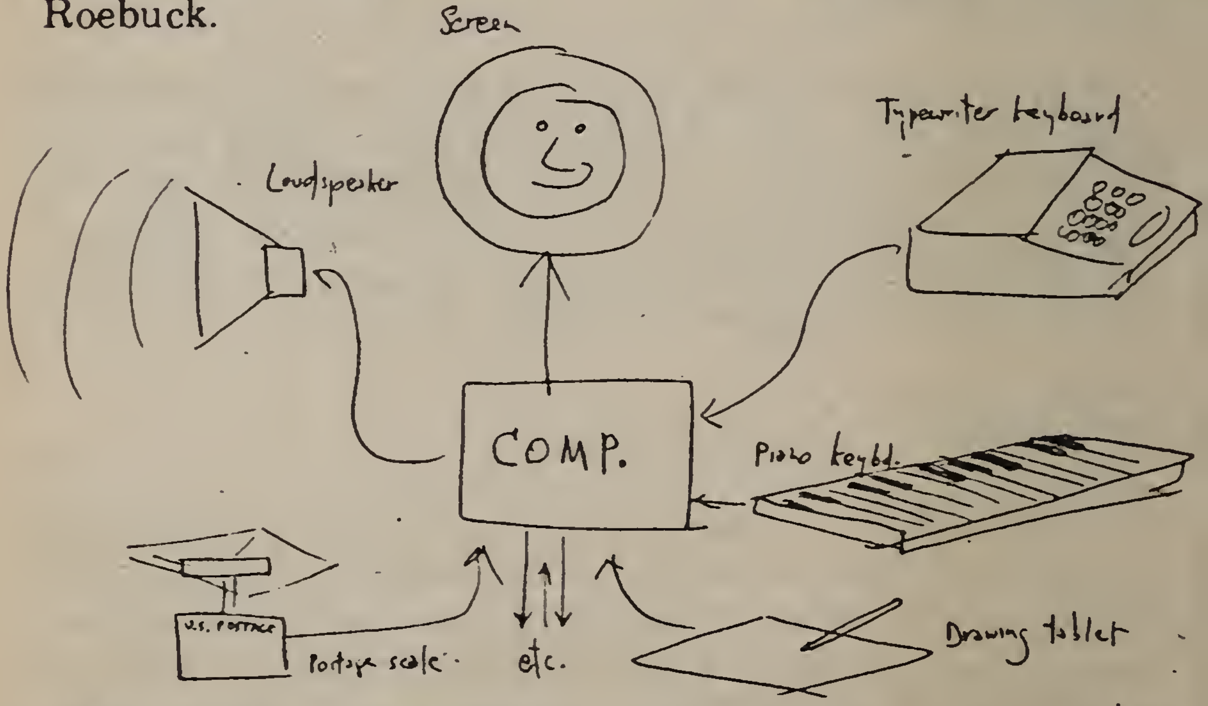
Radio Shack, a discount electronic supply house with thousands of stores across the nation, specializing in both parts and bargain equipment under their own brand names, announced a computer in August of 1977. This differed from the Heath announcement in several important respects: first, it was completely assembled, meaning that Radio Shack would take care of making sure it worked before and after you bought it. Second, it used the Z-80 chip, the super-8080 from Exxon that runs circles around the competition. And third, it offered the service department of Radio Shack if anything went wrong — a curious echo of IBM's preeminence in servicing, mentioned earlier. And fourth, if you wanted you could hook it to your own video monitor and pay less.

BALLY

Bally, the nation's manufacturer of pinball and gambling machines, has announced a home system — but the keyboard is not yet available.

SEARS

A home computer has been announced by Sears Roebuck.



PERSONAL COMPUTER (after the author's 1965 slide).
Now available in kits; or, approximately, in prebuilt units.

THE UNDERGROUND

The sixties also saw a sizable computer underground in this country: people whose idea of computer applications was not what industry wanted to pay for.

These were the computer freaks, the mad longhairs who hung around computers for the games, the thrill of power, and the excitement of the intellectual exercise. Counterculture computing continued on a running basis — indeed, a guerrilla basis — throughout the 60s, and into the present, with countless mad college dropouts plunging into employment at computer centers. In exchange for their dogged and tireless programming efforts for whatever cause, they would sneak their own games and pranks and private projects on their beloved computers.

Why didn't people pay more attention to the computer underground? Why didn't they realize it was the wave of the future? Because, bustling about in their let's-get-the-job-done manner, the establishment saw their ideas as only a nuisance — necessary because talented people held them, but a nuisance nevertheless.

There was always a running battle between the computer managers and the people who like to play games with the computers.

But the funniest part was that the straights were always looking for new business, and didn't see what was plain as the nose on their face: that new business was what they were treating as a nuisance.

Games.

The first official counterculture computer publication was a newspaper entitled *People's Computer Company.* A mad finagler named Bob Albrecht, who signs himself "the Dragon," had set up a nonprofit organization actually called The People's Computer Company in California, near Stanford University. Machines were set up, and townspeople were encouraged to play and learn on them. The strongest contribution of the People's Computer Company has been the development and publication of a number of computer games, written in the Basic language. Interactive and challenging, these games will fascinate anybody.

PCC was ahead of its time. Their games are now reaching many home users of computers, and in the late seventies they will be used by tens of millions.

In any case, the computer freaks live on. Since the Altair, many have merged with hobby computing. Others develop a sudden need for money and success, and stop playing. Still others are spoiled by their big machines, and become marooned in their ivory-tower playpens because nothing else is good enough for them.

"I'm waiting till I can afford my own PDP-10", they now say. (But even that may be only a couple of years, since a California firm says its hobby computer, the "Vacuum," will equal the PDP-10 — or the IBM 370 — next year, for $3000.)

WHAT IT IS

Undoubtedly you know the story of the blind men and the elephant. According to the legend, several blind men were introduced to an elephant, not knowing what it was. One felt its trunk and said, "the elephant is like a snake." One felt its side, and said, "The elephant is like a wall." Some felt its legs and said, "The elephant is like a tree." And one felt its tail and said, "the elephant is like a rope."

So it is with computers, only more so.

You could be led blindfolded to feel the separate parts of the elephant, and confused completely; but then by taking off the blindfold I could show you the real, breathing, physical elephant itself, and quickly you would see how each impression — the tail, the trunk, the leg — had shown you a part of the whole elephant.

There is no such panoramic view of the computer; no visual embrace will do it. I could show you, at a computer terminal, how the computer might do your income tax, or type your poetry, or draw a picture; but showing you the computer itself, as with the elephant, would not help nearly so much as showing you the elephant. Because the computer itself is merely a box with blinking lights — or without blinking lights, as the case may be.

To open it up is no more edifying. Will we see inside a little gremlin, pacing back and forth and scratching his head, who shouts down a tunnel when he has figured out an answer? Assuredly not. All you will see is flat boards of blue or green fiberglass, covered with tiny rivulets of silver that zigzag among shiny black dominoes of invisible purpose. You may see wires, and switches, and a light or two; you may see metal clamps with hundreds of fingers that bite the boards and connect their electrical circuits.

But you will see nothing that hints as to the meaning or purpose of a computer, or its operation. And to top this all off, if you open up a different brand of computer, it may look entirely different. It is not necessarily as with cars, whose four wheels and engine can be spotted by the rankest beginner.

No, there is a higher truth to be understood about computers, regardless of what is inside them, or what color they are painted. That higher truth is what they have in common. And it is so different from properties of any other machine that neither you or anybody else has a starting point for understanding it in previous experience.

Because of these mysteries, each person sees in the computer some quality, some purpose, that reflects his or her own background. For just as there are a myriad of human purposes, there are a myriad possible human uses for the computer. And just as there are thousands of places you can drive in a car, there are thousands of things you might decide to do with a computer that are different from what anyone else might want to do.

An astonishing variety exists in people's ways of looking at computers. And the way you look at a computer determines the way you will use a computer.

Even people who know a great deal about the computer are led by their preoccupations and ways of thinking to very different ways of seeing it. To most accountants, the computer is an accounting ledger with a little magician inside who is at heart an accountant. To many scientists, the computer is an enactor of mathematical expressions, helping them analyze their data and make theoretical predictions. (To pure mathematicians, the computer is a silly toy.)

To today's computer hobbyists, the computer is the most wonderful electric train you could ever play with. To tomorrow's computer kids, it will be a toy for making branching movies.

They are all correct.

The essential, central, basic, unparalleled nature of the computer is this: *it follows a plan.*

A computer is simply a blank device whose purpose is chosen, and whose steps are chosen, by a human being.

These steps are woven into an overall plan which is put into the computer. This plan is called a program.

The computer comes blank, without built-in purposes, and making it do what you want it to do consists of enumerating the actions that you want as a series of finely detailed steps. The resulting list goes into the machine, which then carries it out.

The crucial function of computers, the unique trait that makes them different from all other machines, is that they follow plans: lists of things to do. And because they have no intrinsic purpose, any list of things to do is just as appropriate to its circuitry as any other list of things to do.

The computer has no intrinsic purpose, any more than the typewriter. Just as a typewriter may be used for either commerce or art, to type sad words or happy words, a computer can follow lists which direct it to draw pictures, send out bills, or select the names of political enemies for dire treatment. It is not the nature of the computer to do any of these things. It is merely the nature of the computer to follow instructions.

It is up to a planner, the person who creates these lists, to study out the most practical series of operations for his purposes, and create the lists that the machine is to follow. It is in such a list that the planner will say, "ring the bell," "water the lawn," or "type the letter."

Computers can be hooked up to any electrical device in the world; and any machine can be run by electricity. Thus the computer can be made to run any machine in the world. So that the actions of a computer may in fact be actions of any conceivable machine, as woven into the plan stored in that computer. Computers can activate printers, picture screens, lawn sprinklers, musical instruments, juke boxes, music synthesizers, rocket launchers, atomic weapons, electric trains, cameras, water pistols, puppets, exhibits, theater lights, fish feeders, cattle gates, movie projectors, bells, whistles, klaxons, foghorns and chimes.

Actually the computer is simply the most general machine, since it can be instructed to follow any series of actions and control any other machines. It should never have been called a Computer in the first place. (John von Neumann, one of the first theorists of computers, insisted that it should be called the all-purpose machine, and he was right, but the first thing it was put to doing was repeated calculations, and thus it got the wrong name in the hands of narrow-minded people.

The French have an excellent word, far superior to ours, for the computer and its activities. The word is *l'informatique,* a combination of the words for "information" and "automatic." The same phrase in English would be "informatics." It would be a far better term, clarifying the idea and suggesting its breadth of application; but unfortunately that just happens to be the name of a large programming firm, a fact which has prevented the word from coming into usage. While some possibility exists that this word could be wrested from its present trademark status — a development which would be an immeasurable blessing for the understanding of computers — it seems unlikely that this will occur.

If you disagreed with this, you might point out that a rose by any other name would smell as sweet, and a computer by any other name would still be the same thing. Yes and No. People have been scared off by a lot of things, beginning most importantly with the word "computer" itself, which awakens everyone's old fear of arithmetic or mathematics. If we had had a word like "informatics," for everything the computer does, millions of people might have understood the computer sooner. These things matter.

THE SPEED OF COMPUTERS

As we have explained them here, the important thing about computers is that they follow plans. Well and good. But what if it took fifteen minutes for each small step of a plan? That would mean four steps an hour, or a hundred steps in a long day. You couldn't do much with such machines.

But as it happens, modern electronics makes computers very fast.

Very, very fast.

In fact, the speed of computers is hard to imagine. Most hobby computers carry out about *a quarter of a million operations per second.* The steps can be very small by themselves, as will be explained later, but their great rapidity enables a great deal to get done.

But this is almost an accident. Computers used to be a lot slower — twenty years ago they could only do a thousand or so steps per second. But the fundamental part is the following of a plan.

That has not changed.

DISGUISED COMPUTERS

The computer may be made to follow any plan and manipulate any other machine; thus it is the "most general" machine.

There used to be a brand of typewriter set up to type identical letters, whose contents had been prepunched on paper tape. When it got to the place where a specific address or name had to be typed in, it would stop and wait for the stenographer.

That machine was wired up specifically for the purpose. Today's corresponding machine isn't. There is just a computer with a permanent program that does the same thing.

Now that chip computers cost so little, there is little point in wiring up complex machines for particular purposes any more. It is easier to take the outside parts of the desired machine, like a typing mechanism and a typewriter keyboard, and hook it up to a chip computer inside. Thus all the inside wiring that was necessary before is no longer needed: instead a permanent program is set up to run the chip computer, and the machine behaves like the previous, and more complicated, automatic typewriter.

Thus there are general-purpose computers which you can set up with your own program, and *disguised* computers, which behave in a fixed way, because of the permanent, built-in program that is out of your reach.

Already we are in contact with many disguised computers. Video games in bars and arcades are now usually disguised computers. *Traffic lights* are often now disguised computers.

And this brings us to a general point:

EVERY OBJECT WILL SOON BE A DISGUISED COMPUTER.

The punch line of this is that every electrical object will soon be a disguised computer. Washing machines, refrigerators, radios, telephone switchboards, even automobiles, will use chip computers with permanent programs to avoid the cost of more complex circuitry, and to make them more versatile and flexible.

(Even computer terminals, these days, usually contain at least one computer chip, and thus themselves could be classified as computers. But because the computers have been programmed to behave simply as a terminal — passing on the alphabetical codes that the user types or the computer replies, terminals they are.

DATA AND INFORMATION

The words "data" and "information" are perfectly clear in ordinary life. Why, then, do they become so mysterious and sinister when computer people say them?

"Information" in ordinary life means whatever you hear, say, see, write down or find out. "Data" generally means the particular information that's needed for some purpose.

To computer people these mean *exactly the same thing.*

Because computer people have to deal very exactly with information, they've developed ways to measure it, box it, tie it together, ship it around. They can transform more things than you'd expect to information, including pictures and music. But the information, when boxed, is still about what it was previously; and it's still what you thought information was.

For computer use, all information has to be put into symbols; but the choice of these symbols is completely free. *Computers can store, and work on, any symbols you choose, according to any rules you wish.*

"But aren't there some things that can't be put into symbols?" asks a timid person in the front row. "What about poetry?"

Why, poetry is in symbols already: the letters of the alphabet.

The letter of the alphabet, indeed, is the fundamental unit of information. One letter — or indeed any character you can hit on a typewriter, including the space, the backspace and the carriage return — occupies a space in the computer which is called a *byte.* Computer memories are usually measured in thousands of bytes.

A thousand bytes is called a "K." Actually a K is like a baker's dozen, with a little left over: a K is really 1024.

WHERE TO PUT IT

The greatest problem with computers is *space*. There seems to be never enough memory space for what you want to do. (Cynics say there never will be, for the desires of computer people grow even faster than the memory equipment that becomes available.)

There are two kinds of memories: fast and expensive, and slow and cheap.

The basic memory of your home computer is not yet very big. Today's average home computer has a memory big enough to store only about 8 pages of typewritten material, or its equivalent. (That's 16K.) But much of this space must be given over to the instruction lists, or "programs" which tell the computer what to do; so there may be only 2 or 3 pages, in the typewriter sense, to store information that you want to work on. (This of course depends entirely on what you're doing.)

For this reason you need the other kind of storage, mass memory.

Today you can store any form of information on plain old audio cassettes — tens or hundreds of thousands of characters per cassette, depending on the form of storage used. It's the cheapest form of storage, but it's not very convenient, since you have to wait and wait for whatever you want to get in or out.

Everybody prefers to use disks. Disks cost much more than cassettes, but you can get at the information faster, and step through it much faster, and revise it much more quickly.

This is not the place to get into the disk problem. If you're ready for that, you're ready for another book. Or try your local computer store.

SOME PROGRAMS NEED DATA, SOME DON'T

In the guessing-game, earlier, the program required no data to be put in in order to start. (It *made up* the one piece of data needed, the random number.)

The secretarial program didn't need data to start; but the secretary had to keep putting *in* data.

Programs are all different in this regard.

PROGRAMMING:
WHAT THE COMPUTER IS REALLY ABOUT

It can be fun. It can be hard; it can be tangled and tedious; it can be very exciting.

Welcome to it.

Programming is the art of making things happen on a computer by listing the separate operations that you want it to carry out.

LEARNING TO PROGRAM

The best way to learn what a computer is is to program one. This means to create the detailed plan of operation which you want carried out.

There are many degrees of "knowing how to program." You do not have to master it to enjoy your computer; you can simply use programs and games others have prepared already. But if you are going to have any understanding of computers to speak of, you had better do a little of it to get the feel.

Indeed, there are several reasons for learning to program.

One is as a possible career. (Whether there will be many more opportunities in this field is open to question.)

Still another is that you, as a personal user of computers, may someday want a program that does not yet exist, or you may want to change a program that does not do what you want it to do. And in that case, you (or your employee) will have to fix it up.

But most important, you simply will not understand what computers are until you have at least tried to make one do your bidding.

To learn what a computer is, basically you must know how a computer follows a plan; and to understand this you had better try your hand at creating such a plan.

THE WONDERFUL WORLD OF PROGRAMMING

Hundreds of thousands of people are currently employed as computer programmers. It is a clean, fairly well paid occupation, mysterious-seeming to the general public and thus a little awesome.

But a good garage mechanic deals with technicalities which are just as difficult, perhaps more difficult, than most computer programming. It is merely an accident of our society that programming has developed an elite mystique. (More people are available to be garage mechanics, and consequently the trade is less respected.)

Not everyone programs a computer who touches one. The user of a simple game or business system, like those described earlier, is not learning how to use computers in general; he or she is learning the particulars of that individual system.

If you are a user, all you have to do is learn the *local* rules of that system. It's therefore as easy to use a computer, or as hard, as the local rules of that program make it. In a video game, the user is there to have fun. Thus the game must be devised in such a way tnat any new player can grasp the fundamentals in some thirty seconds. (Business programmers should learn this kind of simplicity. See the later chapter, "The Frontier: Clarity.")

The games player is not programming, he or she is a user. The secretary is not programming, he or she is a user.

But if you are a programmer, you must think out the machine's plan. More, you must reduce the details of the machine's plan to a very precise and complete list.

For instance, we earlier described a number-guessing game where the user tried to guess a number in the computer. The steps followed by the machine have a very simple structure:

Choose a random number.
Print (for the user to read):
"I am thinking of a number. Try to guess it."
A: Await user's typed guess. When it's received,
Compare user's guess with secret number.
Are they equal? If so, go to E. Otherwise,
Compare user's guess with secret number.
Is it higher? If so, go to H. Otherwise,
Compare user's guess with secret number.
Is it lower? If so, go to L.
H: Print "Your number is too high. Try again."
Go to A.
L: Print "Your number is too low." Go to A.
E: Print "Congratulations ... "

(Additional steps, allowing the computer to count the guesses, are omitted here for clarity.)

The above is a list of exact steps which specify how the computer must act to play the guessing game.

However, it is in plain English. There is nothing wrong with plain English, except that it is not for programming. While it could be made to work in this case, in general it is both inconvenient and plain difficult to use as a programming language.*

* A firm called Microdata has attempted to trademark a computer language *called* English, which is not likely to be honored by the law. Since English is notoriously a language which is owned by no one, it is hard to see how a court would allow "English" as a trademark to distinguish a particular brand of computer language to be sold for programming.

Many people, cropping up spontaneously in various places, come to the belief that computers can be programmed in some sort of ordinary English. This belief comes in about 28 flavors, which we will not bother to deal with here. However, many others believe English-language programming to be a mistaken notion, like another idea that many people have — spontaneously — of being able to square the circle. We will not take sides on this one, except to expresss doubt.

In order to make a real computer program out of this list of steps, we would need some means of expressing these steps exactly in a form that can be electronically responded to by the computer.

Any set of symbols can be used; the point is to devise a *regular* method of making such lists. Any such method, by which a human being may make explicit a series of operations to be carried out automatically, is called a COMPUTER LANGUAGE.

In practice, such a computer program might look like this:

```
10  PRINT 'I AM THINKING OF A NUMBER.
    TRY TO GUESS IT.'
20  LET A = RND
30  INPUT Z
40  IF Z = A THEN 100
50  IF Z > A then 70
60  IF Z < A THEN 90
70  PRINT 'YOUR NUMBER IS TOO HIGH. TRY
    AGAIN.'
80  GO TO 30
90  PRINT 'YOUR NUMBER IS TOO LOW. TRY
    AGAIN.'
91  GO TO 30
100 PRINT 'CONGRATULATIONS...'
```

(Again, for simplicity, we have left out the counting of guesses.)

This is a program translating our guessing game into the language called BASIC; it is thus in a form that can actually be used by a computer that accepts that language, as most hobby computers do.

In another example we considered earlier, a secretary types names and addresses onto a screen, prompted by a computer program. The steps are, roughly:

A: Print "NEXT NAME?"
Accept what the user types.
Print "Address?"
Accept what the user types.
(The user has now typed in two strings of alphabetical characters.)
Store the strings of characters on tape. (Or disk.)
Go back to A.

This, too, can be translated straightforwardly into simple computing languages.

But the mechanics are not the important thing, particularly as we get interested in more complicated things for the computer to do.

A MORE COMPLICATED EXAMPLE: VIDEO GAMES

Today's video games — the ones you see in penny arcades — are in many ways the forerunners of tomorrow's home computer systems: they have vivid graphics, zippy interaction, interesting noises.

As you play a new game in an arcade, or look over the shoulder of a player, consider the separate details of what the system is doing as:

A motorcyclist with wheels that appear to spin zips across a screen and appears to jump over a row of cars.

Two gunfighters stalk around the screen and appear to shoot at each other.

Two tanks, evidently thrashing about the ruins of the Roman Forum, take pot shots at each other. (When one is hit, a wiggly cloud replaces its image and you hear a crash. Then it rematerializes.)

One driving game is outrageous. You are driving a racing car, which must steer around many other racing cars, on a simulated roadway. In order to win the game, and get free time, you must rush and swerve around these cars at a heedless pace. Often you crash, and your car disappears in a wiggly cloud for a moment, then your car rematerializes, stalled in the middle of traffic. You begin again.

The video games were pioneered in the early 70s by a firm in California, Atari. The first of these was of course "Pong," a simpleminded ping-pong game where each of two players manipulates a line on the screen, called a "paddle," and a little square, called a "ball," bounces back and forth between the paddles. Suitable clip-clop noises are heard. Some people find this endlessly fascinating.

The bloodthirstiness of some of these games is depressing. One of the most alarming of these games was called Death Race 2000. You drive a car around the screen, and little people — identified as "gremlins," so as not to inflame any humanistic lobby — rush about the screen. You attempt to hit them. For each gremlin you hit, you get a point, but a gravemarker appears at the point of impact. This in turn becomes an impediment to further play, as it stays on the screen. Winning strategy is to wipe out the gremlins as close to the edge of the screen as possible, so as not to clutter the center. (This game appears to have been withdrawn, due to protest.)

In any case, more and more complex video games, with graphics, are coming out. There will be more and more of them.

Most of the video games now contain computer chips, and the creation of new video games is thus a good example of complex computer programming.

The visible process of programming consists of writing the steps of what is to be done by the computer in excruciating detail. (Programmers who are doing this look busy, which falsely reassures their employers.)

The more important part of programming is *thinking* about what is to be done by the computer; the thinking underlies and precedes the written program. You must think about what you want the computer to do, in all the different cases that will occur, and make numerous decisions that determine how the commands will be written.

The detailed steps of a program must always be completely specified one way or another; the computer cannot ordinarily guess what it is you intend for it to do.*

* There exist certain high-falutin' techniques for making the computer guess what you mean, but these are a lot of trouble and not of any common usefulness.

As we said earlier, the process of programming is really the process of *thinking* about what you want, making decisions and refining them to more and more detail.

The thinking that goes into video games is thus part of programming them. As a form of programming, the design of video games is a complex art, consisting principally of thought. Each kind of event must be completely thought through. The designer of a video game must work out in fastidious detail exactly how the game is to behave under every possible circumstance.

Just as for the secretary and the player of our number-guessing game, each small step of the computer's operation has to be exactly specified. A video game tricky enough to catch a passerby will need a large number of steps in its program, including the testing of responses by the player, changes on the screen, and sounds. You have to think it all out: how the controls will work, and connect the user to the world within the system; how the pictures will move; what visuals and side effects will be seen; how it will sound; what instructions to the user, and other numbers and symbols will appear on the screen; what little pictures and animations to use to simulate the game's events — explosions, puffs of smoke. Should characters say "Oof"?

THE TINIEST STEPS

In a program you give directions for thousands of things to happen within the computer itself. You may want information to be moved around and tested and replaced, and added together and subtracted and multiplied and divided, and slid back and forth, and used as directions leading to more information. Or send out as directions to accessories. There are many different external devices that can be hooked up to computers, and each of their separate functions has to be commanded somehow.


BAM?
OOF?
POW?
AIEEE?

All these are valid things to do; all are necessary. It is for these reasons that a computer must have many different built-in functions, smallest acts whose performance is wired in to the computer as an elementary reflex. These are called the primitive commands of the machine, or, more usually, machine-language commands.

At this most fundamental level, programming is done by making lists of these primitive, indivisibly simple acts. Ultimately these are the steps the programmer must put in his plan, or program. These are the elemental acts which are indivisible from the programmer's point of view. They are not made up of more than one command themselves. (For instance, if the computer sends a character to a terminal, that is such an indivisible primitive step. There is no command within that command, say, to transmit half a character.)

Programs can be written in these indivisible machine language steps, but that is rather like telling a man how to go to the drugstore by writing "Step with your left foot, step with your right foot" a hundred times. Instead we give someone directions to the drugstore by telling him where to walk, what signs to look for, and when to turn. We leave to his discretion whether he begins on the left or the right foot, skips, stops to look in store windows, or goes out of his way for an ice cream cone en route. The manner of locomotion is up to him.

It is the same with creating steps to instruct a computer. Though millions of the primitive steps usually take place when a program runs, the programmer need no longer concern himself with listing them exactly. The programmer is able, instead, to build his program of larger steps and decisions. Just as we told our friend on the way to the drugstore to look for a large blue sign, and turn left when he sees it, we would like to direct the computer without paying attention to the tinier steps.

(TOUGH OPTIONAL PART)

COMPUTER LANGUAGES

The original computer languages were simplified ways of writing down the undermost small instructions of the computer, one by one. This is still very respectable. But when such details do not matter in their particulars, we use programming systems that take care of such things for us, letting us think clearly about the instructions that really need our attention. These systems are often called "higher" languages, or just computer languages.

Surprise! There are thousands of different computer languages.

These are at least two dozen important computer languages; the experienced programmer generally knows between three and seven.

THE HIGHER LANGUAGES

A computer is, as we have seen, a device for following a plan. This plan can be expressed in any number of ways, provided that the computer is properly set up to recognize and carry out the steps of the plan. Computer languages are simply these different ways of expressing plans. And there is no single standardized way.

The different computer languages arise from the profusion of things computers can do. Computers can do so many things — pictures and music and printing and sorting, not to mention numerical applications — and more you think about it, the more different possible things you may want the computer to do.

There are many kinds of things people want done with computers, and many styles for doing them. Indeed, little astonishes the newcomer as much as the complete blankness of the computer, the fact that it really can be made to do anything whatever that its electronics will allow.

But different people have different things in mind. Since the very beginnings, many have used the computer for rapid numerical calculation. Others use the computer principally for business accounting and for storing records of business transactions. Yet others see the computer as an extremely deft motion-picture toy.

All these people are right, no one is wrong. But with these different emphases, and the natural variation of human mentality, many different styles of programming, and local rules of operation for programmers to follow, have come into being. By and by, using the computer for a given range of problems, and in a certain style, gives rise to a new programming language. A computer language does not jump out of the air. It is designed by someone to be a useful way of telling the computer what he or she wants.

Each of the higher computer languages allows you, as a rule, to program some particular range of problems, and in a particular style. In part this is because each language handles a lot of details for you automatically. Today's larger programs call in dozens or thousands of littler programs which have themselves been perfected — little programs for putting things in alphabetical order, typing a character on a terminal, moving a picture on a screen, and thousands of other functions. These are called *subprograms* and are of various types. While you do not want to have to create each of these subprograms, you want to be able to use them. So you need a shorthand method of telling the computer to carry out these little programs, and of tying them together. And such a shorthand method is a computer language.

Beginners are startled to learn what a lot of different computing languages there are and what little agreement about their merit among experts. Indeed, laymen commonly ask "how do you say in computer language?" and this has no general answer at all — because there are so many.

Just as the blind men misconstrue the elephant, and just as different computer users see the computer differently, different computer users likewise prefer different languages, because the different languages are tied to people's different ways of seeing and areas of concern.

People get very uptight about computing languages; the subject is as touchy as religion, if not more so. If you insult a man's favorite computer language, you cease to be his friend.

Indeed, there is no more emotional issue in the computer field than that of computer languages. While physical violence rarely occurs, the levels of emotional commitment and rage to be seen when computer people discuss computer languages is truly awesome. Many hobbyists who have only learned BASIC tend to go through this stage. Since all they have seen are programs in BASIC, all they can imagine is programs in BASIC, and thus they naturally think computers can have no uses *except* those which are easily programmed in BASIC. And indeed they get indignant, just like regular computer people, to hear anyone say they might be missing something.

The most important subject for the computer beginner is not electronics or mathematics; it is a subject that did not in any way exist thirty years ago. It is the subject of computer languages.

THE MAIN COMPUTER LANGUAGES

While this is not the place to get into computer languages deeply, let's at least do a rundown of some main areas. Because there are thousands of computer languages, there are also many different ways of categorizing them. This is a simple book, and the categorization we will make here is a simple categorization. (It might startle some professionals.)

TRADITIONAL LANGUAGES

In lumping together the following as "traditional" languages, I am taking a few liberties, but anybody who minds is probably too mad to have gotten this far in the book anyway. Traditional languages require the programmer to figure out ahead of time the exact division of memory to be used for each piece of information that needs to be stored or operated upon. One way or another, the programmer sets places aside for each kind or piece of information that will be needed. (This is one of the main pitfalls of the traditional languages, as it reduces their flexibility.)

FORTRAN

Because the first use of the computers were for arithmetical and formula computations, it was natural that a computer language should be developed which simplified the programming of algebraic formulas. This language was called FORTRAN, supposedly standing for "formula translation." Because it was the first, it became standard. One it was a milestone; now it is a millstone. People learn it first because it is standard. It was originally designed for mathematical applications; but it is in most cases far inferior for these purposes to APL (described later). But still they go on teaching it in the universities.

COBOL

Spurred particularly by the efforts of Grace Hopper at the department of defense, a language was devised for business application, called COBOL (Common Business-oreinted Language). It has certain strengths, but is very inflexible compared to the lambda languages (described later). COBOL programmers are the coolies of the computer field.

ALGOL

In Europe, mathematicians and scientists who became disturbed at the inflexibility of FORTRAN created a language capable of expressing (and thus programming) much more elaborate and subtle types of procedures. The resulting language, ALGOL, is widely used in other countries, and is standard even in this country as a way of writing down computing procedures so that other programmers can use and understand them. This is because it has no extraneous features, as does FORTRAN.

PL/I

The language PL/I (Programming Language I) was developed as an IBM product. Roughly speaking, it is a combination of FORTRAN, COBOL and ALGOL all together, preserving the complications of each and the distinct philosophy of none. Many companies with IBM computers use it, however.

BASIC

A group of determined young men at Dartmouth College, in the early 60's, created a computer system for everybody there to use, acting on the determination to make computers easy. For this they created a new programming language called BASIC, which was the simplest of all languages to learn at the outset. Since that time, BASIC has become the standard language of hobby and amateur computing, and indeed has caught on throughout the world for many other purposes.

"Basic" is not a description, it's a name. Essentially BASIC is a simplified FORTRAN. The BASIC language, then, is not (as you might think) language somehow intrinsic to computers, but a language which was created to make programming quick and easy.

The fact that BASIC is easy to use does not mean it is efficient, and there are a lot of things that simply cannot be done in BASIC. Truly complex programs can be created in BASIC only with the greatest difficulty. However, the new computers being set up for home use all come with Basic, and so its use is growing dramatically even while it limitatpons are felt ever more painfully by those concerned with creating really versatile and complex programs.

By common consent the amateur world is deeply committed to BASIC; but there is no exact standard of what BASIC is, and so there is plenty of room for improvement. One possible hope is that the best elements of LOGO (see below) could be slyly introduced to BASIC, until BASIC comes more and more to have some of the power of LOGO. (One sort of superBASIC, called GRASS, may become available soon for amateur machines.)

THE LAMBDA LANGUAGES

The second category of computer languages will be, in the opinion of the author, the important ones for tomorrow. They offer a power, and in some cases a simplicity, that has not been widely seen as yet. The Lambda languages are called that because they are based, somewhere deep down, on something called the Lambda calculus. But you don't have to know about that.

This mysterious thing, the Lambda Calculus, is simply a systematic way of tying things together; of taking the results of one operation and making then the starting point of another operation. The Lambda languages, accordingly, are extremely versatile, as the results of any operation can be used as the beginning of any new operation. Thus, they have few of the restrictions that are so common in the other languages. Space need not be exactly prearranged, as in the traditional languages.

The Lambda languages were first used in obscure research laboratories, especially those where many delightful odd people work on what is referred to as artificial intelligence (to be discussed later). The original Lambda language is called LISP, and it is so intricate and obscure to most computer people that its practitioners have come to be seen as strange eccentrics — a priesthood within the priesthood. Yet there was a reason for this strange computer language, and all of its frightening parentheses: anything which can be done in any other computer language can be done in LISP, while things can be done in LISP that cannot be done in any other compter language.

People versed in FORTRAN and COBOL were alarmed by LISP because it contained hundreds of parentheses. The parenthesis is the most common character in LISP. This annoys and offends those who don't understand it, because they naturally think anything can be programmed in FORTRAN and COBOL, which is not true.

But LISP ordinarily only runs on big machines (although a group at MIT is endeavoring to build a LISP machine small enough to be a personal computer.)

There are, however, other languages which have all the power of LISP and yet have certain other advantages. An important one of these is LOGO. Created by Papert, Feurzeig, and others, LOGO is as simple to use as BASIC, but far more powerful. It may well become available for hobbyist computer machines in the near future.

A group at MIT, doing research in LOGO as a tool for teaching programming to children, asserts that in two weeks of instruction, children who were taught LOGO could program circles around children of the same age being taught BASIC for comparison.

But LOGO has so far been a washout for political reasons.

Picture the situation if you will. Some extremely bright and visibly eccectric people, who have very little respect for computer programming as it is ordinarily done, have been saying that computer programming should be taught to very young children in a way that most computer programmers don't understand. They have asserted that this scheme will make the children better programmers than the professionals; and they have sought funds to carry on this teaching in schools where nobody knows what a computer is at all.

Such is the computer field.

Another Lambda Language which may become important is TRAC language, invented by Calvin N. Mooers, the same man who brought you the phrase "information retrieval." (Mooers may sue me if I neglect to mention that TRAC is the trade mark and service mark of Rockford Research, Inc., 140 **and one half** Mt. Auburn St., Cambridge, MA 02138. He does make things difficult for those who try to use it without his permission.)

TRAC Language will run on a much smaller computer; one authorized version of TRAC language runs in only 8K spaces of the main hobby computer. TRAC Language is like LISP in that is uses many parentheses. Computer people who have been turned off to LISP — and that seems to be a lot of people — see the parentheses in TRAC and say, "Forget it". People who only know BASIC often have the same reaction.

But TRAC has certain special qualifications which ideally suit it for the very small computers that are now becoming so very widespread. It does not need large amounts of memory, and it has important features for highly interactive systems. The ability to control user input, so that if a user types the letter "F" he instantly sees, say, a picture of a fish instead of the letter F — is an extremely important feature for user-level systems of the future.

The last Lambda language we will mention here is probably the most exciting. It is called SMALLTALK and was devised by Alan Kay and his associates at Xerox Palo Alto Research Center. It's written up with neat pictures in the September '77 issue of the *Scientific American*, 231-244.

This language was created around Kay's notion of a personal computer, which he calls a "Dynabook." (Apparently the term Dynabook simply means a computer that you can program with the SMALLTALK language.) But Kay and his associates have proceeded on the correct assumption that it would be possible within a few years to build a computer the size of a book that will run on batteries, have an elaborate graphics screen, and sell for $400.

This prediction, which seemed outrageous to some people only a few years now seems firmly possible for the year 1980. Whether the management of Xerox, deeply entrenched in a paper-oriented way of thinking, will understand this development and bring it to market, remains to be seen. SMALLTALK, anyway, is a Lambda language with numerous exciting features. The parentheses are few, not the tangle of LISP. Instead, some commands of the languages consist of smiling faces and pointing hands, amongst the other symbols and phrases.

Secondly, the language is set up for the use of a finely detailed computer screen, of some half a million dots, on which the programmer may typewrite in numerous typefaces. SMALLTALK may produce dazzling animations on the screen, interacting with the user. (In another amazing form of interaction, Kay hooks SMALLTALK up to an organ keyboard coming out of loudspeakers through the computer. At the same time, the SMALLTALK program shows the notes on the screen, transcribed from his pressings of the keys.)

SMALLTALK programs are sectioned into a number of parts, called "processes," which are independent entities with a special kind of autonomy. Processes cannot interfere with each other, and thus a program may be debugged, or corrected, by sections.

But numerous copies of a process may exist. SMALLTALK programs, amazingly, are much more "like real life" than most computer programs. For instance, if you write a program to simulate traffic, you have one copy of the "car" process for each car on your highway.

IF you've done ordinary programming, you know how odd that seems to most programmers. Yet it has an intuitive simplicity. Thus SMALLTALK may turn out to be both the most powerful computing language and the ideal language for beginners. (Let's hope Xerox management gets moving on it.)


PARROT

OTHER LANGUAGES, ESPECIALLY APL

There are many other languages; some have very specific ranges of purpose, others are "general purpose" but reveal a certain slant and certain special aptitudes. Foremost among these other languages is APL, or "A Programming Language," devised by Kenneth Iverson. Iverson is a fiery and upright figure, with the dignity and self-certitude of a Raymond Massey, or a religious leader.

Iverson claims that his language was always intended as a way of writing things down, especially for mathematicians and scientists, and feigns surprise that it turned out to be "a good way to drive a computer," For Iverson's notation is a powerful and elegant system of expressing mathematical meaning. Having detected, as a young mathematician, that the notations of science and mathematics are really quite chaotic and irregular, he began writing them out in a form which adhered to certain basic rules. Working all this out, he gradually put together a notational system of complete generality.

No attempt will be made to give examples here. But Iverson's language has become one of the most influential forces in the world of scientific computing. APL is a work of art, not unlike a beautiful set of surgical tools, or a set of matched gems.

Iverson's language permits the expression of mathematical concepts from across the whole of science and statistics, thousands of different ideas and functions each resolved to a crisp and concise expression in this new, common form.

The language requires learning new symbols, but a few hours of time spent with an interactive terminal and a good tutor make one able to do astonishing things.

It is interesting to note that APL has come into use almost entirely on a word-of-mouth basis. An ever-growing fraternity of scientists (and, more recently, business users) have discovered its power for a vast assemblage of purposes.

The original APL program was created within IBM, not as a planned product, but as a private project at the initiative of Iverson and his friends. But the language then caught on within IBM, becoming addictive to its users, and became a part of the IBM product line by popular demand from the outside. It is now affecting the rest of IBM's product line, as both scientific and business users work with it more and more.

APL is now available for personal computers, especially the 8080. (Prices vary from $10 to $650 for different versions.) One version sells for as little as ten dollars; but that from Microsoft, a very respectable programming firm, it is expected to sell for about $650.

For many purposes, APL is slow and inefficient — especially for interactive graphics and music. But then again, David Steinbrook, a doughty young composer, is using it as a music machine anyway, and maybe he's onto something.

IBM sells a small computer that runs APL. This is one of IBM's best products. However, because of its cost ($5000 to $15,000), we will not consider it here as being within the range of personal computers.

OTHER NON-STANDARD LANGUAGES

There are fifty or a hundred languages that ought to be mentioned. But you can see there is no room for that here. The different languages embody different ways of thinking, different styles, different purposes. Many are variations of ALGOL. (If you want to immerse yourself in the great range of them, Jean Sammet's monumental book on programming languages is surprisingly readable.)

Suffice it to say that if you get serious about computer programming, you can make computer languages your never-ending study. Of if you go to do research at the Gazerkis Institute o f Tough Science, if there is such a place, you will probably become a fan of their language and see no other.

THE PROGRAM YOU SEE MOST OF

Many computers, big and small, come with a program that serves as a general butler of the computer system. Sitting at the system, you ask it to bring forth whatever programs you want to use, or put away data in the closet (i.e., on disk or tape).

This program butler is the operating system, or monitor. They are good things to have. They are offered for many dinky computers.

Sometimes a language processor, such as the BASIC processor, serves also as a monitor, and will store data and edit files with you. Such monitors come with most amateur versions of BASIC.

UNIVERSAL PROBLEMS OF SOFTWARE

"Software" means computer programs. Regardless of your area of interest or the language we use, some questions are inescapable.

DEBUGGING

It is natural to make mistakes while you are programming. Some people get better and better at programming, and make fewer and fewer mistakes. However, the mistakes anybody makes can be awfully big ones.

Mistakes in programming, also called bugs, are not easy to find. Surprisingly, it is impossible to tell by looking at a computer program whether it will work or not. The only way to test a program, except in a small number of mathematical cases, is to try the program and see if it works. Indeed, a program may work correctly at one time and yet have hidden bugs that may make it fail later on.

The problem gets worse as programs get bigger. Ordinarily a medium-sized program does not work the first time. Or the second. Or the tenth. But the human creating this program, struggling to find his omissions and mistakes, perfects small pieces of it at a time. And with the perfection of each piece, gets a sense of drawing closer to the overall goal.

The complications of computer programming were not obvious at the start. Henry Tropp, who has done a research project on the history of computing, interviewed the man who discovered debugging, an English scientist. He wrote a program for a computer of the nineteen-fifties and discovered that the program did not run correctly. He found one of the errors, changed it, and discovered that the program still did not run correctly. With sinking heart it occurred to him that he would spend the rest of his working life "attempting to correct my own mistakes."

The programmer subsists on piecemeal reward, sometimes a little reward for a lot of effort, sometimes a great reward for a little effort, sometimes seemingly no reward at all. Yet this intermittency of reward, and the rare grand feeling you get when it works, seem to be enough to keep great numbers of people hard at work in programming activity. (Behavioral psychologists are quick to tell us that intermittent reward is the kind that promotes learning most effectively. But what may be more important is the good feeling when the program works.

CAN A THING BE DONE?

We get lots of ideas for things to do with computers; but not every idea is doable.

A very serious problem for the beginner is not knowing what constitutes an undoable problem, or one which is just too big. The beginner, successful with a small project, rushes right on to attempt the impossible, rushing in where experts fear to tread. (But it is just through the fearlessness of the newcomer — the kids who know no fear or modesty — that many important innovations occur.)

STRUCTURED PROGRAMMING

A new set of rules is having a great impact. "Structured programming" is a set of rules for writing programs that are easier to debug, cheaper to produce, easier to improve or fix up. Basically, structured programming means dividing the programs into blocks of certain kinds, which behave and interrelate in certain ways. The rules are just a hair too complex for this volume.

Structured programming has become sort of a religion in recent years, spread by its founder, Edsger Dijkstra of the

Netherlands; by Harlan Mills within IBM; and many others, notably Henry Ledgard, author of *Programming Proverbs*, and Brain Kernighan, author of *The Elements of Programming Style*.

Basic, and some traditional computer languages, are difficult to use according to the rules of structured programming. This is beginning to look like a strong argument for the Lambda languages, and certain others; they make it possible to get your programs running faster, and change them more readily.

THE COMPLICATIONS GO ON AND ON

Mankind is just learning what the consequences and complications of such plans of operation — detailed computer programs — really are. In the twenty years since programming began, it has been studied extensively. A great deal has been learned within the field about how programs work, and even more has been discovered that is confusing and unknowable. As the amount of known territory has increased, the amount of unknown territory has increased even faster. Programming is still an art, not a science.

Each small step forward has revealed the immensity of the unknown void beyond, just as astronomy in the 20th Century has shrunk mankind faster and faster in an unthinkably large universe.

SOFTWARE QUAGMIRES

It is all too easy to keep trying to fix programs that were really very bad in the first place, and throw good money and effort after bad. What is worse, other people have to use the quagmire that is thus created. (IBM, indeed, is notorious for their cumbersome and sprawling software — but if the customer is locked in, IBM profits from the inefficiency of that software).

Yet, like the Vietnam War, software can become justified simply on the grounds that too much has been invested in it already.

It is best to take the advice offered in *Programming Proverbs* by Henry Ledgard. "Don't be afraid to start over."

PROGRAMS AND PROGRAM PACKAGES

Since the 1950s, computer programs have been valuable objects of sale. Programs have been sold for wide varieties of purposes — usually for business, but also for science and government.

The price of individual program packages has always been, of course, what the market would bear. It is not uncommon for a language processor to cost tens of thousands of dollars to a user organization. Application programs — for specific business uses on large computers — can also cost tens of thousands of dollars. Programs may be rented instead, in which case the monthly payments can be very, very high.

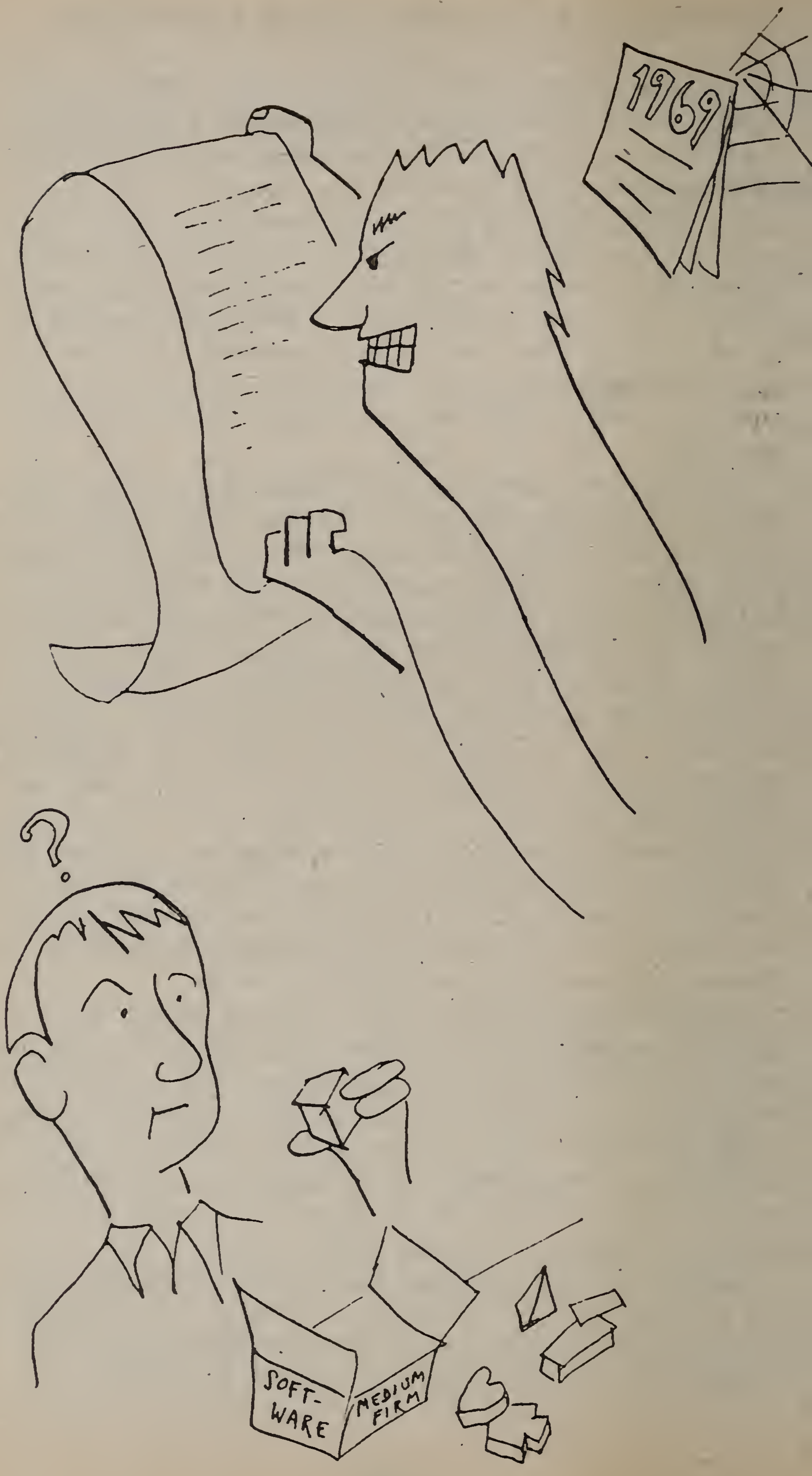
This has been the world of software. The little computers, though, should have a drastic effect on the price and style of software. Right now nobody quite knows *what* effect. What is going to happen with software in the amateur market is a mystery, but we can expect the price to go down for businesses. The price of good programs for personal users may go up into the hundreds. (Thousands???)

Depending on what hardware becomes popular, programs may be sold in little wafers, or sticks like chewing gum, or cubes, all plugging into the computer somehow.

And some will be sold as they already are, on cassettes and paper tape and disks. All these are merely forms of storage for the programs, the series of commands that run the computer. But because programming is hard work, the programs may be sold as objects of value.

The principal software for the personal market will consist of canned interactive systems for an every-widening spectrum of purposes, and in a growing range of styles. The programs for your home computer will not merely be sold singly. They will also come in suites, that is, integrated collections of programs that fit together. (We may even look forward to panoramic software, linked programs for a broad spectrum of personal uses.)

We discuss below the legal matter of software protection. But whatever outcome there is to this legal issue, there will surely evolve a stable fashion by which developers of good programs can receive financial reward for them.


1969
SOFT-
WARE
MEDIUM
FIRM

CAN YOU GET THE PROGRAM — AND IS IT IN YOUR STYLE?

You can do anything with your computer that you have a program for. If you buy a canned or prepared computer system for some purpose, you do not have to learn to program. You are like the game-player and secretary mentioned earlier. Most personal computer applications are going to use software somebody else has developed.

Now, either the program exists, or it doesn't. But just because there exist programs for a given purpose does not mean they are any good.

There is usually considerable leeway in how a program can be designed. Programs that supposedly do the same thing can be as different as hats, or dogs. Many writeups on home computing in the popular presses might give the impressions that the computer will do whatever you want, in the style you expect, with someone else's program. This is almost never true. You will have to adapt to another's idea of what aspects are important, and how they are best explicated in the program. Even if a program like the one you wanted exists already, it probably is not in the style you would like. And if it does not exist, you are going to have to create it. One's personal fantasies, often so clear, tend not to be what the other guy programmed. (Great disappointments occur.) Each person's preferred style of use may be different from another's.

Unless you are the one who programs it, it will not be focused as you would have it, nor as flexible in ways you might want.

If you are going to use a pre-existing program, you have to adapt to *it*. Otherwise you must program it yourself, or adapt the pre-existing program. Thus you must learn how to program. The same goes for many of these new applications you are going to have to program yourself, or have somebody do. "If you want a thing done right, do it yourself." The way you design it is crucial. Can it be made easy to use? Making things easy for people is hard. But it can be done. You have to try hard enough, and be able to visualize. (See the later section, "Virtuality.")

SELLING SOFTWARE AS HARDWARE

Programs are being sold on paper tape and cassette. When loaded, their contents slide into the otherwise empty spaces of the machine. When you're done, you obliterate the old program and use the computer's memory for something else.

However, programs will also be sold by some manufacturers as little plug in thingies. "Thingies" is a vague term, but these plug-in programs can come in any size and shape. Some are now sold, not for computers but for calculators, in little wafers the size of sugar cubes.

These are ROMs — Read-Only Memories. These little memories, filled with their programs, behave just like the regular changeable memories of the computer when they are temporarily loaded with a program. But the ROMs are permanent.

There is no real logic distinction between one type of program and the other. But the Roms are more convenient — and people are a perhaps less likely to copy the programs that are on them.

But this is not clear. Let us consider, at this point, steps that can be taken to enforce the ownership and salability of programs.

PROGRAM PROTECTION

Most amateur computers can presently use each other's programs. By law, the owner may charge anyone who wants a copy of a program he has developed — but in fact, one hobbyist may easily give a copy to another on the sly. This is the copyright problem. There has been a great deal of program copying by hobbyists in the last couple of years. Nobody knows how much, and of course nobody — except a few troublemakers — is going around bragging that he has done this.

It is easy to make a perfect copy of much of the software for little computers.

Herein lies the temptation.

Business users pay readily for software, since it is an obvious business expense.

However, amateurs who only "want to have fun" are hard to persuade that making a copy is as dishonest as counterfeiting a dollar bill. In the next few years, however, it will become clear how much most people will depend on programs that are developed by others, and how very much better some of them are than others. This will affect people's thinking on the issue.

INTELLECTUAL PROPERTY

Just as background, let us review the main ways that United States law allows you to own something you come up with in your mind. A lot of people seem to think you can patent or copyright anything. This is far from the truth.

(Note that these laymen's descriptions should not be taken as a legal guide. Consult a lawyer for the exact information — and the latest. Things are changing fast.)

The law provides several methods by which people are granted certain rights to things they make up:

PATENTS

The most famous of these is the patent. The patent is expensive to get, may not be binding, and lasts for only 17 years. It protects your invention only in the narrowest sense: with reference to certain specific features which nobody can copy without your permission.

The patent was established by Congress with the stated intent of encouraging the communication of technical knowledge. For this reason it must describe fully what is being covered. In return for this description, the government gives the inventor exclusive rights to the invention — in the narrow sense covered by the wording on the actual patent — for the 17 years.

To patent something, you must search to see what is already patented, or known, that is like it. If you think yours is original you submit a patent application. Your attorney argues with patent examiners for months or years, then maybe you get it and maybe you don't.

The expense of getting a patent is generally several thousand dollars — in part based upon the attorney's judgment of your willingness to pay, in part on the complexity of your patent application. But such sums are usually out of reach for people trying to start a business on a shoestring, as most of the people interested in this matter are.

Furthermore, there is some considerable doubt as to whether patents can be obtained for computer programs. The Supreme Court has ruled lately that programs by themselves are not patentable, but that clears up less than some people think.

Until such cases have been further tried in the courts, the true status of the law will not really have been decided. That is the way American law works.

PATENTING SOFTWARE AS HARDWARE

A number of patents have been issued on fictitious machines. These machines are described with care in the patent documents but the claims are written to be actually satisfied by ordinary computers holding a certain program.

Thus, since the description in the claims exactly applies to a computer holding this program, the document could be said to have "patented the program" by patenting all uses of it. No one knows how many of these things there are or whether they are valid.

COPYRIGHT

Another very important form of intellectual property is copyright. This was originally instituted for the purpose of protecting an author's right to publish his own literary works from those publishers who might otherwise print it without paying him.

For this reason, the copyright is granted to an entire body of writing, — say a book or a play — almost automatically, and no attempt need be made by the artist at the outset to decide what, if anything, is unique about the work. That is left for the courts to decide if and when a copyright holder sues someone else as infringing on his copyright.

Copyrights are cheap — the material is supposed to be filed with the library of Congress, but in some cases is thought to be protected even without such filing. This will be rendered more precise by the new copyright law soon to go into effect, which holds that an individual need not even file his work for copyright. It is automatic. Soon there need not even be a copyright notice printed in it, a ritual observance formerly considered to be at the very core of copyright. The old copyright law held for 26 years and was renewable for another 26; the new copyright will hold good till 50 years after the author's death.

For computer programs the copyright question is this: since computer programs may generally be written on paper, and consist of symbols, are they writings? Does the 26-year protection of copyright, now granted almost automatically to authors of novels, apply in equal force and sense to the copyrighting of computer programming?

Some think so and some think not.

To those who believe in copyright, the same protection applies by a very clear extension of the existing laws and court decisions. Those who are not in favor of it point out that it was never intended to cover such things and therefore should not.

We take no position on these matters. They're tricky.

3. TRADE SECRET

There is also the possibility of simply keeping software's content secret, which can be done in various ways. This "trade secret" is the final, and perhaps the strongest, way of protecting intellectual property rights. Many programmers claim to be able to figure out how any program works. They probably can't.

HOGGING THE MACHINE FOR TEMPORARY PROFIT

Standardization benefits you far more than you realize, because it is actually an absence of impediment. Imagine if half the different states of the U.S.A. drove on the left. Or had different official spelling. Or used different sized typewriting paper, or used different television systems or had different rules about what advertising could be used on the air.

And so on. Forever.

Stress the "forever," because there is no limit to possible impediments! It's just that when things are standard, there aren't any.

The reason standardization is good is that it makes things easier, and so you can do more with less effort. (Those who argue that things should be made difficult for some moralistic reason are not welcome in this book. May their corridors never end and their forks be too heavy to lift.)

Furthermore, any standard is far better than no standard. When it comes to important things, like which side to drive on, or how to make home computers, a standard is crucial. One has come into being for home computers. It is called the S-100 standard. Designed as the interconnection system of the Altair, it is the arrangement by which all the compatible accessories manage to be compatible.

The manufacturers of the prebuilt computers, however, seem to want to monopolize their machines so that only programs they write will run.

But these people should study the example of the Altair — and of the original Philips cassette. Both are more or less in the public domain, but by setting the standard and allowing other makers in, both manufacturers assured a much greater market than they could have had if they had kept a monopoly.

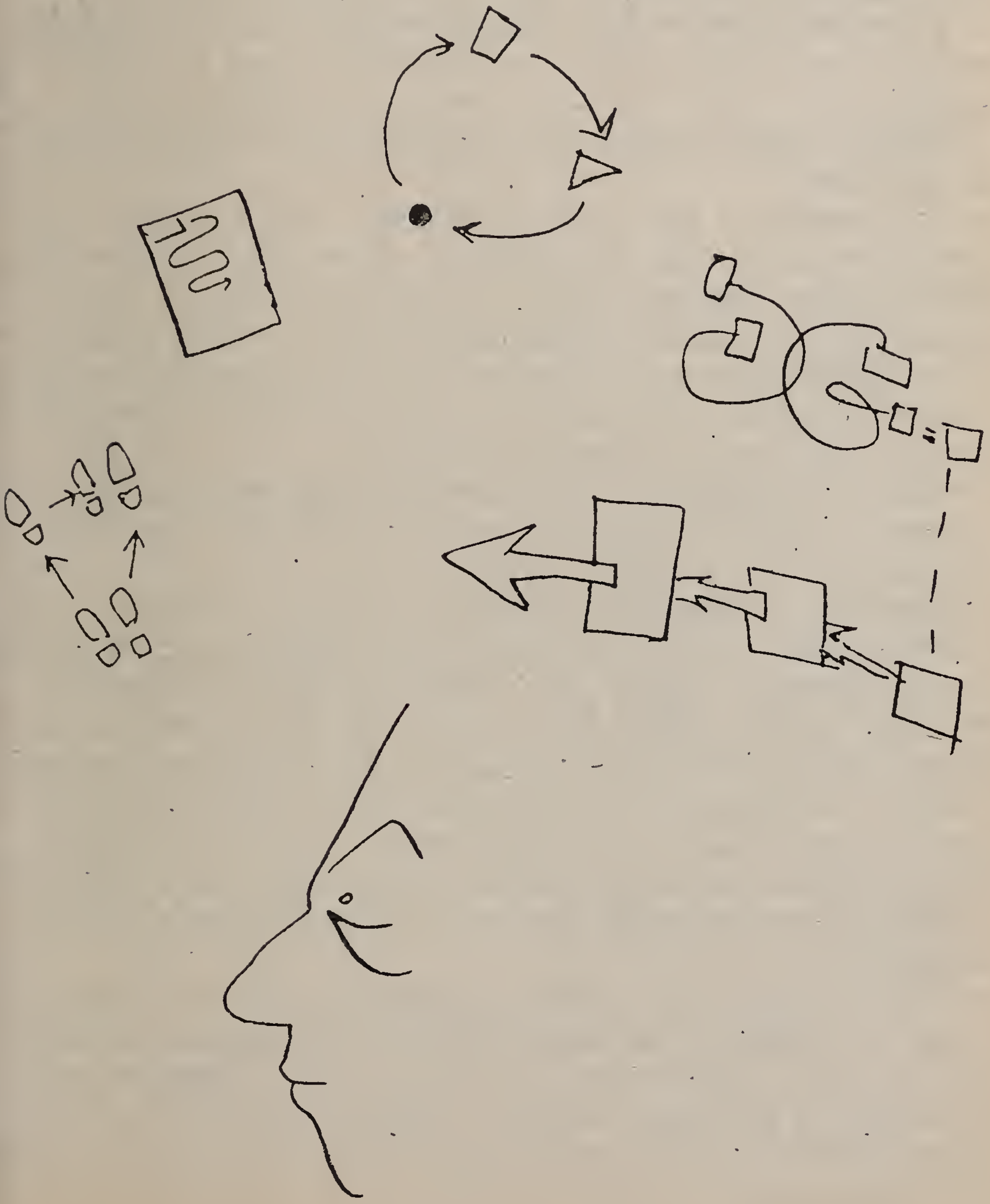
The manufacturers who have broken away from the S-100 standard have set things up for themselves in the short run. Software for the PET, or the Compucolor, will be sold in permanent memory chips which are plugged into the computer. Since the programs will not work on any other computers, these manufacturers have assured a captive market for their software which other sellers are not likely to invade.

But this is self-defeating in the long run: it leads to increased separatism, diversity, and incompatibility among computers. Users who realize this may search for machines closer to a standard configuration.

Trying to keep other people's programs off the hardware you manufacture is self-defeating.

Trying to sell a machine that only takes your programs is like playing roulette at Monte Carlo; you may or may not make it big. But letting everybody sell software, and taking a cut, is like *being the house* at Monte Carlo.

(END OF TOUGH OPTIONAL PART)



HOW TO GET STARTED

There are many ways to begin your involvement with computers, depending on your finances, your time, your personal style, and what you want to know most.

There are many levels of involvement, and some people want to get closer to the tiny steps and inner wires than others.

The best way to learn about computers is almost surely to buy one.

You can buy a high-power computer — the equivalent of fifty- or hundred-thousand dollars machine in 1962 — for four hundred dollars. Add a TV screen, and you're set to start playing — ahem, *exploring* the magic of the computer.

It happens that the best way to understand computers is, to try to program just as the best way to learn to swim is not to read about it, but to try it.

And to understand a computer is,most fundamentally, to understand how it follows a list of instructions. In other words, practice programming it.

But wait — shouldn't you take a course instead?

Well, most courses are taught on equipment that is obsolete by comparison — and will continue to be, since the schools hesitate to get new equipment. If you take computer courses in most schools, you have to punch cards which are fed into a big machine, and wait around for results, interminably. (One fine Eastern college still has an IBM 1130 as its main teaching computer — a machine which is about the equal of the new Radio Shack computer at $400.)

Okay, let's say your local community college offers a course for $30. But you aren't likely to have nearly as much fun in it as you will hacking and discovering with your own machine. If you have your own machine, you can try out anything you like and see instantly what its consequences turn out to be.

LEARNING TO PROGRAM: START WITH BASIC

If you want to learn simply what computers are about, all you need is a computer that runs BASIC and any kind of terminal at all — or a screen and keyboard. You don't have to know anything about what's inside the computer, and you can learn what programs really do.

Now you *could* begin by opening a manual, turning on BASIC, and struggling to build your own program.

A much more exciting way to begin is to load one of the many game programs which runs in BASIC, and playing with it, and then studying it to see how its separate parts work.

Since BASIC is the most popular language among hobbyists, you might as well start there. The rules of BASIC are reasonably simple. (They are given in hundreds of books and articles on BASIC, and hence omitted here.) One very good way to learn a computing language, especially BASIC, is to sit at the computer with a friend who already knows it, and try out his suggestions. Because he already knows the language, he can see what things you probably ought to know next, and make suggestions accrodingly. This is interactive learning.

One of the great things about learning a computer language interactively is that you can see the results right away. The machine helps you see the outcome, there is action, and perhaps noise. This makes it much more fun. It also makes it easy to see what mistakes you are making.

When you begin to program, the variety of feelings about it is phenomenal. There is the sense of power, of being able to command vast resources and energies with the flick of a finger, like Mickey Mouse directing the brooms in "Fantasia."

There can also be a sense of frustration, of not being able to make things come out the way you want, either because you don't know how to do a certain thing, or you're not sure if it can be done, or something's wrong with the machine, or you can't even find out which of the above is the case.

Or there can be a feeling of achievement, a feeling of triumph when your program works. People struggle against the dumb perversity of computers, putting hours or days or weeks on their programs, in part because the computers reward effort with care.

DON'T GET SIDETRACKED INTO HARDWARE (unless that's what you really want)

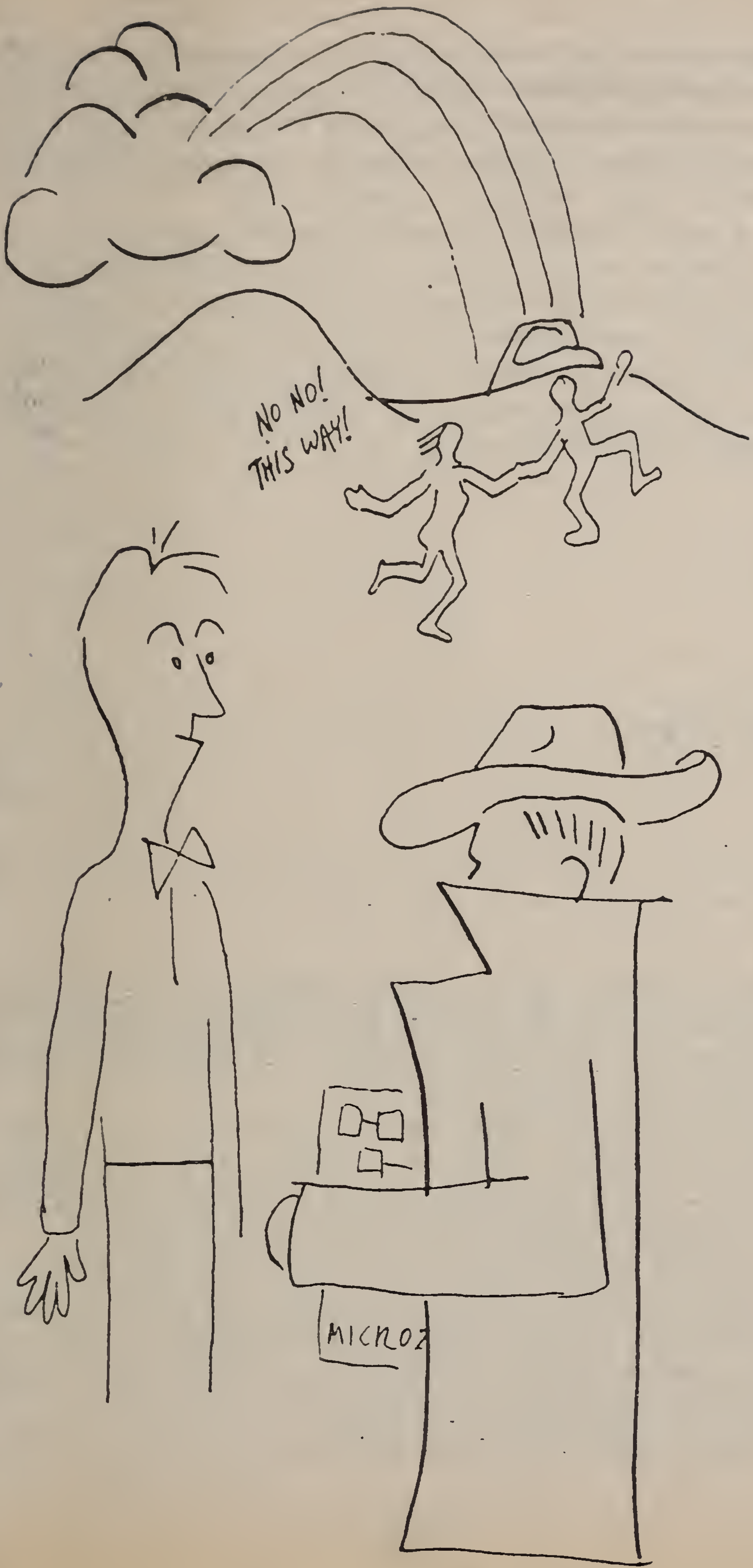
There are many electronic engineers who love only to talk about computers in terms of the way they are hooked together; and because of this obsession, these same people will tell you that the way to start in computers is to hook their parts together.

There exist a number of training courses and kits whose avowed intention is to teach you "all about microcomputers" — but on examination these turn out to be instructional materials concerned with the hookup of computer parts, and only the most minute and shallow forms of programming.

Indeed, probably, anything with "microcomputer" in the title is likely to be about either the wiring of circuits or bit-level programming. Either of these will take the beginner a long way from sophisticated applications. It's like learning about engines when you'd rather *drive.*

If this is what you want, fine — it's very respectable to know electronics. But to know hookups is *not* to understand what computers are really about. Don't let them talk you into electronics or bit-level programming unless that's what you really want. If you want to *understand computers,* start by learning a higher language, like BASIC or FORTH or PASCAL. That way you'll learn what it is to make computers *do* things.

Something else to beware of is the "one board computer" or "designer board" which contains a computer chip, some memory, and a way of getting rudimentary programs in and out. Such boards include the KIM-1 from MOS Technology, and the Mini-Micro Designer from E & L Systems. What these do is give instruction at the most detailed level, far from the higher programming concepts. (If you get serious about computers, you should eventually learn the lowest-level programming — but it's not the best place to start.) But you can't use them to learn programming with any breadth.


NO NO!
THIS WAY!
MICROZ

WHAT COMPUTER SHOULD YOU BUY?

Right now there are perhaps fifty brands of personal computer on the market. From the beginner's point of view, there are four kinds of computers: Prebuilts, Standards, Elevens and Others. Which to buy? It depends on what you want to do, and what you already know.

If you want to learn the principles of programming, or start in simple graphics, the matter is easy: you want a prebuilt. If you want to feast on the banquet of computer accessories now available, you will have to get into the hardware complications of the Standard computers. Only that way can you benefit from the spectacular variety of add-on hardware available for them.

PREBUILTS

There are presently six prebuilt computers being offered to the home market. All but two have appeared since April, and the number will multiply rapidly. (Addresses omitted for mass-market units.) They are:

The Commodore Pet, $650. This has a computer, keyboard, screen and cassette recorder, all in one jazzy-looking case. (Comes with system of interconnections standard for Hewlett-Packard and electronics industry.)

The Radio Shack computer, $400. This looks like a keyboard because *the computer chip is inside the keyboard.* You need a TV monitor and cassette recorder; you get to have them all together for $600. Comes with BASIC and 4K memory; additional 16K, $300.

The Sears Roebuck computer. No information at press time.

The Apple II, $1300. Color. This, too, offers the computer inside a keyboard, but a big one. It will hook to a *color* TV monitor or TV set. It also has a little sound-effects speaker inside. Comes with BASIC. (Apple Computer Co., 770 Welch Rd., Palo Alto, CA 94304.)

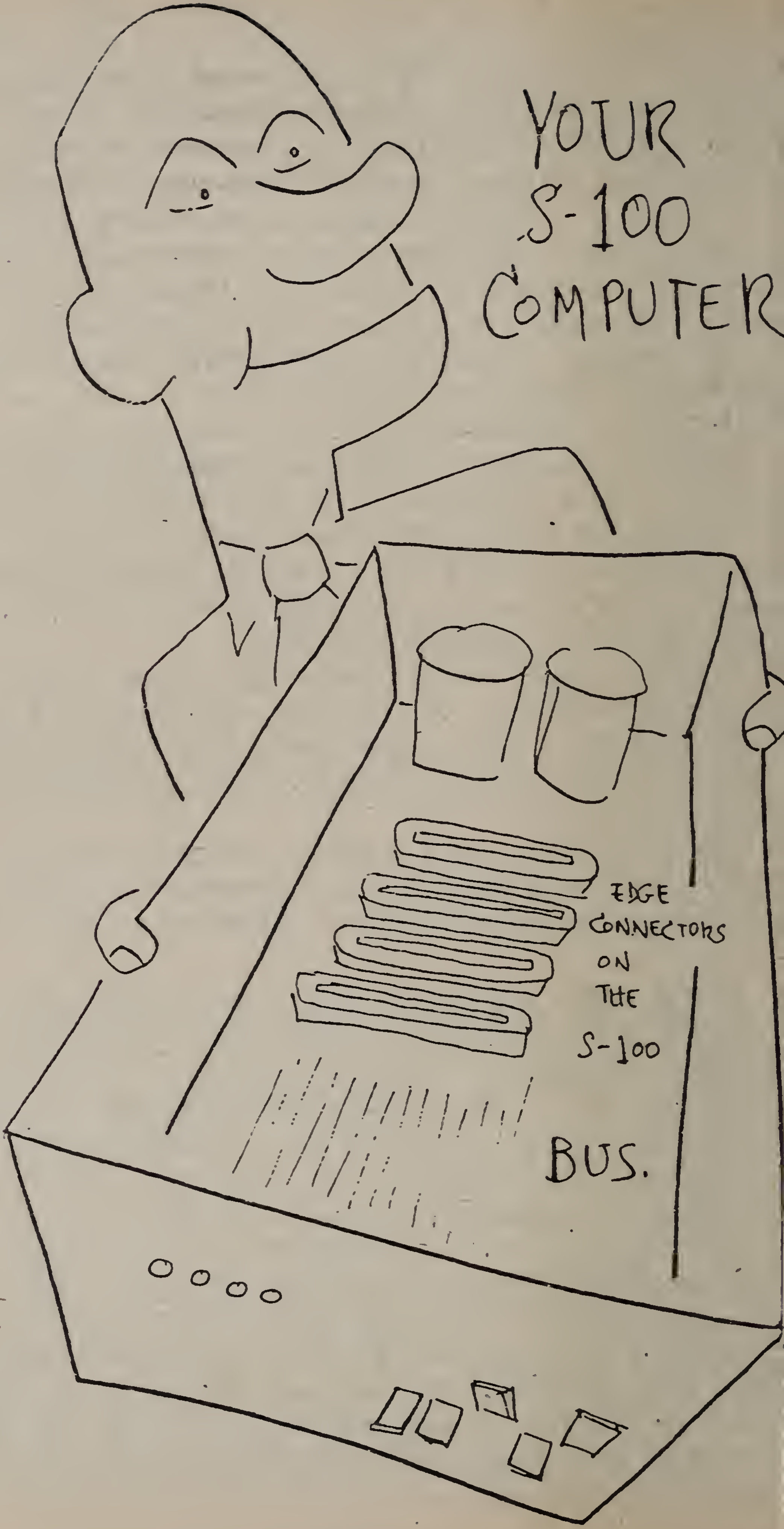
The Bally Home Library Computer, $300 plus $500 for keyboard. Color. Developed out of the video-game tradition, this is offered without a keyboard, but with *four pistol-grip pointers*. The keyboard will be available next year with additional memory, $500. This unit also offers color graphics — it hooks to your TV set — and a speaker. Comes with BASIC and 4K; additional memory will come with add-on keyboard.

The Compucolor ($3,000 from Intelligent Systems Corporation, 4376 Ridgegate Drive, Duluth GA 30136.) This is a unique unit from a small manufacturer. It comes with its own built-in color TV (nonstandard), graphics capability, and the ability to put more written material on its screen than any other common machine.

Another object on the market is almost certain to be revealed shortly as a computer. This is the Fairchild "Channel F," a video game costing $150 but containing a computer chip and suspicious circuitry indicative of imminent growth.

We may soon expect prebuilt computers from Timex, RCA, and Texas Instruments. Within a year there will be at least twenty manufacturers, geared up to sell an awful lot.

(IBM offers a small computer called the 5100, whose price is in the $5000 to $10,000 range. Out of fairness it should perhaps be mentioned among the prebuilts, especially as it is one of IBM's finest products, and some professorial types are buying it as a personal computer. It offers all the elegancies of the APL mathematical language, discussed elsewhere. However, it is quite unrelated in its scope of application to many of the home-computer applications that have been discussed in this book; it cannot be made highly interactive; and its cost is considerably higher than the other equipment discussed in this book. But IBM is to be congratulated on this product and encouraged in any possible way to become further involved with APL, which will prove a liberating force in many other areas.)


YOUR
S-100
COMPUTER
EDGE
CONNECTORS
ON
THE
S-100
BUS.

THE STANDARD COMPUTERS

The momentous Altair computer of 1975 established a standard form of interconnection. Many electronic engineers throw up their hands in horror; nevertheless, it *is* a standard. Dozens of computers, hundreds of accessories, are available in this form, and we will simply refer to them here as "standard" computers and accessories. (The official designation is "S-100," referring to the 100 lines of its interconnecting system.)

There are several reasons for getting a standard computer rather than a prebuilt.

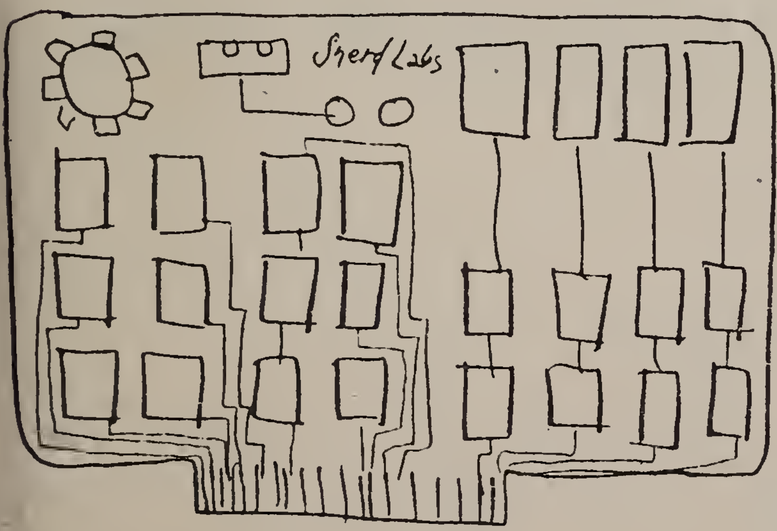
1. You like to play with hardware. In that case you are just a visitor in this book, and can find your information elsewhere.

2. You want to be able to get all those luscious accessories — the music synthesizers, and graphic displays and analog circuits and telephoning equipment and clocks and therometers and goodness knows what next.

3. You want to play with *software* at a fairly sophisticated level, and realize that you will have more alternative languages, and more exacting control over your programs, with prebuilts now available.

The main standard computers are the following.

(Some of these use the original 8080 computer chip, some use the Z-80. We won't distinguish these here. For all of these, except the SOL, you will need a terminal.)



Blinkies: ALTAIR, IMSAI, Z-1.

These are classic computers, with the blinking lights and switches and everything. There are a number of advantages to these at the machine-language level, if you know what you're doing.

ALTAIR: MITS, 2450 Alamo S.E., Albuquerque NM 87106.

IMSAI: IMS Associates, Inc. 14860 Wicks Blvd., San Leandro CA 94577.

Z-1: Cromemco, 2932 Charleston Road, Mountain View CA 94043.

Blankies: Z-2, XITAN, Polymorphic.

These are computers with a blank front and a switch. That means that when you turn it on — with the switch — there is already a permanent program that handles starting and certain other functions. Not as colorful, but for most purposes just as good. The Polymorphic is small but comes with graphics built in.

Z-2: Cromemco, already listed.

XITAN: Technical Design Labs, 342 Columbus Avenue, Trenton NJ 08629.

Polymorphic: 737 S. Kellogg, Goleta CA 94608.

SOL: This is both a computer and a terminal. Think of it as a blankie, with a switch, that comes with built-in keyboard, video text output and cassette interfaces. Very handsome, built-in keyboard, takes six S-100 boards. (Processor Technology, Inc., 7100 Johnson Industrial Drive, Pleasanton CA 94566.)

(The SOL is heartily endorsed by the author. After heavy use by students, his is going strong.)

SELECTED S-100 ACCESSORIES

The following items will give you an idea of some of the accessories available for the Standard computers.

PICTURES

The Levine board: a black-and-white graphics board of 256 X 192 dots is available for $500 from Itty Bitty Machine Company, 1316 Chicago Avenue, Evanston, IL 60201.

The Video Dazzler makes color pictures, 64 X 64, on a color video monitor. $300 from Cromemco.

A television camera input is available. It can be used by printers to scan art work or by kids to immortalize their funniest expressions. Environmental Interface 1, $595 including camera, from Environmental Interfaces, 3207 Meadowbrook Blvd. Cleveland OH 44118.

SOUND

Processor Technology, already mentioned, has the cheapest S-100 music synthesizer at $25. A four-channel music synthesizer is $159 from ALF Products, Inc., 128-S Taft, Lakewood CO 80228. The most expensive music synthesizer for the S-100 computers will also do voice and is pretty sophisticated. $525 from Logistics, Box 9970, Marina del Rey CA 90291.

The Computalker allows your Standard computer to speak English, of sorts. $400 from Computalker Consultants, P.O. Box 1951, Santa Monica CA 90406.

The Speechlab, $400. Allows your computer to recognize speech, carefully spoken, after a training period. From Heuristics, Inc., 900 N. San Antonio Road, Suite C-1, Los Altos CA 94022.

MISCELLANY

The telephone supergadget: permits your Standard computer to talk on the phone to other computers; allows it also to dial and answer automatically. International Data Systems, 400 N. Washington St., Suite 200, Falls Church VA 22046.

If you want to control all your appliances by computer, turning them on through the power cords (like electric timers), try Comptek, P.O. Box 516, La Canada CA 91011. It's expensive: a 32-line board for the computer is $360, but the individual outlets are $52 each. Similar equipment is also offered by Gimix, Inc., 1337 W. 37th Place, Chicago IL.

One firm has taken a digital wristwatch chip and put it on an S-100 board so that the computer program can always check the time of day and the date. There's even a battery, so it keeps track of the time when power is off. Computime, P.O. Box 417, Huntington Beach CA 92648.

Disks: floppy disks in both sizes from a variety of manufacturers. Hard disks from MITS and Alpha Micro Systems. For low price, note especially the $600 microfloppy from North Star Computers, 2465 Fourth St., Berkeley CA 94701.

IMSAI, already mentioned, has hardware that enables you to expand S-100 machines up to a million bytes of fast memory. (It will cost you upwards of fifty thousand dollars, however.)

A special bus adapter allows S-100 machines to use accessories made with electronic-industry-standard connectors, especially those of Hewlett-Packard. From Pickles and Trout, P.O. Box 2270, Goleta CA 93018.

PRINTERS AND TERMINALS

It should be noted that the most expensive things about your computer is the printer: and there are few decent printers, capable of doing acceptable business-quality typing, for under $3000. (IBM Selectrics cost less and type well but cannot be highly interactive.)

An important exception is the new Teletype Model 43 from Teletype Corporation, Skokie, Illinois, at only $1200.

UP THE CREEK WITH THE S-100

As we have said, S-100 machines offer the widest choices of accessories, opening possibilities of every imaginable form of experimentation. Your S-100 machine has thousands of different accessories available for it — memories, and graphic systems, and audio systems, and special hardware processors; but to be able to use them, you have to be able to pay the price. To begin with, you may buy your S-100 machine in kit form. This means you will spend weeks or months wiring it together, an occupation which does not bring you closer to understanding the programming of it, or the true nature of computers. Worse, most builders don't do it right the first time, so subassemblies may have to be shipped back for checkout, and all that.

Setting it up for anything involves complications. Even if you skip low-level programming, you'll have to familiarize yourself with the address space of the 8080, and how to allocate it, and how to set up the separate input and output ports.

A very strong and important criterion is: can you get it repaired around here?

In this, buying a home computer is like buying a car. It does not make sense to buy a brand of automobile whose dealer is far away. There is a strong case for getting the kind your computer store sells and repairs.

Finally, unlike the prebuilts, the standard machines also put you on your own for repair and adjustment of your own hookups. (This is called the "interface maintenance" problem in the straight computer field.) Here your computer store may or may not help. Talk to them about it a lot.

ELEVENS

It is widely agreed, by a large proportion of those who understand computer architecture, that the best computer in the world is the PDP-11 from Digital Equipment Corporation. But while the PDP-11 is probably the best computer, that does not mean it is the best to have. Just as the beginning photographer may get more satisfaction from a simple Instamatic camera than from a $1,000 Nikon system, with all its adjustments and complications, the beauties of the PDP-11 are especially for those who wish to work programming it at the innermost level of machine language. To use BASIC, you don't need an 11. But for the most sophisticated users, who want languages like BLISS and LISP and MUMPS, the PDP-11 probably offers — through the Digital Equipment users group — more powerful languages than any other machine.

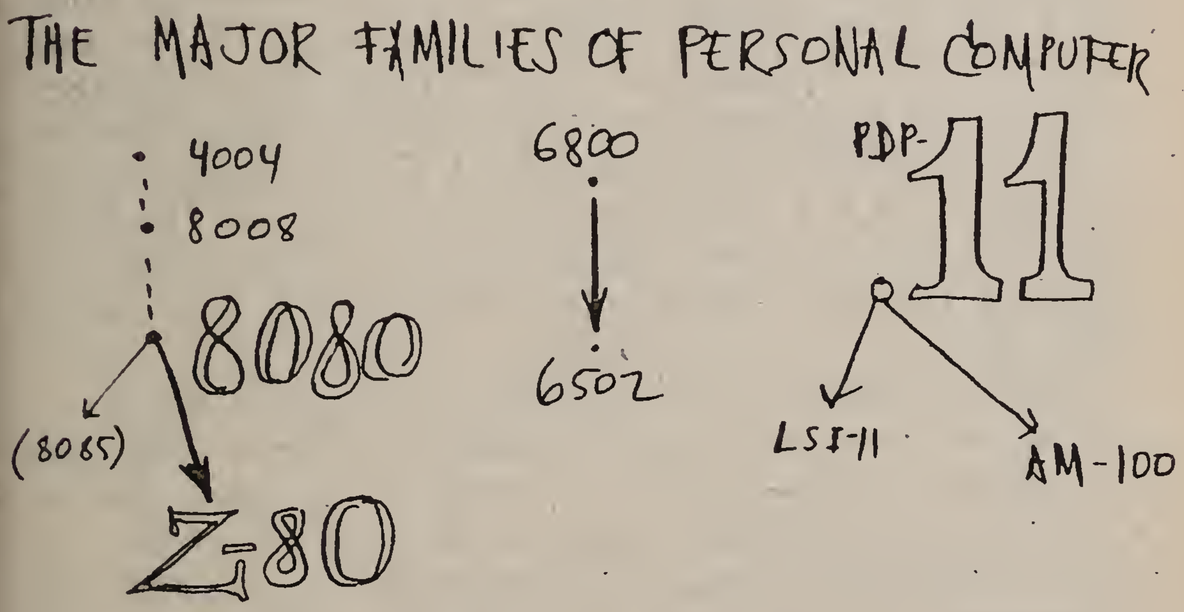
The PDP-11 now comes in small versions which some personal users may be able to afford. Several Different versions of the PDP-11 are available from Digital, but we will consider only the smallest and least expensive model, called the LSI-11. The fundamental electronics of the LSI-11 are contained on a single card, about a foot square, listing for just under a thousand dollars. But this is incomplete. It needs a power supply, cabinet, and much more; Digital will sell it to you complete, under the name 11/03, for about $3000. This is too high for this book.

But as of August 1977, the PDP-11 is now available as a Heathkit from the Heath Company, Benton Harbor, Michigan. The price of the kit, including the basic board, power supply and certain other gear, is $1350. This is about a thousand dollars less than Digital's version.

Health has a very cooperative arrangement with Digital, whose ins and outs are fairly intricate. Heath's LSI-11 package is about the same basic hookup as that sold by Digital. It seems like a bargain — but it turns out that the accessories for it must be purchased from Digital Equipment Corporation. No saving there.

A special advantage to buyers of the LSI-11 system, from either Digital or from Heath, is that a big library of software — programs which have been stored by 11 users in Digital's program library, DECUS — is available to them. (Cynics might point out that much of what DECUS offers is uninteresting to hobbyists.)

There do exist alternatives to Digital's monopoly on accessories for the LSI-11. One is to use the hookup supplied by a firm called General Robotics, which allows the LSI-11 to be hooked to the regular PDP-11 bus structure. This means that the regular accessories for the PDP-11, available from dozens of sources and hence cheap, may be used.



UNIX

Probably the best operating system in the world is the master control program made for the PDP-11 by Bell Laboratories. It is called UNIX.

It is expensive, but you can buy it for your home PDP-11 for a price in the neighborhood of $4000.

(The phone company is plainly operating outside the law in selling computer software, but no rivals would sue, since then they would merely *give it away*, ending the sale of most other PDP-11 software.)

THE AM-100

Because Digital Equipment has patented the PDP-11, and apparently been able to make that patent stick, they have foiled competitive companies attempting to imitate the PDP-11. Several firms have offered machines which were copies of the 11, only to find themselves trounced by lawsuit.

An important exception seems to be a firm called Alpha Micro Systems, which is manufacturing a computer very like the PDP-11 for the S-100 bus. This means that if you are a fan of the PDP-11 structure, as thousands are, you can run PDP-11 programs on your personal computer — and still use the S-100 accessories. (Alpha Micro Systems, 17875 N Sky Park North, Irvine CA 92714.) A special advantage of the AM-100 is that it can be used with the S-100 bus, and thus may avail itself of all the extraordinary accessories that have come out for the Altair and similar computers.

There is, however, a catch. Programs written for the 11 will not run on the AM-100 in their final form; they must be adapted slightly. And according to some arcane agreement with Digital, there is no way to convert the final, tiniest-step program from the PDP-11 to the AM-100.

However, if you have the PDP-11 programs in the original form they are written in — the source code — a number of small conversions will supposedly make it possible to run them on your AM-100. So for those who want to program at the bottom level, the AM-100 is an important alternative to the pure 11.

Another catch with the AM-100 is the price. It is only sold with double floppy disks, and it requires a certain kind of memory board, the so-called "static" memory board, which is considerably more expensive than the regular ones. Hence a complete system presently costs $6000. However, they are going like hotcakes.

Another very important advantage of the AM-100 is its operating system: those programs which are supplied to make programming more useful. The AM-100 operating system is a creditable imitation of the excellent operating system of Digital's great PDP-10 computer.

Finally, whether Digital's software library will be open to AM-100 users is not clear at press time.

THE GENERAL INSTRUMENTS CHIP

A computer chip bearing a curious resemblance to the PDP-11 structure was announced some time ago by General Instruments. Sometimes it seems to be available, sometimes not. If it really is an eleven, and cannot be quashed by DEC, it should become a hot item.

THE TI 9900

Texas Instruments, that formidable manufacturer, has a computer chip that is also close enough to be lumped with the Elevens. This is the TI 9900. Its structure, however, involves certain crucial changes to evade DEC's patent. Eleven purists won't want it, but those interested in quantity production and low price may well.

However, potential manufacturers of home computers may be reminded of how Texas Instruments, after selling many calculator chips, brought out calculators themselves, to the great disadvantage of some of their former customers.

OTHER MACHINES

To the beginning user, systems which are neither prebuilt nor standardized are probably of little interest. This is not to derogate their manufacturers, but to face facts. Many companies, seeing the S-100 bus, have scoffed and said, "We can do better." And off they have gone to do so, scarcely realizing the part that standardization has played, and must always play, in the computer industry. Four or five companies, then, offer hobby computers outside the standard system of interconnection. And they have gotten themselves into a dead end. Undoubtedly their interconnections are designed better than those of the classic family; but it's like the Land Rover over the Chevrolet — hardly sensible for the average user.

Earliest to buck the Altair trend were the Sphere Corporation and Southwest Technical Corporation. Sphere took a financial beating, though is reported to be still hanging on, but one must suspect that standardization would have helped their position. Southwest Technical Products is still hanging in strong with their well-respected 6800 system. This unit is of some interest, in that a few accessories are being marketed for it by other makers; its interconnection system is now called the "S-50" bus. (The household control circuitry of Gimix, Inc., is made for this machine.)

Digital Group, of Denver, offers a line of computers highly regarded for their basic engineering soundness. They also offer a low-cost cassette storage unit of the all-digital type, which is clearly superior to the audio recording used by most of the other cassette systems. For electronics fanciers this is a fine system.

Another interesting challenger is Ohio Scientific Instruments, of Hiram, Ohio. They too denounce the S-100 bus, have chosen their own system of interconnection and offer some interesting things.

YOUR COMPUTER STORE

Your computer store — you *do* have one, almost anywhere in the USA — if you don't mind driving, that is — your computer store, anyway, gives advice, sells equipment, and helps you with repair. They also may sell prebuilts, possibly even prebuilt S-100 machines, assembled in the back room and guaranteed by themselves. There you can see the machines in operation, daydream with other customers, or get advice from the staff.

An important function of the computer store is to provide free information; and an attitude of general helpfulness without regard to sales, with frank answers, appears to be a widespread norm. Naturally they will try to sell you the computers they prefer to sell, but in general the computer stores seem to be fairly straightforward and honest about facts and advice.

They have to, to establish a reputation for integrity; because the hobby grapevine is ferociously efficient.

Your computer store will do repairs, but expect repair work to be expensive: good technicians cost some twelve to twenty dollars an hour. This is because there aren't many of them and they are in demand.


YOUR FRIENDLY COMPUTER STORE
YOU WIN

HOBBY COMPUTER MAGAZINES

Herewith your main hobby computer magazines. More are starting constantly.

1. BEGINNERS' MAGAZINES

ROM: Computer Applications for Living,
Route 97, Hampton CT 06247.

Creative Computing,
P.O. Box 789-M, Morristown NJ 07960.

People's Computers,
1263 El Camino Real, Box E, Menlo Park CA 94025.

Personal Computing,
167 Corey Rd., Brookline MA 02146.

2. HARDWARE MAGAZINES

Byte,
70 Main St., Peterborough NH 90701.

Interface Age,
P.O. Box 1234, Cerritos CA 90701.

3. A SOFTWARE MAGAZINE

Dr. Dobbs' Journal of Computer Calisthenics and Orthodontia,
Box E, Menlo Park CA 94025.

THE ART OF THE COMPUTER SCREEN

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 256x192 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, 192x160 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 book! 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 intereactively. 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 genious 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, throughtful complex on the outside of Boston where they do electronic 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 SKET-CHPAD.

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 "rubber-band 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 abstracted 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 light-pen. 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 light-pens 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.*

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 Engelbart 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, mentioned earlier. 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.

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.

* See T. Nelson, "Getting It Out of Our System," in Schechter (ed.), *Critique of Information Retrieval* (Thompson Books, 1968).

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.

Never mind about the different *kinds* of screen; that you can find out elsewhere. This is a non-technical book.

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 light pen 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 light-pen 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 light-pen 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 labelled, "Play It Again, SAM."

And as the computer re-plays 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.

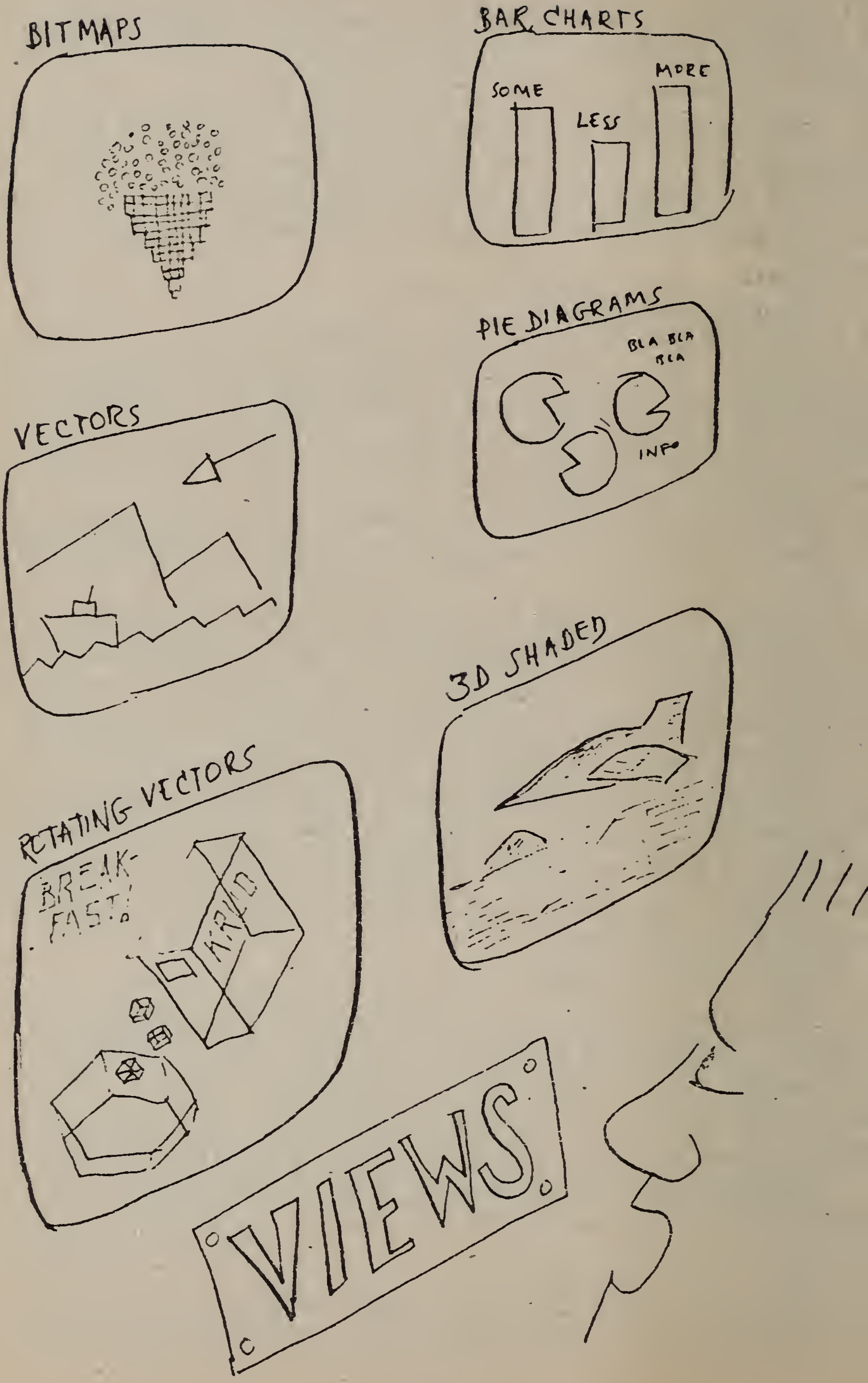
But wait! Why be satisfied with numerical tables? The graphical screen can be easily be programmed to give you bar charts, pie diagrams, diagrams in proportional shades of grey. Or even new kinds of diagrams that can be rotated in multiple dimensions, presenting to the eye things you could never see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


BITMAPS
BAR CHARTS
SOME
LESS
MORE
PIE DIAGRAMS
BLA BLA BLA
INFO
VECTORS
3D SHADED
ROTATING VECTORS
BREAK-FAST!
VIEWS

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 is 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 diagram 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 is 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 highschool 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 tattooed on everyone who designs consumer products.

Many engineers and technicians have clamed that this can't be done. Balderdash! It is merely difficult. Moreover, it takes intense dedication to clarity, and repeated revision and re-thinking. 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, and that is why they are technical people;* and so 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 book 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.

---

*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.

FANTICS OF DESIGN

Actually, what is needed is an entire different idea of design. The design of something today should be principally the psychological design of how its controls will appear and feel.*

The central issue is really one of *fantics,* which is the art and technology of showing things to people, and making things clear. This field stretches across quite an expanse: at one corner is *writing,* which is trying to put ideas and feelings across to people through words; at another corner is movie-making, which is trying to put ideas (and feelings) across through pictures. Creating diagrams, giving directions, setting up exhibits, putting up signs, setting up stores and campgrounds — all these are activities where some kind of ideas and understandings have to be communicated to people, either explicitly or by suggestion.

The fantic problem, then, is getting something across, either by what you say or what you suggest.

But to make it easier, the thing to be communicated had better be simple and uncluttered. And so the fantics of *design* is how to design things whose function *can* be easily communicated.

In this new era, simplicity and clarity and ease of use will become important as never before. These matters have always *been* important, but buyers haven't known it; they have bought what had the most shiny buttons, rather than considering whether they would eventually be able to learn how to use the thing.

But that ends now.

The buyers who have supported Kodak, those purchasers of Instamatic and Box Brownie cameras, have had good reasons for their choice: simplicity and clarity. And these principles must be served if people at large are to be served by computers.

---

* This study is often lumped under a heading called "human factors" or "human engineering." This would be all right, except that it seems well-focused when it isn't. Most people who do "human factors" measure shinbones and reaction times and the like. A new name helps to focus attention on what is really important.

CLARITY IN PROGRAMS FOR PEOPLE

People are going to buy computer programs the way they buy cameras, for their personal use.

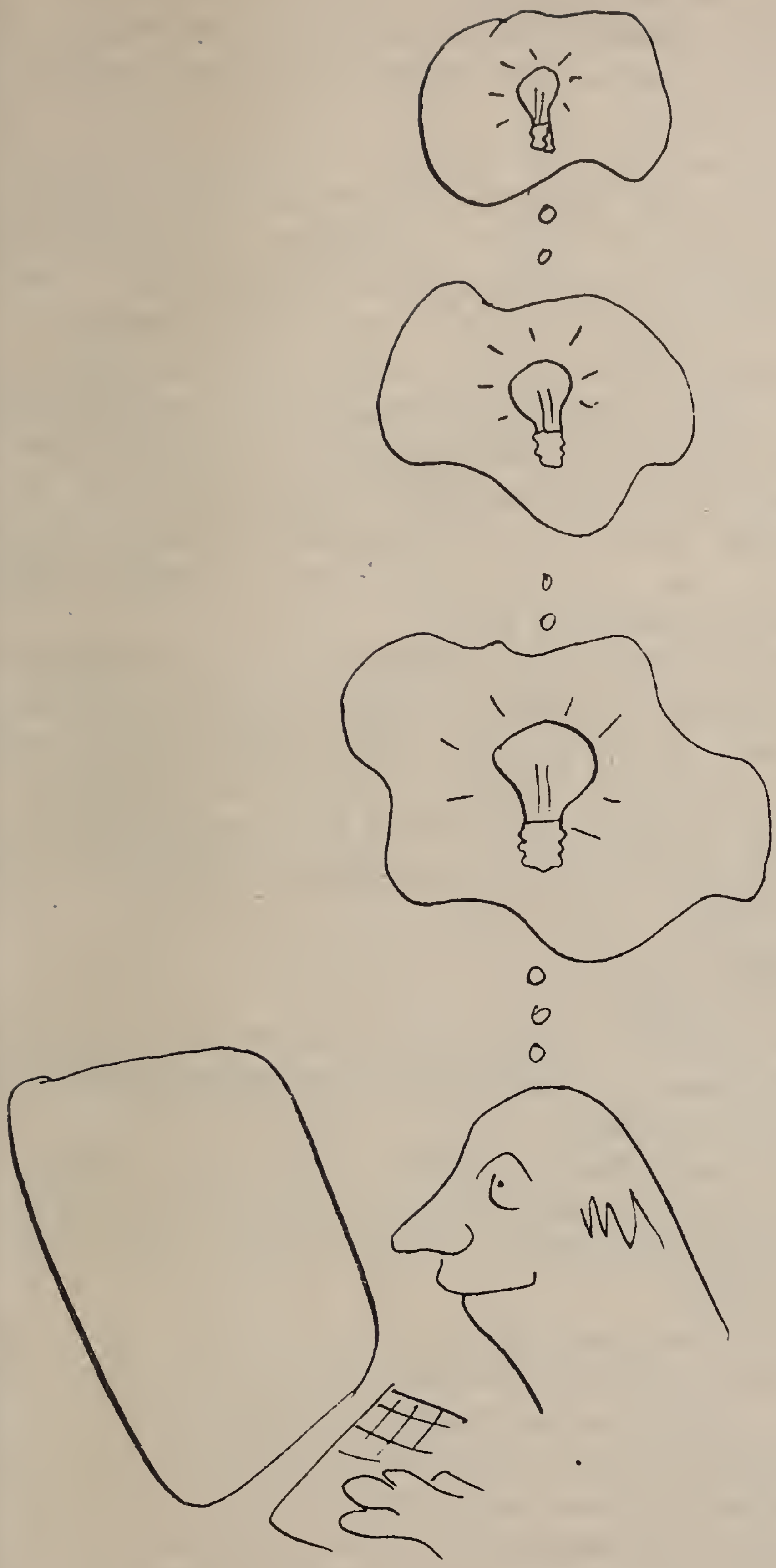
But computer programs are not shiny, and they do not have buttons and knobs. Either they come in a cassette, in which case they all look alike, or they come in some other inscrutable form that has nothing to do with their purpose.

So their outward appearance does not matter at all; they have to do what's wanted in a comprehensible way. Each impediment to understanding will stand in the way of sale. They *have to be simple and clear.*

There is a natural limit on the complication of physical equipment. It is hard for even the worst designer to introduce more confusions than there are buttons on the box. And if purchasers do not understand a mechanical system at all, they usually do not purchase it, so that presents a final limitation.

But the number of complications in a computer program can be infinite. There is no limit. Till now, computer programs of unspeakable complication were purchased in the corporate world. But individuals won't get taken — at least not to that degree.

There is a myth that things which are simple and clear are not powerful. This is ridiculous: combining simplicity and power is the problem that now confronts us.



If you and the computer are merely typewriting, the effect can be exciting enough. But there are many ways that the computer can make pictures, and that these pictures can respond to our actions. This will come into our lives for recreation, for business, for literature, for art, for every possible thing we do.

Here on the screen you can see a schedule for all the television shows you might like to watch this week. You point at the ones you think you'd like to see, and the system automatically makes note of them. Later, it will turn on the TV automatically at the times planned.

Here is a cartoon character — say, Irving the Elephant. He is looking out at you on your computer screen and his talk balloon says, "Well, what should I do next? Try to catch the crook, go after the girl, or go swimming?" You now point at the words on the screen which represent your decision, and the cartoon character directly begins whatever you commanded him.

Here on the screen is the beginning of a book: say, *Alice in Wonderland*. By pointing, you may choose to see it in a plain type-face or in fancy typography with flowing illuminations; with the original Tenniell illustrations, or with animated cartoons illustrating the story; or with annotations (like those assembled by Martin Gardner for *The Annotated Alice*).

Simply point?

What is this magic?

The answer is that when the computer is set up to show things on its screen, its program can also sense *what you point at* on the screen.

So that in any situation where you have a choice to make, the simplest way to make that choice is simply to have the alternatives listed on the screen, and point at what you prefer.

Now what do we mean by "point"? Well, it depends on the setup. Some computer screens actually allow you to use your finger, poking the screen at the position you choose. But such setups tend to be unreliable. So most systems that allow pointing give you some sort of a tool — going by names like lightpen, joystick or mouse. We won't try to explain the differences here; when you get to use one, you'll see how to use it in seconds anyway.

The computer's response to a choice is called *branching.* Branching illustrations and cartoons, allowing people to explore new works of art and writing, offer limitless possibilities which have scarcely been explored at all.

Your graphic screen can be used for home study of the freest kind. Suppose on the screen you see a picture of the interior of the human body, with different internal organs labeled. Suppose now you can point at the one you would like to know more about — the heart, or spleen, or pancreas — and get an explanation of it with animated pictures.

Or say you are lost in a store. But here, next to the escalator, is your friendly computer screen! You point at the department you want to go to, and a map appears, with a little cartoon character — say, Howard the Duck — waddling along a dotted line that takes you to the department you want.

So far, interactive screen systems, with pictures and branching, are only to be seen in the video games. But tomorrow, as businessmen and consumers begin to realize the possibilities, they will be everywhere, for doing everything.

Computer screens on the kitchen table. Computer screens by the bedside. Computer screens on the office desk. Computer screens on the school desk. Computer screens on automobile dashboards, Coke machines, ticket dispensers. (In a few years, it will be cheaper and simpler to put the directions for a new machine on a computer screen, built onto the side of the machine, than to print them on paper.)

Each activity, each form of work, will of course require programming to make the wonderful new environments. And this will require a new kind of artist: someone with vision first, and the technical understanding to carry the vision through.

VIRTUALITY: THE WORLD BEYOND THE SCREEN

Everybody knows what "reality" is supposed to mean, though we may disagree over what's out there. But "reality" generally means the nuts and bolts, the solid metal and dirt, as distinct from ideas and feelings.

Very well. An important opposite of "real" is *virtual*. It means *as-if*. ("In essence or effect, although not formally or actually." — Oxford Universal Dictionary.)

Virtuality, then, is the *seeming* of a thing, the ideas and impressions and feelings you get from it. The virtuality of a magic show is the illusion of doves from a hat, or a sawn woman walking away intact. The virtuality of a Cadillac is a cushy drive and the sense of luxury; the virtuality of a movie is the world it puts you in.

The important thing about computer screen systems is their virtuality. You do not care, as a user, whether they work by transistors or by rice pudding. You do not care whether the screen is on the end of a big vacuum tube or a sandwich-panel of neon.

You care *what it's like to use it*. That is the virtuality.

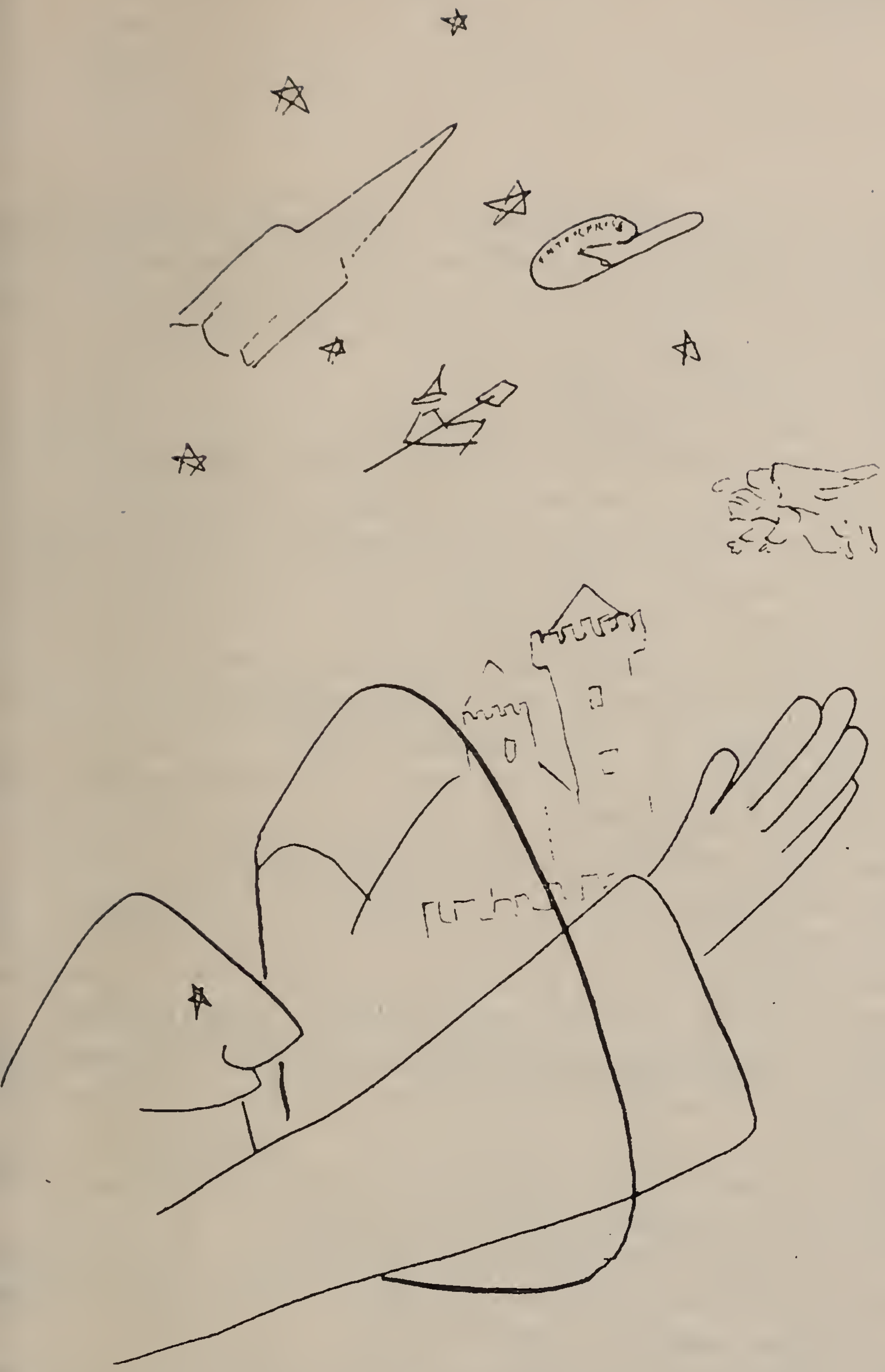
The virtual structure of a screen-system is what structure it seems to have.

For instance, the virtual structure of SKETCHPAD.

SKETCHPAD was a facility where you could make master drawings, and combine copies of these drawings of any size. If you changed the master drawing, all the copies would change correspondingly.

The virtual structure of SKETCHPAD, then, was of a space which could be stretched to any degree on the screen, and instances of pictures that could be copied.

This is the virtuality of SKETCHPAD, or at least that part of SKETCHPAD we have described here. The kind of computer it ran on was not significant to its virtuality, nor was any other feature of the computing hardware. Its virtuality was what you could do to the stored pictures through the screen, and how it felt.



Designing Screen Systems: VIRTUALITY FIRST, TECHNICALITIES SECOND

When a technically-minded person creates an interactive computer system, he generally decides first *how he wants it to work internally*, then considers its external details one by one, as if they were superficial and cosmetic.

This is entirely wrong.

The way to design a screen-system is like making a movie: decide first what is to be its virtual structure, what are to be its basic concepts and performance on the screen, then devote technical effort to making that come out right. Only this way will it have the conceptual clarity, or the feel, that you wanted to put into it.

For instance, suppose you are programming a simple system to let users write and revise their text on a screen. (A "text editor" or "word processor.")

This is a common sort of program, and most programs of this type are abominable.

The way a typical programmer goes about it is this.

First he decides how the text should be stored, based on what's easiest for him.

Then he figures out a convenient way for the program to make insertions and deletions within this stored text.

Then he figures out some way for the user to command the system to insert or delete; based again, usually, on his convenience.

Then he puts it all together and gets it working, and there it is — another lousy text editor.

(What about rearrangements within the text? "That can't be done," he says. Meaning that he didn't feel obliged to make the program deal with it.)

The *right* way, of course, is to think; what sort of structures of text might a user want, and what kinds of operations upon them? With a lot of thought, you might very well decide that plain sequential text is inadequate. What virtuality then?

(For more on this subject, see *Computer Lib.*)

It's much like making a movie. The movie-maker begins by deciding what story to tell, and what overall quality the finished film should have. Whether it should take place in ancient Rome, in a haunted house, or on another planet; whether it should be cheerful or sinister.

Then plans are made for how to make the film: what scenery to build, what to film on far-away locations, and so on. When it appears that one method is too expensive for a given effect or scene, another is chosen.

The same goes for interactive computer screen systems.

Now the design of screen-systems has many complications. Some tricks on the screen can be done easily by computer programming, others cannot. Some things can be done rapidly by the computer, along with other programs, on the side; other things will take all the computer's time.

The design of screen-systems requires that you know what effects you want, and take whatever steps you must to get them. When something doesn't work, you see what else is possible; but always, like the movie-maker, press toward that structure and quality you want to achieve.

Virtuality consists of both the conceptual structure of a thing, and its feel. The conceptual structure must, for the most part, be planned out first; the feel can usually be fine-tuned later. But all aspects of virtuality should be considered from the beginning of the design process.

CONTROL STRUCTURE AND VIRTUALITY

The controls by which we operate something have a virtuality, a seeming: they feel a certain way, and they make the things you are operating on seem a certain way.

For instance, the control structure of a car — with automatic shift — is basically very simple. You go or you stop; and you turn left and right when needed. Forward, sideways, sometimes backward; and that's it.

But the controls of the car do not reflect this. You have one pedal to go, another to stop; and only gradually, as a driver's skill grows, do starting and stopping become in the mind a unified continuum.

In other words, the separateness of the controls promotes a conceptual separateness which is only gradually unified in the mind.

It's the same with all machinery. The skilled user is someone who has, in his mind, learned to see all the separate pushings and turnings of the equipment as parts of overall movements and intentions.

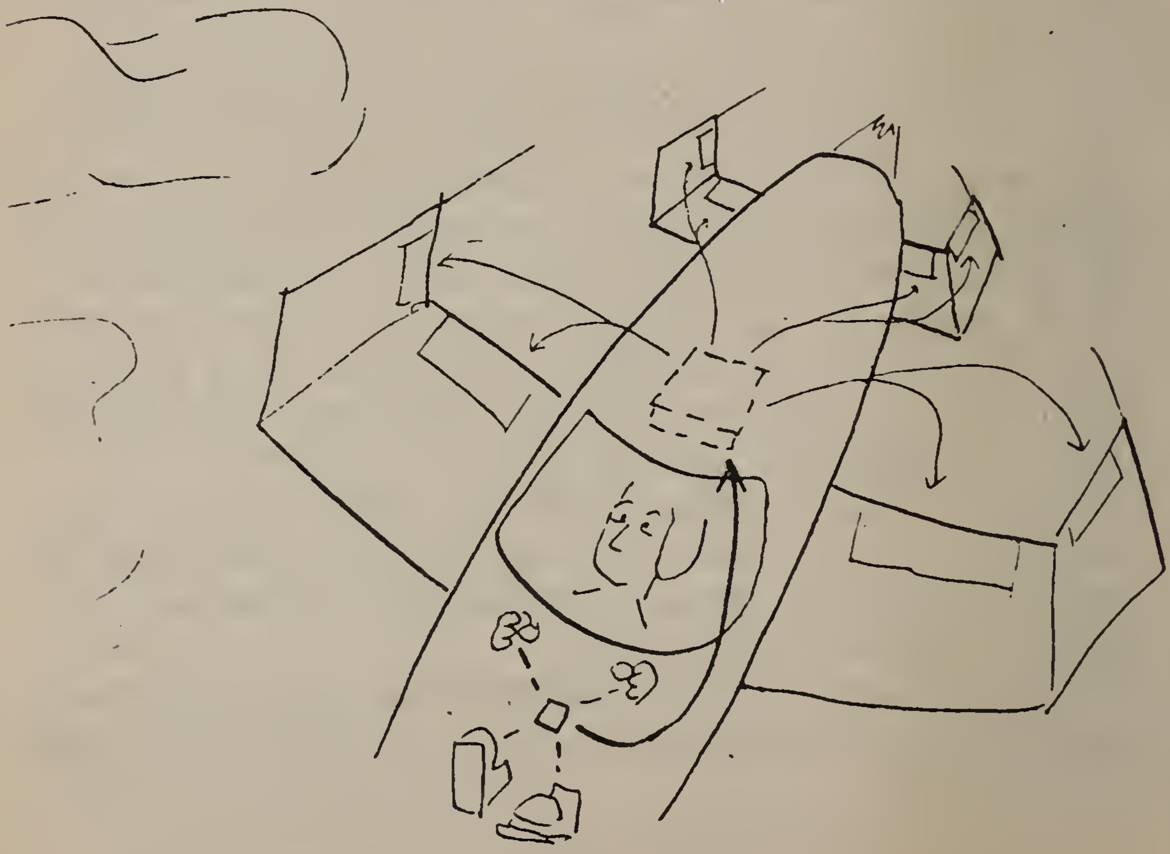
It follows that we can make people skillful much faster, and less likely to make mistakes, if we think out controls to be conceptually unified in the first place.

That is the design of control virtuality.

Example: the pilot of today's aircraft does not directly control the individual flaps and throttles by hand. These separate actions are taken by an "avionic" computer. The pilot's controls are designed as *mentally unified* operations; the signals from these controls are translated by the computer into the separate flappings and turns.

(The pilot of the F-111 can simply set his plane to follow the terrain at any given height — with a bumpy or smooth ride.)

This kind of conceptual control design is what we will be doing for tomorrow's screen computer systems. Control of all other machinery and information will be handled, not with buttons and levers anymore, or the way laymen think of "computers," with numerical codes; but with clear, elegant pictures and diagrams and texts on computer screens.



HOW COMPUTERS SHOULD ALWAYS BE USED

The true meaning of interactive screen-systems is that people can do things easily and without confusion. Sitting at screens at home or the office, they will type — or point — and the system will respond clearly, with a clear virtuality.

Such systems should always involve *text*, to explain what to do, ask questions, present information.

Such systems should always involve *pictures*, helping to clarify and visualize, adding fun.

Such systems should always be *highly interactive*, so that each time you press a key the system responds at once.

If these standards are correct, it follows — paradoxically — that *IBM computers can never be used as computers should be used.* This is because IBM computers cannot ordinarily be set up to respond to each stroke of the user at the keyboard. Whatever key or keys the user wants to press must be followed by another nuisance pressing, either the key called "Carriage Return" or "Attention." This makes highly interactive systems totally unworkable on IBM computers. (Xerox's Dynabook system and the PLATO system now being sold by CDC both have avoided this, and will respond to any key.)

THE COMPUTER MEDIA

In an earlier chapter we considered the different machines and instruments which have caught on in the consumer market in the last century. The biggest have been the automobile, the telephone, movies, radio and television. Computer media will have an impact comparable to these big five.

The new computer media, however, will uniquely combine elements of all of these: the visual entertainment of reading and of television; a personal environment comparable to the automobile; and the personal intercommunication of the television. The computer media on tomorrow's screens will include text and visual material, animation, and branching alternatives.

Text describes or explains or narrates. Pictures illustrate, provide a mental framework, give atmosphere, decorate. Making the pictures move — animation — lends understanding, emphasis, a sense of action, heightened involvement for the user. Or it tells a story.

But we have not yet worked these things out.

In the book, movie, radio and television, forms have gradually been discovered for organizing and segmenting the material, for orienting the user's thought, for creating continuity within a work, and for keeping up consumer interest between works. The same thing will happen for the computer media. We will be discovering and inventing new presentational materials for some time to come. We will be discovering the viable forms, structures, organzitions, continuities, segmentations of the new computer media.

Besides plain programs and games, we will soon see works for reading and visual exploration, adventures and stories for the screen that are like movies, cartoons and comic strips; and interactive diagram-wonderlands. All of these, of course, will branch, allowing the user to make a variety of choices as he goes along.

But the real precedents for what is going to happen with computer media are not to be found in the 19th or 20th centuries. The real precedents are the printing press, and before that the spread of writing itself.

INVENTING COMPUTER MEDIA

Many separate screen techniques have been invented by the pioneers of computer graphics, especially Sutherland, Knowlton, Baecker, Kay, and a handful of others. (See *Computer Lib* for more details.)

But these separate techniques are not the same as a structured system of media.

This can only be explained, right now, by analogy.

When movies began, they did not have closeups, or any editing at all. They did not have pans, dollies, cross-cutting, mattes, double exposures or zooms. These evolved.

When radio began, they did not have station breaks, theme songs, announcer transitions, musical bridges, or time slots. These *evolved.* As did commercials, jingles, and other mixed blessings.

When television began, they did not have the above elements, or the voice-over, holding visuals, visual transitions or anchormen. These *evolved.*

In all these media, too, the units of presentation — the shows and programs — developed gradually as a genre.

In other words, there is strong reason to suppose that the computer screen media will evolve in similar ways to a system of structures, units and transitions.

However, because interactive computers include pictures, sound, and (implicitly) all the other media we have just mentioned, we may expect a higher virtuality to appear that combines these other elements and weaves them together.

*The author has conjectured that a standardized form of storage and interchange will evolve for works of this type — involving text, pictures and branching. See T. Nelson, "A Conceptual Framework for Man-Machine Everything." Proceedings of the 1973 National Computer Conference. So far, though, there are no signs in this direction.

A NEW KIND OF MENTAL LIFE

Some have thought that in the new age computers would order us about. This, as we have seen, is fiction: though a useful fiction to those people who want to order us about. At the other extreme are those who think that computers will suddenly become a forest of intelligent companions, and that you or I will roam among them, receiving their tutelage, like Mowgli the Jungle Boy. This is an interesting idea that requires much more demonstration before we can take it seriously.

Between these two extremes — oppression and companionship — is the more sensible possibility. It is this: that computers, by helping us juggle and feel our way through information, will help us to find new thought processes, new understandings, new ideas, new learning, new skills.

HOW COMPUTER PEOPLE THINK

Some time ago, when I was a college student, I got to know a boy of twelve from Georgia. Our worlds were foreign to each other — he was interested in firearms and fishing, I was interested in a lot of different things. Or so I thought. But he complained that I had only one interest. "It's all *books*," he said to me in confused amazement. "Everything you're interested in is in *books*!"

I did not consider this true at all, except insofar as somewhere, in some book, there is something relevant to any subject. Yet to him this made it all one.

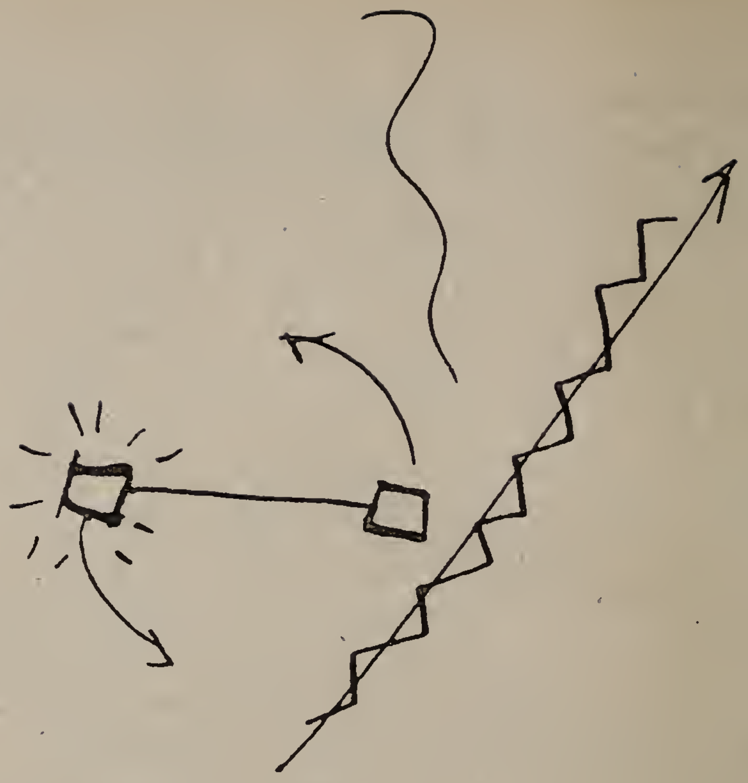
Well, I have thought about this encounter for a long time, since it exposed such a peculiar intellectual gulf. The "books" merely represented to me a form of transmission and an aid to thought. To him the books themselves seemed a fundamental force, an obsession.

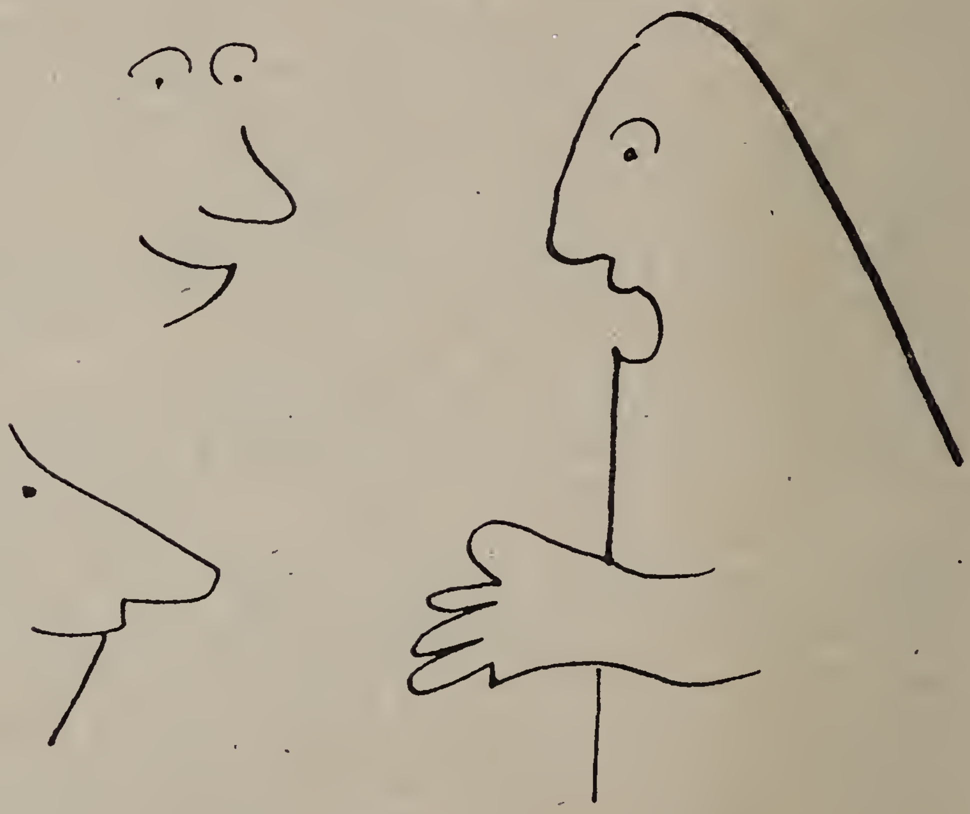
In much the same way, computer people feel themselves deeply misunderstood by civilians. Because computers figure in the thoughts and conversations of the computer people, outsiders think all the thoughts and conversations are *about computers*. This is not the case. Computer talk is not necessarily about computers. Once you have become interested in the computer, it enters all activities and all zones of thought, both as a tool and as a source of new concepts. In other words, because it is such a general device, it has assisted in our understanding many *general issues* as we never have before.

And so the conversation of computer people is sprinkled with terms which outsiders think are about foreign and mystical things, but which are in reality an incisive way of understanding the outsiders' own world.

Just as the computer is the most basic machine, and in its simplicity is adaptable to all purposes, so many computer concepts are likewise adapted to a great variety of subjects, and naturally enter the thoughts of any computer-experienced person when he goes to look at that subject.

The vocabulary of computer people allows them to speak to each other intricately about the generalities of the computer world: about arrangements in time and space — scheduling and where to put things; about organization, interruption, confusion, dissent; about method, exploration, uncertainty, muddling through.





The special vocabulary of computerdom allows you to say these things more precisely than in ordinary life. But this also means you can speak about *ordinary life* with either a precision, or a metaphor, that was impossible before.

Indeed, this is what has already happened in many of the learned fields where people have begun to use computers.

In linguistics, which a hundred years ago was a curious amalgam of multilingual eccentrics, we have seen the computer contribute surprising new insights. Noam Chomsky discovered in the 1950s that language can be seen as a way of expanding and combining expressions in ways that can be precisely formulated. From this discovery, the study of linguistics abruptly became a race to create computer programs to manipulate natural languages. Researchers attempted to decipher and generate, by computer, real sentences.

This quest has turned out to be much more difficult than people thought it would be. And if no computer systems capable of full language behavior have come into existence, yet the search for language mechanisms has shown us what language is really doing.

The linguists' new view of language is of a sweeping system of processes — ways of combining phrases, and their complicated coupling structure.

The new understanding of language, then, is an increasing understanding of what kind of mechanisms can carry out the language process.

Similarly, in psychology many insights have been gained from the examination of rational activity, and mimicking it with computer simulation. The computer, programmed to muddle through various reasoning problems, shows us how very many steps and inferences the problem really takes. We have come to see, through such computer research into reasoning, that what we thought were simple, rational processes are really search procedures of the most intricate sort, combined with complex ways of storing and coding the intermediate thoughts.

In other words, by using the computer to enact the processes we call language, and thinking, we have come to see what these processes entail in a deeper sense than we knew before. Thus the computer, which we took for a helper, has become a tutor in understanding complexity itself.

In the study of economics, too, the computer has provided new tools for thinking: especially ways of looking at rates and changes in complex entities like the national economy.

RATE-COMPLEXES

Things change in very complicated ways. Falling bodies fall faster and faster. Adding cold cream to warm coffee makes the coffee cooler. A dollar doesn't go as far as it used to. All these things can be expressed in terms of rates of change, but that doesn't mean we can necessarily figure out how they are going to change, when or why or even in what direction.

This is because changes can occur in so many complex ways. Some changes are simpler than they look: the acceleration of falling bodies is exactly specified by Newton's law of physics, discovered hundreds of years ago, and we can predict with remarkable accuracy how fast a falling stone will hit the ground. But we cannot predict the economy, except one step at a time. The falling body can be predicted with a single-step analytic "crunch," but many of the more complex things cannot.

The only way some of the complex rate-phenomena can be predicted is through simulation: the following-through of what happens one time period after another. From a set of starting conditions, we calculate the results one step ahead. Then we use that step as a starting condition. By doing this time after time, we come to a representation of all these changes taken together.

It was not until we had computer programs for these enactments that people in complex fields like economics began to appreciate the complications and ramifications of their own predictive models.

This is an example of why the so-called "games" we play on interactive computers can become so important.

Among the games put out by the People's Computer Company (in BASIC) there are several that will show you what unpredictable rates can do. A popular and instructive one is called "Kingdom."

In the game of Kingdom, you become a monarch with a certain population of subjects to feed, a certain number of acres of land, and a certain amount of grain.

Play goes by years. At each turn, which is a new year, you must feed your subjects and plant grain for next year. Each year your grand vizier tells you how much wheat you have and how many subjects. As the monarch, you now make the decisions for the coming year: what to plant and what to give away as food.

The results are quite astonishing. First of all, the result of what you decide is rarely what you expect. Your subjects are always hungry. If you give them too little food, some of them starve. If you do not plant enough, not enough grain will grow the following year.

"Kingdom" is very frustrating. In it you can both hoard food and hoard land. You discover in the course of play that you cannot feed all your subjects and survive as a nation. (A friend of mine says he has studied "Kingdom" completely, and claims that the only way you can win in "Kingdom" is to starve your subjects down to a handful and speculate in land instead.)

To play such a game is not merely a pastime. It is a distinctive and vivid learning experience. It gives you a strong sense of the wildness of the world.

If this were *merely* a game, it would be unworthy of note. It is a form of learning. It is not about wheat, or population, or real estate, so much as it is about *change*.

Now the kinds of change this game demonstrates could also be described by differential equations, and taught in a mathematics course. But not to ten-year-olds. And the game of "Kingdom" is just great for ten-year-olds.

An afternoon spent with "Kingdom" is powerful and memorable. It is like several months of business experience.

This is what "computer games" can do: by abstracting ideas and transforming them into vivid experience, they expand our minds. That's a lot.

The higher computer games will become, in the next few years, an important new contribution to our way of seeing. The design of games to get ideas across in an experiential fashion will become an honored skill.

Computer games that add graphics, and animation, will do even more for developing our capacities of seeing and understanding.

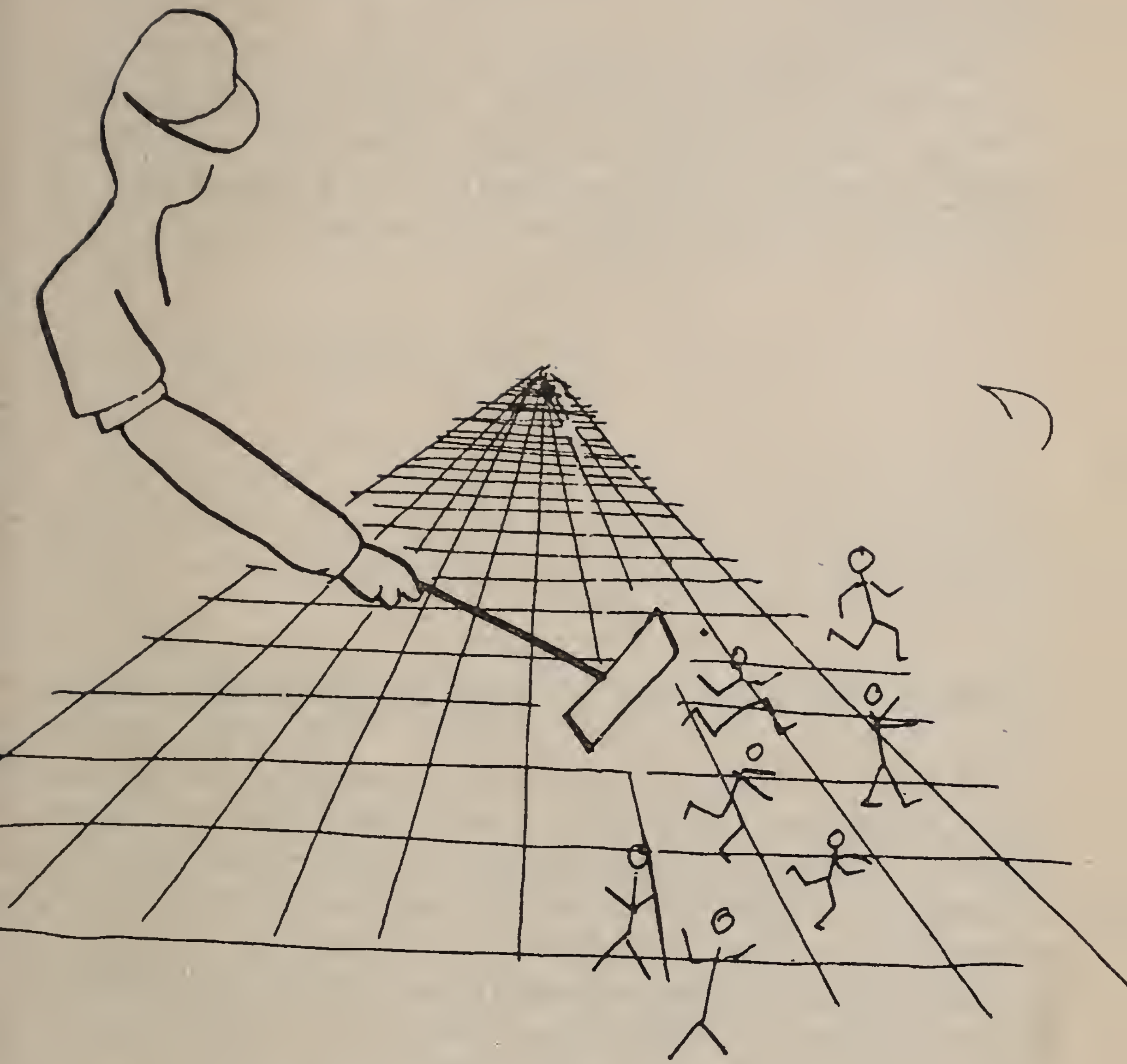
GAMING YOUR OWN WORLDS

In the game of Kingdom, the rules are set up for you. But if you write a program yourself, you make up your own rules. And just as the rules of Kingdom brought their own surprises when actually carried out, the rules you make up may have surprises too.

Creating your own imaginary world, and having the computer act it out for you, is called *simulation*. It's one of the most exciting and powerful tools, and pastimes, of our modern world. Fancy simulations have been terribly expensive, but in the near future they will be simple and easy on your home computer.

Simulation programming, especially with languages like SMALLTALK and PASCAL and APL — now becoming available for your home computer — offers you the chance to ask, "What would a world be like that had such-and-such?" and try it out.

Imagine a computer program imitating the population of the United States. Suppose there had been better birth control in 1910 — where would we be? The screen can show us.



Imagine being able to make a complex model of our whole society and its institutions — and populate it with little programs, imitating the behavior of people! Watch them scamper, as from a high building. Now satisfied with that, try out new kinds of society — and new kinds of human being.

To try out your own imaginary worlds does not give you concrete knowledge; it gives you abstract understanding. And this *kind* of understanding — the ability to juggle mentally many ideas and alternatives — is one of the best things that computer experience can give to the mind..

THE CLUB OF ROME

A group called the Club of Rome has tried gaming alternative futures. We must ponder with the greatest seriousness the studies which have emerged from their work. Beginning from some very reasonable hypotheses about the effects of these changes on each other, the Club of Rome has enacted a number of possible scenarios for the future of mankind. The results are bleak, even terrifying. Because, based on their assumptions, and starting with all kinds of different possibilities, the Club of Rome researchers came to the unvarying conclusion that the world will see catastrophic famine within twenty years, wiping out at least 90 percent of mankind.

These economic predictions come from certain views about the interrelations of population, growth, and pollution. Critics have been eager to pounce on aspects of the assumptions, and we can only hope that either the assumptions are wrong or that solutions can be found. The Club of Rome study has caught the attention of thinking people everywhere, and consequently many attempts are being made to build better crystal-ball simulation programs for the uncertain and increasingly ominous future of the world.

A NEW WORLD OF IDEAS AND INFORMATION

All knowledge, all history and science and literature, philosophy and sociology and music, are a great inseparable whole, knowable and to be known, and it should be one of the principal purposes of human life to know as much as possible about everything. Since of course nobody could know everything, the next best thing is a way of life that brings you more and more insights and understanding, information and ideas, from all over the great tapestry of human knowledge and experience.

THE NEW PUBLISHING

More than this.

The personal computer, and the communication links that will come into being for it, mean that a great profusion of new information can come into your home when you need it.

Reading from screens will not be like buying a book, which costs a large amount of money and then usually sits around unread.

Now you can have materials that come as you request them and the instant you need them.

The possibilities for the human mind are truly marvelous; unfortunately they can only be covered briefly in this book. (See, in the "future" chapter, section on the World Electronic Library; and appendix, "Personal Digital Services.")

THE FUTURE

Every computer book has its predictions about the future. Since few computer books have foreseen the personal use of computers, most such predictions have been fairly obtuse and boring.

On the other hand, some writers feel no compunction over promising you computers or boxes of some sort that will understand your problems and become your deepest friends and advisors.

The predictions here are somewhere in between.

(In the short term, the most important question is whether manufacturers will be smart and provide standardized interconnections, or be piggish and themselves lose by trying to keep their machines unpluggable to those of rivals.)

TWO YEARS FROM NOW

Ten million computers in American homes.

COMPUTERS IN YOUR POCKETS

By 1980, functioning computers will come in shoulder-strap, belt and pocket models.

The purpose of these units will be severalfold. They will be important for the control of other portable machines: dictation machines, cameras. But they will also have many independent uses by themselves.

For instance, pocket typewriters will appear soon.

It should be realized that the typewriter as we know it is one of hundreds of designs for the purpose which appeared in the late 19th century. It won out for a variety of reasons, but many other arrangements for typewriting — that is, fast transcription of written characters by finger — are possible.

Outstandingly, Douglas Engelbart's chord-writer, developed at Stanford Research Institute, is a small typewriter that looks like six piano keys. Whatever letter you want to type is made by a combination of several keys pressed at once by different fingers. Each combination of strokes, or "chord," makes one character in some code.

This will be easily miniaturized to be carried in your pocket, allowing typewriting wherever you go, with even more ease than pulling out pencil and paper. Reporters would not be so conspicuous when they wrote things down. Professors would be able to make their notes as they wandered absentmindedly around the campus. Editors could type their editorials on the commuter train, and simply plug in their machines to the typesetter when they got to the office in the morning.

A more elaborate setup would have a tray-like keyboard hanging in front of the user, like the nightclub cigarette girls seen in movies of the 1930s. This could show you what you type on a little screen, yet fold up for convenient carrying on the same shoulder-strap.

Another very important usage will be as portable record-keeping devices, glorified typewriters which actually record data on whatever subject is important to the user, assimilating it for popping instantly onto the desk computer back at the office — or home.

Already such portable digital record-keeping devices exist. For instance, they are used for inventory control in such places as supermarkets and warehouses. (One company already has a service: invading a supermarket with a team of inventory clerks, each carrying a battery-powered recording device. This recording device naturally contains a computer chip. Into the recording device they enter the number of cans of coffee on the shelves or of marjoram in the spice racks. In a few hours they grab an instantaneous picture of all stock. Nothing is brought from storage onto the shelves during this Blitzkrieg operation.)

THE GAME GETS SERIOUS

The design of working environments for people's daily use will come to be seen as what it is: not a *technical* problem, involving deep programming tricks, but a *fantic* problem — creating *video games for the mind.*

Mere programmers will be demoted from this job. A new talent will be recognized and sought after: that of designing conceptual environments, their visualizations, their feel.

There will not be many such talented people, and their work will be distinctive.

CREATIVE FACILITIES

Already interactive screen systems of sorts are available for writers, musicians, motion-picture editors, makers of charts and slides for business use. In the next few years we will see as well screen-systems appearing for a wide variety of business operations and secretarial work; and for graphic arts and drawing, motion picture cartooning, sculpture, and much more.

These will offer more than just ways of expressing and executing creative choices and seeing the results immediately. As such systems advance, they will allow the users to try out and visualize complex alternatives. It will be possible to try out your alternatives in some detail before deciding on final versions of whatever you are doing.

ELECTRONIC AND AUDIO CENTRAL

Your personal computer can become the controller of your other audio (and TV) equipment: a glorified clock-radio and answering service, embracing all forms of personal electronics.

HI-FI AND VIDEO

Right now, the most bothersome hookup for the home computer is that from the computer to the television screen. But in fact the television set can more easily have the computer built into its initial design than attached later on, and more cheaply. By 1980 we may expect to see television sets with built-in computer, and vice versa (depending on whether the manufacturer wants to call it a television first or a computer first).

We may expect also to see programmable high-fidelity sets. These will be hi-fi components which have cable connectors that can go into the computer.

We will also see high-fidelity sets with built-in computers, offered with "entertainment switching" software. Basically these will be computer programs that allow you to pre-schedule what you want to hear, see and record. Such systems should also allow you to condense the things you want to do into highly simplified controls, so you can switch between programs, fade in and out, and in general manipulate all your audio wants with simple gestures at your screen diagrams.

High-fidelity makers are quite likely to cooperate with the spirit of amateur computing, and allow easy control by your home computer. This is because the high-fidelity market rests on the standardization of signals and signal plugs. People there already know that standardization is to everyone's best interest in developing a viable industry.

Television manufacturers might well think they might want to go it alone, providing computer control to the TV without including the hi-fi system. This would of course be very short-sighted, since most people's leisure time seems to embrace both audio and television. Thus it would be self-defeating for either group to plan on a system not integrated with the other.

As in other areas, short-sighted manufacturers might also try to build into the TV or hi-fi a computer that will not connect to your other, regular computer. This would be dumb, for obvious reasons.

RECORDERS

Recording appratus, almost always badly designed anyway, can yet now be greatly simplified at the user level.

Instead of having levers and buttons whose function is to stop and record, and having to do all the rest by combining these with a series of complex manual operations, the user can do everything much more simply when the controls go to screens or other electronic simplification. This also means a new ease for the more complex functions: mixes, fades, sound-on-sound, sound-with-sound, and all the rich tricks of today's recording studio.

Such manipulations of tape recordings are presently so complicated that only a few people — usually professionals — bother to do them. It is none of it that complicated *in principle*, it is just that the more you add to the problem, the more details there are to be worked out, and kept track of mentally. Computer-screen controls will make it all simple.

In addition, timing tracks can be easily added to all recordings that will make mixes and other functions far simpler — and easier to do on less equipment.

DICTATION MACHINES

One special kind of audio recorder, the dictation machine, still languishes in a primitive state. True, they have gotten very small; but not yet *better*.

Tomorrow's dictation machines will have clock chips and leave timing tracks, so that it will be possible to ascertain exactly when a given dictation occurred. This will be useful for all kinds of functions in which timing is critical, as in bus dispatching, security recordings.

Timing tracks will also make it possible for the typed record of speech, as in stenography or court reporting, to link directly to the actual recorded instant. This will be very important for verification — as well as "oral history" libraries.

It should be noted that machines do not exist, and cannot be presently built, which will recognize speech and transcribe it to the printed word. Many people say these will come about soon. The Speechlab, available for your classic computer, recognizes a few words. But the recognition of real speech, as ordinarily spoken, is still far away.

THE REPLACEMENT OF CLIPBOARDS

We will see this approach extended for a large variety of fields. The subway trainman will enter the times of arrival and departure at various stations as they occur. Policemen on the beat will use a small portable computer for their standard form reports. Interviewers will use an electric portable typewriter instead of a clipboard.

In general, the use of a clipboard indicates a function which will be taken over shortly by a computer recording device. Clipboards are ordinarily used to hold printed forms, which divide and categorize information for some sort of record-keeping. Such records are then ordinarily taken to a central place, keypunched, tabulated by computer.

To replace these with computer devices is simpler, requires less training for new personnel, and permits all information to be directly entered into a computer when the device is returned to home base.

Dispatchers, interviewers, policemen on the beat, watchmen, drivers — all will use portable computers for their regular reports.

Personal time recorders, on which you can also make notes, will become popular for industry and home. Personal time recorders, their computers tied to scheduling programs, will make it easy to keep track of what you do with your time. Executives will use them to make note of work undone. Future Prousts will be able to know where they were at any given minute or hour of their lives.

YOUR TELEPHONE CONSOLE

You'll be able to make all your phone calls without dialling a number: just point to the name, on your screen, of the person you want to talk to. (Bally has announced this already.)

This means that the same screen you do your writing and memoranda on becomes your telephone console as well.

Most usefully, you can "stack your calls." The system can even take the initiative in dialling the calls you've listed when you pause in other work. Or at intervals, as a break.

In addition, a more advanced system will feed you notes you've prepared for the call and make a log of what you've said, which you type in as you talk.

We will see interesting switching setups for the telephone — possibly the same ones as for other audio. So you can switch other audio signals in and out. The computer can also be used for controlling conference calls, extensions, and so on.

(Regrettably, this also means that nuisance calls, as for advertising, can be made more easily and cheaply than ever.)

PHOTOGRAPHY

Every good camera, after 1980, will have a built-in computer. This will record the hour and place of the picture in the picture's margin, among other things.

Instead of having to have your own darkroom, there'll be a service available that gives you fine control of photographic printing. You'll use a screen, on which you can crop, dodge and modify till you know what you want. At that time the computer, too, will know what you want, and do it for you.

It seems likely that a combination device, the camera-typewriter-recorder, will appear for journalism, then be taken over by individuals for general use.

PERSONAL SERVICES

To complement your personal computer, a wide variety of services will be available. (See Appendix, "Personal Digital Services.")

DIALLING IN FROM YOUR OWN COMPUTER

Often you must deal with the information banks of large organizations or bureaucracies.

You want to place an order, or find out why your order hasn't come, or apply for unemployment insurance, or find out why *that* hasn't come, or reserve an airline ticket, or whatever.

As things are, you have to call up or go in, and talk to some sort of clerk, and this *clerk* has to go fish the information out of the computer.

This is silly and wasteful.

There is no reason you yourself shouldn't have access to the computer, dialling in your orders from your home computer, checking out what's wrong on your screen.

(Indeed, one home-computer company has already announced that you can order from them by telephone — computer to computer.)

There are two possible objections.

1. *Using the system you're dialling into takes a lot of training.* Well, it shouldn't. Systems that just reflect the content of paper forms should *certainly* adhere to the ten-minute rule; more like a one-minute rule.

In other words, if a company's computer system is too complicated to let you into, it's badly designed.

2. *Danger of misuse.*

Well, there are several points here.

One is that the credit-card companies allow telephone orders and yet manage to live with possible fraud. In other words, precautions can be taken.

The other is that there are ways around it. For instance, if the computer is too easy to break into — as most are — attach a little computer to the front end for handling outside transactions. And program the little computer to allow only certain types of questions and input. And have *that* little watchdog computer pass on the requests.

It can be done.

PERSONAL SCHEDULING SYSTEMS

Most of our acts are not isolated, but parts of larger plans and ongoing activities in which each step relates to those before and after. (Note that the plans we are talking about here are not computer programs, but plans for human projects — like building a bridge or having a picnic.

Some parts of a plan can be carried out quite independently of one another. Others cannot be begun till still others are done with.

It is often helpful to map out these complexities, what has to come before what in a project or a chain of events. Such maps are called PERT charts. A PERT system makes clear what things are required before you can start what other things. In recent years, government and industry have used PERT to get many complex things done on schedule — such as the Polaris missile.

Most importantly, these systems allow the changing of complex plans in midstream.

For instance, in studying the map of a plan, you may see that it is possible to speed one particular step of the plan, thereby speeding up a dependent step; or to gain time by making valiant and extraordinary efforts on certain other steps which would otherwise slow down the whole project. Thus the project's overall time can be speeded up.

Such systems will soon be available for personal use. And simplify our lives a lot.

YOUR CAR

Already many cars offer checkout couplings - sockets whereby a computer can check various internal conditions of the vehicle.

Shortly, Detroit cars will have computer chips to control the whole electrical system.

Many cars already have electronic ignition; later models will have computer chips that adjust the ignition based on style of driving.

More fanciful purposes will come soon enough. Audio systems for vans under computer control, for instance, that offer a fade between sets of speakers — perhaps when the vehicle has been parked for a certain length of time with the radio on.

In a couple of years, you'll be able to get a message board for your rear car window that will tell other drivers specifically how you feel. (Watch your language.)

Deluxe vans may be offered with a full graphics panel on the side, instead of a paint job. This would permit animated cartoons and movies as you drive — and emergency announcements, or advertising, by the roadside.

The complete Computer Van will be the hottest thing on wheels, soon. It will offer the computer hobbyist a complete system, together with transportation to work, all for one monthly payment.

YOUR FURNACE

Furnace manufacturers will build complete electronic packages for computerization. The manufacturer of a furnace will sell the necessary switches and connections as an accessory package. Indeed, the manufacturer might provide a built-in program which you can't modify or replace. This seems unlikely, however, because there is no simple formula for calculating the heat necessary in human habitation — there are too many different factors having to do with the building, the climate, and the people.

On the other hand, the furnace application is one for which the price of the adapter kit — several hundred dollars — might just as well contain the computer chips as well. Why tie up your main personal computer when you could have it all taken care of by another unit? But you'd better be able to control it from your main computer.

TEMPERATURE CONTROL OF THE SHOWER

If we can put a man on the moon, and a computer in your pocket, we can spare some circuitry to make showers stay the same temperature — no matter what happens elsewhere in the plumbing.

FIVE YEARS FROM NOW

Five years from now, virtually nothing you do will be done the same way.

Mechanical typewriters will be as extinct as mechanical adding machines are today. All writing, and much reading, will be from screens; not to write at a screen will seem as anachronistic as to write with a quill pen does today.

CHANGES IN THE WORLD

At the end of 1982, there will be between twenty million and thirty million dinky computers in place, in the United States alone.

There will be hundreds of different brands, with Asian firms prominent.

IBM will be in a precarious position. Just as IBM has had its customers "locked in," IBM is locked in too — the jailer himself manacled. *No part of IBM's product line is immune from the effect of dinky computers.* However, to introduce them to its own product line at a price comparable to the others could undercut the whole carefully-built edifice of products. And a noticeable slackening of profits, from the impact of the little machines, would drastically lower IBM's stock prices, its prestige and standing — and quite destroy the myth. IBM could fall far and hard.

If Xerox moves swiftly to exploit its SMALLTALK language, they could sweep the entire field of computing with whatever machine they chose. Since they probably will not do so, the Kay group might break off and take it elsewhere: a joker in the deck.

The main computer chips will probably be the Exxon Z-80 and the Texas Instruments 9900. This because both products are good and both firms sell hard; the superior LSI-11 will probably be left behind by flaccid marketing.

"STRAIGHT" COMPUTING FIVE YEARS FROM NOW

The ever-expanding use of big computers will deflate. It will not end, but it may fall, as so many of its uses turn out to be easier on dinky machines.

COMPUTER LANGUAGES FIVE YEARS FROM NOW

A variety of high-power computer languages, offering fine control of computer animations and sound generation, will be available. Such languages will include SMALLTALK, PASCAL, APL, LISP, SIMULA, and many more.

DIFFERENT STYLES OF SOFTWARE

We will see a growth of more and more different programs for the very same applications, each embodying particular styles of operation which match the personality styles of different client groups of users.

"ELECTRONIC MAIL"

The post office is a shambles. Your personal computer can send writings and receive them, probably for the same price as text sent in envelopes.

No commercial system is presently available. While these will apppear within five years, the hobbyists could really set up the whole thing themselves.

Networking appears likely to develop among hobbyists on a round robin or grid basis, if a truly cooperative spirit can be maintained.

A precedent for this would be the fellowship among radio amateurs by which messages are relayed; while this tradition has persisted in CB, it is not strong there. (Another tradition is Russia's *samizdat* publishing, the secret copying of writings forbidden by the state, where each member of the chain makes a certain number of copies.)

But the commercial electronic mail systems should be going strong by '82.

AUTOMATIC TEACHING

Much has been made of possible systems for teaching by computer (called Computer-Assisted Instruction, or CAI).

Basically the idea is to break up any subject into small chunks, present them through the computer screen, and then drill and test the pupil and clear up any misunderstandings. This is expensive, difficult and restrictive.

In experiments with computer-assisted instruction, researchers have created branching programs which appear to mimic human teachers. This is extremely difficult and time-consuming. Besides requiring exhaustive analysis of the structure of the subject, it tends to force the formalization of such subjects into stereotyped sequences of presentation. It is the opinion of the author that such situations represent a dead end, or at least a stagnation. (See *Computer Lib*, 113-110.)

We will see a great deal of this stuff in use five years from now. However, it will be more in the area of *specialized training* than what we think of as broader education. It will help people learn skills or particular areas, but probably not to get intellectual overviews.

MAGIC ENVIRONMENTS

Fanciful light-show environments, as for discotheques, exhibits and other purposes, will all be under computer control.

Complete "computerized" houses — that is, with all the light-circuits and appliances under central control — will become common.

However, there is a safety problem, and such systems can have very inconvenient bugs. (TIME recently cited an apartment of this type, probably the first, that "went down" — making people have to eat dinner by candlelight when they hadn't meant to.)

COMPUTER FANTASYLANDS

There will be worlds people can play with and control on the computer screen.

These will be wonderlands with characters that the user can control or interact with. The user can fly over landscapes, wander down roads and be surprised by who and what he finds there (as in the Oz books). Characters in the system can be his pretend-friends.

This may or may not be good for children. Adults will use it too. It may fit on your home computer or it may require bigger setups, available only in arcades.

COMPUTERIZED SEX MACHINES

The already-booming market in "marital aids," and other raunchy paraphernalia, will offer new and better forms of stimulation including the computer; these will have the unusual new capacity of responding in an integrated and lively fashion to user input. Rather than being repetitive — and thus "mechanical" in the usual sense — they may provide more fun for their users than some live partners.

COMPUTER SCREENS FROM THE PHONE COMPANY?

Even the telephone company may be offering facilities for personal computer use.

Charles Jiudice, at Bell Laboratories — the development arm of the Bell System — has been working out a computer screen display that could become a home terminal. Working within the unique organizational politics of the telephone company, he has had to present this internally in terms of the psychology of telephone executives, for some use they consider "practical" — but it may result in a screen display that the Bell System can rent you by the month.

ROBOTS

The image of the "robot" has grown on us for years: the humanoid that talks and thinks like a person.

Well, it can't be built; nor is it conceivable in the foreseeable future (i.e., before 1987). Nothing genuinely like a human mind can be put on a man-made computer of any size — let alone mounted in a body that walks.

The so-called "industrial robots" are huge mechanisms for assembly lines: huge arms that rotate automobile bodies and dip them in paint, TV cameras that check for defects, and so on. The only reason they are called "robots" is that the word was lying around with nothing to refer to, and the builders of industrial-arm equipment took it over for their own use.

However, "robot pets" — wandering computers with wheels — will probably turn up soon. They won't do much, but should be fun for entertaining cats and toddlers, or making recorded announcements.

NICE NEW WORLD

In *Brave New World*, Aldous Huxley foretold a world populated by various levels of fool and ignoramus, and presided over by an elite, the Alphas, who were the only ones to know and understand anything.

To some people this has become a vision of what the world should be like. Indeed, to some people such a status system of technocrats and drones seems too be implicit in the world of computers; so some people have to be degraded, and suffer, while others lord it over them.

But it is possible to hope that small computers can, if we want them to, make life more pleasant, make the inequalities of mankind less painful, and help us all pursue our individual wants freely in our own personal styles.

This is what I mean by the Nice New World.

But the first computer age has created and abetted many traditions of oppression. To eliminate them we must understand them.

THE ELIMINATION OF FORMS

Many people associate the computer with obnoxious forms which have to be filled out. This is because, in customary computer use, programs have required information to be set up in very standardized and boxed-in ways, and a customer or client is required to fill out a form which corresponds to these computer codes.

Such forms constitute a general harassment of human life, and it is understandable that people think that computers need them. However, exactly the opposite can come about.

What such forms do is arrange the information for the convenience of the computer program, and of the typists and clerks who are saddled with the monotous job of transferring this information from the paper into some computer form. Ordinarily they are typing onto punch cards, and by tradition these punch cards have been laid out in very rigid ways. Hence it has come about that such forms have arisen: as paper maps of how the punch card is divided into sections.

And there tends to be no comprehensible explanation of just what is wanted in particular boxes.

This has been because the boxes are very small, and because nobody knows how to write directions, and because within a bureaucracy they want to avoid clarity to avoid criticism or understanding from above or outside.

And so it has come about that the forms must be supplemented by clerks, who explain what is really meant by the forms, and who have ample opportunity to insult and degrade the people they meet for somehow not understanding the incomprehensible, garbled and irrelevant.

All this, all this can go.

In today's computer world people fill out forms (dimly understood) which are then keypunched, funneled into the computer. There the information collides, is found not to match, and peremptory replies are issued, which are mailed back to the customers.

But instead of having to fill out forms, the user should be given a computer screen. There he can type or select his answers, and when the question is not understood, get clarification. And the machine can be very polite.

BUREAUCRACY: THE LARGER QUESTIONS

But the problem of "forms" is in some ways shallow. Forms are just the checkerboards of bureaucracy. (By "bureaucracy" I mean organizations in which people supposedly serve the public by rules, but actually wield extensive discretionary power.)

Bureaucracy is strange. Many factors make formerly decent people, in a bureaucratic setting, tightfisted, secretive, abrasive and cruel. Bureaucracies may say they are for the good of the public, but in their actual workings, they become something very different.

"Red tape" is composed of those demands made on the clients or victims of the bureaucracy: the nuisances, petty requirements, permissions and bottlenecks that make life difficult when we run into the organization of government, hospital, university and so on. It is the forms to fill out, the signatures to get, the variances to be approved, the permissions and filings.

Sometimes these are honestly set up, for well-meaning reasons.

And sometimes they are intentional systems of obstruction, designed to create the appearance of opportunity while in fact thwarting it in most cases.

Generally they are both.

In many situations, such as welfare and medicine, resources are allocated for the good of the public, but their dispersal is carefully controlled according to informal rules different from the official ones. In welfare and unemployment administrations, employees are directed to be as stingy as possible in giving out the funds, regardless of official eligibility. Similarly, in medical clinics, nurses can withhold access to the doctor, or qualification for tests and other benefits. Thus all bureaucratic confrontations become a game of the client versus the gatekeeper. In other words, bureaucracies are often set up to *prevent* service.

Another main principle in bureaucracies is that of individual power, and whoever is able to hoard to himself the decision to benefit or not benefit someone thus has more power — in some cases, power to improve his ratings on the job, in some case competitive power within the bureaucracy, and in some cases power for the mere thirst of it; in some cases power to do what he or she actually thinks is right. (And, in some cases, an opportunity to fish for bribes.)

THE COMPUTER EXCUSE

Till now, the computer has been a pretext for the oppressions and evasions of bureaucracy. That pretext can be taken away, and computers can go a long way toward making things better.

We could have little computers quiz applicants and clients on their names, backgrounds, and whatever is needed for the medical history, the welfare eligibility check, or the like. Such a form of input would be far more pleasant and complimentary than the forms of interview which now occur.

The computer can be patient, since it is really indifferent as to how quickly the person answers. It can be courteous, explaining to the user just what sort of thing he should put down, even though it has done so a hundred times before that very day. And it can be clear. The branching capability of computers means, for example, that each user can get instructions oriented to his own particular needs. (This assumes, of course, that the system's designer has the skill and foresight to put the instructions there.)

It is perhaps ironic that a computer can be more human than a human; but if you've had experience with the unpleasantness of human interviewers in bureaucratic, academic or social-welfare systems, it should be obvious. While many such people are extremely courteous, many are not. Such jobs make people skillful at the small insult, the twist of scorn, the neat humiliation.

The little computer, however, cannot take away the structural evils that computers previously had to front for.

If people are going to be insulted and degraded in the future, it *will not be because of the computer!* That excuse will be forever torn away. It is going to be because the programmers didn't feel like bothering, or because those who administered the system do not want to be hassled, or it is going to be because nobody cared enough to do better, or because the bureaucrats enjoy their petty power and the degrading of others. But by 1980, anybody who says incomprehensible forms, red tape or obnoxious treatment are the computer's fault will be either a damned liar, or an ignoramus, or the prisoner of a rotten computer system.

(For more on this point of view, see *Computer Lib.*)

THE FAR FUTURE (beyond five years)

Anyone who tries to predict beyond five years is crazy.

ARTIFICIAL INTELLIGENCE

There are certain people in the computer field who believe they can make computers behave like people. Eventually, through programming or special hardware, these people believe they will approximate, or equal, or surpass, the thinking processes of the human brain.

Not just calculations, or behavior in specific areas, which we may grant them easily. But *all* behavior.

In other words, they believe that at some near date, by methods they have not decided on, they will make a creature superior to *you*.

Spokesmen like Marvin Minsky, perhaps speaking tongue in cheek, claim that machines the equal of people are not far off.

This topic deserves much more space. (It is covered at some length in *Computer Lib*.)

Others believe that their research yields techniques that will make computers operate for us better, and anticipate our needs, even when *we do not understand how the programs work.*

We will consider these views in turn.

Joseph Weizenbaum is a thoughtful researcher at MIT. A few years ago he created a program called ELIZA that appeared to talk back to you intelligently when you conversed with it through a typewriter terminal.

Astonishing feats of seeming intelligence were demonstrated by ELIZA — actually a fairly simple computer program. People went nuts at the terminal; thinking the program was some kind of intuitive psychoanalyst that could see into their very souls.

Now Weizenbaum has written a book that greatly annoys his colleagues in artificial intelligence. Called *Computer Power and Human Reasoning*, the book questions the sense of trying to make machines act human when they really aren't.

Tricks of response and exploratory techniques can be mechanized, and they can even imitate human behavior in certain circumstances. To suppose they can imitate and surpass the human brain, though, is quite another matter.


HEY DINGBAT
HOW'S YOUR OLD
RAZZ
TWEE

RESPONDING OBJECTS IN OUR LIVES

Yet many objects, from vending machines to vacuum cleaners, will be responding in fairly complex ways to what we do.

Instead of talking about the seeming "intelligence" of these machines, which is drivel, we should distinguish between those machines whose behavior is easy to understand, and those machines whose isn't.

Let us call these *clear* responders and *opaque* responders, respectively.

The author does not understand why anybody would want opaque responders around. Human beings are difficult enough.

THE NUISANCE OF THE PSEUDO-PERSONAL

Some of those computer people who think machines will become human, or superhuman, also seem to think any step toward this is desirable — such as having machines that call you by your first name, address you in slang, and call you "turkey."

This is abominable: the everyday insults of the service counter, the employment agency, being brought into the world of computing! Why can't a machine be *more* polite than people?

But the nuisance extends further. By programming machines to *pretend* to converse and understand, like Weizenbaum's ELIZA, they add more bother to our lives. Quickly we learn that the machines do not *really* understand us; but we must waste time and effort figuring out how they are really responding. It is like trying to handle a drunk: a person who one instant is trying to beguile us by acting socially rational, the next moment lurching off on a demented errand that must be stopped.

A WORLD ELECTRONIC LIBRARY?

One of the more dramatic possibilities for the far future is that of the electronic library. While many people have imagined and advocated electronic libraries, the question of just what such a library would do, and how to set it up, have boggled most thinkers. How to create it, who will pay for it, and how it will continue to be supported, are important questions.

If these difficulties could be overcome, the possibilities are electrifying. Imagine being able to dial up the library from your personal computer, and at once to have access to anything you wanted to read!

Now there are many people who do not read much, and many people who do not read at all. (Jimmy Walker, dapper mayor of New York in the early part of the century, once asked someone, "Now, if I wanted to buy a book, where would I go?" He may or may not have been kidding, but the question illustrates an attitude that a lot of people have.)

But there are others — five or ten percent of the population — to whom reading is like the air they breathe, an essential nourishment, an addiction. For these people, an electronic library could have a special meaning.

A world electronic library could mean that you could find an article or explanation of any given subject that would speak to your personal understanding.

A world electronic library could mean that if you wanted to learn a new subject, and didn't know where to begin, you would simply go to any terminal, and browse *toward* the subject you wanted to know about. On the system you could find alternative approaches to the subject until you found one that appealed to your own personal style of learning and questioning.

A world electronic library could mean that education at all levels would be cheaper, more rapid, less offensive and embittering, and more useful to people with a specific and definite need.

A world electronic library could mean that you could write and file for posterity anything you chose, and publish it instantly: the moment you pressed the "publish" button, or its equivalent, your piece of writing would become available to all users of the system, and automatically listed and cross-indexed in the great directory. All frequent readers interested in your subject would be notified of its availability.

A world electronic library could mean that readers could write separate comments on what they read which would in turn be available to other readers of that same article.

A world electronic library could enable researchers needing to pursue obscure writings to get them immediately.

A world electronic library could mean that you could write letters and transmit them instantly to friends and relatives, without the delays of the postal system; and instead of your having to save the letters in a paper version, the library would save them for you, at your expense, in a form no one else could read.

Three centuries ago, the scientists of the world were a small fraternity, well acquainted with one another, who shared a sly, secret knowledge of how the physical universe really worked. Today any highschool student or any thoughtful person who can browse through the *Scientific American* for a year can learn the secrets of those earlier men, and more. Tomorrow, the widest and deepest secrets of today's science, yesterday's history, can be as accessible. Because a world electronic library could bring forth new forms of reading and writing to broaden and clarify human understanding far more than was ever before possible.

It is quite reasonable to fear the world electronic library. Many of us remember George Orwell's book, *1984*, which depicts a terrible dictatorship of lies and manipulation. In this world, all writings are completely controlled by the state, and the hero is a butcher of scholarship who continually rewrites the historical record in the interests of the faraway dictatorship. The articles he removes from the official encyclopedia are consigned to the trash, and thus forgotten forever.

Observers of modern intelligence techniques, and the technology of computers, naturally fear that a truly comprehensive electronic library would threaten freedom in precisely this way: if all writing were stored in a centralized place, the evil and unthinking minions of some future dictatorship, or bad guys like those of Nixon's White House, could obliterate and change history by mere electronic rewriting of the stored documents.

Some electronic libraries that some people have thought of would have precisely this danger; but it may be possible to prevent this in various ways, at the same time providing the public with better, surer, and more opulent sources of written information than have ever been known before.

(Some conjectures on how to organize such a library will be found in *Computer Lib*, pp. 73-71.)

GREAT ISSUES INVOLVING COMPUTERS

All is not sweetness and light. The good side, the attractive part, has been presented. Now we have to talk about what stands in the way — or worse yet, may be aimed at your very heart.

COMMUNICATION PROBLEMS

Many people in the personal computing field are earnestly working out the details of communication-systems for personal use. These are going to have rough going if the equipment is not in place fairly soon.

Much stands in the way of easy communications between home computers.

One of these problems is the telephone system. Home computers will greatly increase traffic on the telphone network within five years. The phone company is not prepared for this, and might come up with some very uncreative solution, such as attempting to ban personal computers on the telephone — which would create a new and expanded coalition of telephone bandits.

Similarly, the Federal Communications Commission, a government group that makes all the rules for electronic communicaions, has no inkling *whatever* that this is coming. One of the purposes of this volume is to warn them. But it is often said that the FCC only responds to pressure. So it is likely that a personal-computing lobby must soon place representatives in Washington.

One outfit that talks to the FCC is the White House Office of Telecommunications Policy. Presumably their job is to provide initiatives and foresight that the FCC doesn't have.

Unfortunately, both of these outfits may have their minds mostly on satellites and licenses for stations and cables. These are nothing to what is about to happen. But how to make room for the coming torrent may not have reached them as a problem.

One important coming development will be laser communication (which is not at present supervised by the FCC).

Lasers now come in all sizes and types. Basically a laser is somehing which sends a small, unspreading beam of light whose pulsations are extremely accurate. A little laser setup could easily connect two houses five miles apart. This provides a continuous telephone link with no money paid to the phone company. One of the byproducts of personal computing could be a private network of laser-links around the whole country — for both data and chitchat, mixed together.

THE COPYRIGHT AND ACCESS PROBLEMS

Much writing, and other material, is going to be stored on computers you can reach.

In American tradition, copyright protection makes possible a profit for the publishers of writings. This has been curiously extended so that, for example, the telphone company has a copyright on the telephone listings themselves.

Such copyright may mean that only certain suppliers will offer certain kinds of information, because they have the copyright. This would be okay if the suppliers could give us the kinds of access we need — such as the ability to make marginal notes, for example.

The problem of how to create data banks that individual users can reach, and add to according to their own needs, is one on which much more will be said.

THE '1984' QUESTIONS

The actual year, 1984, is just over six years away. The scary things in the *book*, "1984," may be here already. Or not. It's hard to tell.

In the book *1984*, there was a perpetual pointless war; people were put in jail for no reason; the government knew all about individuals, and violated their privacy; the government changed the history books; the government did dastardly things to its citizens on a systematic basis.

As far as pointless wars go, we had one in Vietnam until quite recently. People have been imprisoned unlawfully. But let us consider the other items.

THE PRIVACY QUESTION

Many records are kept about you, both by businesses and government. Now that these records are kept on computers, they can be searched rapidly — either for information about you in particular, or, say, in search of potential victims for some criminal scheme.

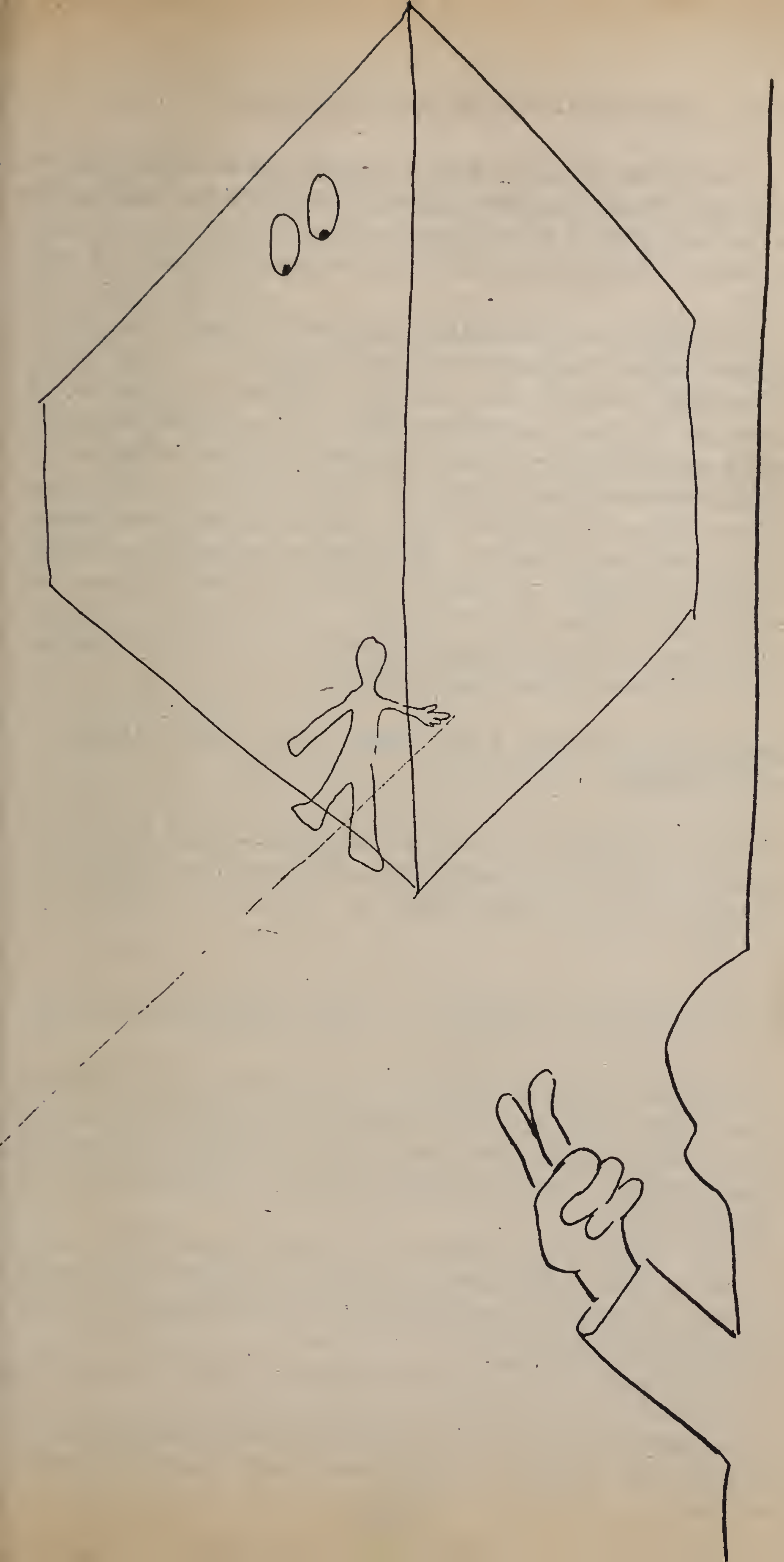
New laws prohibit both industry and government from keeping such records, past a certain minimum needed to conduct their work. So that supposedly a bad guy cannot trace your records and bring them together from a variety of sources. Whether such laws can work is another question.

SECRET CODES FOR PRIVACY

One very threatening problem has been this: can't counterfeit messages be sent easily? When a typewritten message arrives over a computer net, how do you know it's really from the person whose signature is typed at the bottom? And how do you keep it from being read elsewhere in the net?

This becomes, of course, the question of how your stored records can be kept private.

Amazingly, these problems appear to have been solved. Researchers at MIT have discvered a new coding-system which apparently can't be broken, and which actually permits validated signatures! (These new "trapdoor" codes are described in the *Scientific American*, August 1977, Martin Gardner's column, 120-124.)



THE HISTORY-BOOK QUESTION

In American history, the government has often misled the public. The Spanish-American War and the Philippine occupation, the seizure of Panama for a canal, are among the less controversial cases of the manipulation of the public.

Till now, there has been reason to believe that time would clarify the truth; and so it has.

But the prospect of electronic libraries, spoken of glowingly earlier, has this worm in the apple: what is stored electronically can be changed electronically.

Yet there may be considerable hope. If we disperse the storage of our electronic libraries; if we encode them in untamperable ways (below); if citizens can exert an exact watchdog funtion; we may be able to prevent these evils regardless of who gets tempted.

There is no room to deal with this in the present volume; I hope to cover it in a later book.

SECRET CODES AND THE HISTORY-BOOK PROBLEM

The new secret codes of the "trapdoor" kind, mentioned above, make it much more likely that the history books — or their electronic equivalent — cannot be maliciously changed.

"COMPUTER CRIME" — AND INDIVIDUALS

Donn B. Parker, of Stanford Research Institute, makes a good living lecturing on the subject of computer crime. He has collected hundreds of examples of embezzlement, and other swindles, carried out by computer programmers and manipulators.

So far these have been crimes against companies. But this may change: crimes by computer against *individuals* may be in the offing, especially through mailing-lists and credit-bureau records.

Nobody is going to get you through your home computer.

We hope.

The important thing is to understand the dangers by understanding computers in general. Your home computer can help.

CITIZEN AWARENESS

If agencies of our government are out of control, that is hardly something we can deal with here. But increasingly widespread allegations link organized crime to the CIA, and even suggest systematic invasions of individual privacy by various agencies.

Example. It was recently alleged in the news that personal records, illegally kept by the army, were illegally saved over the ARPANET, a government-sponsored hookup between research computers.

Example. The Bureau of Standards has proposed a coding-system to ensure privacy of communication between computers. Yet critics allege that this code has been made to be broken, and that the National Security Agency stands ready to tap into every electronic message that gets innocently coded by this scheme.

How can we know? Well, if government agencies are defying the courts and the constitution, things have come to a pretty pass.

But these are among the issues that citizens — and Congress — are going to need to know about. And the personal-computer movement will soon thrust a lot of people into these controversies.

APPENDICES

THE END OF THE MYTH

The myth is everywhere.

We see it in the computer systems that laymen collide with as victims: as students, as welfare clients, as hospital patients. We see it in the computer systems that people collide with as consumers: charge accounts, billing systems, credit-denial systems, account-shutdown procedures. We see it in the computer systems that employees have to learn to use without understanding.

Computer programmers have been fooled by it, and have in turn fooled others. Computer salesmen have been fooled by it, and have in turn fooled others. Computer manufacturers have been fooled by it, and have in turn fooled others. Laymen have been fooled by it, and have no way to turn. So they hate computers.

The myth, most simply expressed, is:

Computers are oppressive, and this is good.

You have to have a number. Things have to go in on cards. "We can't find that out, it's on the computer." The computer "made a mistake," too bad for you. It can't by changed, it's on the computer. Sorry, your name can't be over twelve letters. You couldn't understand that, it's computer code. It has to be set up exactly this way for the machine. Those are the only categories. You can't understand it, it's the computer's fault, it has to be that way, there is no recourse.

A slightly different version of this piece appeared in the maiden issue of *ROM,* July 1977.

Progress means regimentation. Complication is good. To use computers means everything has to be changed into numbers. When you computerize a company, its systems of work have to be completely thrown out. A man in an expensive suit who uses baffling phrases must know a lot. There is only one computer manufacturer. Nobody could sell a lot of computers unless they were the best possible. If a computer product is complex and imcomprehensible, it represents the latest and best. If a thing is "very technical," it must be right. Computer people always know what they're doing. And progress has to hurt.

It's all one big damned lie.

Intimindation, mystification, regimentation: the everyday kit with which all too many computer people, and some computer companies, have gotten their way; the litany, the "God's Will," of the computer priesthood.

But every computer system was set up by somebody. All those restrictions and nuisances, however excused, come from the conscious choices made by people, somewhere along the line.

(I remember vividly a meeting with the guy who was nominally in charge of security for a big-computer installation. A colleague and I wanted to attach a graphic system to the big computer, because we didn't understand the game. I will never forget the malevolent grin with which this fellow refused our every request. He had an answer to everything, neatly couched in systems terminology, but the grin and the meaning were entirely clear: you *will* not attach a graphic system to my computer. The game was to prevent the growth of interactive, potentially independent, systems. And prevent it he and the rest of them did.)

And now all this is about to change.

Little computers, with easy-to-use programs, are cropping up all over. And many of the people working on these systems have a real commitment to the opposite idea: that computers should be easy to understand and easy to use.

The impact of these developments on the business world is going to be formidable, profound and revolutionary. Easy systems for accounting, typing and filing, scheduling and personal databasing, will appear quickly. First to grab will be the innovative small businesses; but then larger firms will come too.

And a paradoxical situation will emerge. Those companies with big computers and big computer staffs, complicated procedures and everything nailed down, will look on in horror as their competitors — the competitors who earlier lagged in catching up with "progress" — run circles around them. For the age of "one person, one computer" will be here; the age of one-by-one transactions by people who know what they're doing.

The structure of the computer world evolved not from real considerations but from marketing tricks. Big computers and batch processing, once necessary, became a way of locking out minicomputer manufacturers and interactive systems. But it will become plain for all to see: most usage of computers is best done on interactive minis, and should have been all along. Large data bases need large *disks*, not large computers. Interactive input and query systems can eliminate forms, eliminate paperwork, cut down on the red tape we all hate. Small computers for typing can not only make typing accurate and virtually instantaneous; they can automatically file in multiple directions simultaneously, so that (for example) a letter automatically connects with all the files it bears on. Interactive systems mean people can understand what they are doing, rather than be brutalized into dead-end clerical peonage.

Presently, armies of programmers are employed doing complicated things that someone thinks are wanted. But when the smoke clears a great deal of this is going to turn out to have been make-work: ad hoc, ungeneralized and unstructured programming elaborately interfaced to horrid operating systems. The general problem is not *more* programs but *better* programs. Particular and temporary programs will be replaced by the general and simple. I believe that in the future we will find that a few simple programs will do most of what's wanted in business. And a new generation of businessmen will see that computers can be easy and accessible.

Reliability will increase: instead of "the system being down," individual *units* will go down; a guy will just borrow a computer from the next desk.

And we are on the brink of the home computer age. By 1980 there will be computers, I believe, in some ten million American homes.

A host of services for the hobby and home user will appear, evolving from the computer store. Right now, a computer store is a place where you go to buy a computer kit; you're lucky if there's an assembled unit in stock. Tomorrow, the computer store will be an expanded service emporium as well, where you pick up your printout (like film from the corner drugstore, except you tweedled it in over the phone), where you take lessons, rent terminals, copy program cassettes, time-share, say hello to your friends. The terminal cluster at tomorrow's computer store may be like the pot-bellied stove in the general store of a generation ago: a place where you swap gossip and whittle, even if the whittling is done on a data structure rather than a physical piece of wood. When they get their act together, such stores will probably appear in franchised chains that expand like McDonald's and 7-11.

What is not generally recognized is that there will be considerable overlap between home and business systems: my estimate is that they will have about 70% of their programs in common. All comers need text handling, retrieval, scheduling, financial planning, bookkeeping. Home users, however, do not need order processing and inventory; businesses do not need games.

These developments will cause convulsive changes; not merely among users, but within the computer community itself.

Now the programming community contains many Good Germans, doing what they're told and not wondering about it — as well as, perhaps, a few others, who enjoy oppression and know full well what they are doing.

But when people at large begin to find out how basically simple computers are, how easy and how incredibly useful in their everyday lives, it will not necessarily be handshakes all around. I think it likely that there will be a lot of anger, and a lot of hard questions asked. And then the people who made computers oppressive, as well as artifically and intentionally complicated, are going to have a lot of explaining to do.

This may lead to a period of agonizing reappraisal and collective guilt not unlike what followed World War II, when the Good Germans had to walk through the concentration camps and confront what they had been contributing to. Some computer people themselves will come to feel that they have been living a fraud. Combine this with sudden unemployment across the field, as fewer programmers are needed, and we are going to see suicides, weird religious movements, and perhaps strange political developments among the mortified castoffs.

The good goals are still attainable. On our way to a happier world, a better world, a more knowing world through computers, certain unfortunate circumstances have arisen. We can find our way out of them. There will be painful dislocations, but we can get there: a world where the messy crud is taken care of automatically, and information comes to us as, where, when and how we want it. A world with a lot more knowledge spread around in it and a lot more fun.

And it may be that access to information, *real* access under people's total control, may yet make this a better world, may reverse the tides of apathy and illiteracy that rise daily.

Great changes are in store.

A spade is a spade.

The emperor has no clothes.

The true frontier is not technical complication. It is simplicity and clarity.

The human oppression and degradation of the first computer era are coming to an end. The new age of computing will not build on the past, but repudiate it. We — peoplekind — could have used the last ten years. But let us see what we can do in the time that remains.

PERSONAL DIGITAL SERVICES

The true and principal use of computers has never been understood by the computer industry. The computer is properly a tool, toy and environment for individuals. In this it resembles the automobile. But the computer industry has overlooked its ultimate market for thirty years, from 1945 until 1975. It is as if the motor industry had produced trucks and busses for twenty years before it built its first car.

It is my general prediction that by the year 1980 there will be programmable computers in some ten million American homes, and in another ten years, substantially all. (Because some editors have considered these figures outlandish, I am particularly concerned to get them into print.) The present ten thousand or so computers now in American homes represent only an infinitestimal fraction of what is to come.

This of course means a revolutionary market and social change comparable to what happened earlier with the telephone, radio and television; by comparison, CB and digital watches, pocket calculators and video games dwindle to a significance comparable to the hula-hoop.

Versions of this piece have also appeared in *IEEE Computer*, March 1977, 53-4; and McCabe (ed.), *PCC's Reference Book of Personal and Home Computing*, (People's Computer Company, 1977).

What are the uses of the home computer? There is scarcely any use of the computer that someone will *not* want in the home. But delineating the principal markets is something else again.

The distinction between "personal" uses of the computer, and whatever other kinds there may be, is not a sharp one at all. Artistic and frivolous uses, leisure and self-improvement come first to mind. But scientific and business applications come right behind: there are amateur scientists, and professional ones, who would like to work at home; many people have businesses in their homes, or would like to if the paperwork could be eliminated. Thus in principle there is no exact distionction between personal and nonpersonal uses, and all the types of software that have evolved for other purposes will eventually turn up in homes.

Many things will fuzz the line between hobbyist or amateur use, on the one hand, and business and professional use on the other. A large proportion of personal, non-business use will be closely related to business use. For instance, a collector of stamps or antiques will need virtually the same software as a museum, or dealer in such objects: keeping track of objects' descriptions, origins, price, time of acquisition, and so on. Services which are now unusual and innovative in industry will become, as their prices drop and they become simple to use, desirable for many home users. These include digital music synthesis and recording, and, soon, computer-based motion-picture editing; later, perhaps digitized photographic archives and laboratory services.

But such applications will not be the center of use. I see the personal use of computers as having four basic types. First, of course, numerical uses, for personal figuring and bookkeeping, automatic accounting, tax preparation, financial projection and the planning of financial resources. Second, information retrieval and trackkeeping, the filing of whatever information the user wants to keep. This includes inventories and catalogs on the one hand, but in another direction can become a sort of "family Bible" of personal historical information. Third, text and word processing, including correspondence, general writing, recipes, diaries and "guest books." This may extend even to advanced systems of the Thinkertoy class, allowing the intercomparison of complex alternatives that are being considered. (The advanced text systems of course merge with the retrieval function.) Finally, games. By this I mean games in the larger sense. Today's computer games, such as the many available in Basic, tend to be zero-sum competitions or simple brain-teasers and explorations of certain complex events (like *Lunar Lander* and *Kingdom*). Though these can be quite worthwhile, tomorrow's games will do much more.

The home computer revolution will not happen all at once. We may distinguish several eras of the evolving personal market.

It is in the first era that we now find ourselves: a market of electronics hobbyists who want to assemble their own computers, and inexperienced persons fanatical enough to try because they want computers so much.

A second era will begin shortly, however, entirely different from the first. Complete and reliable machines will soon come on the market, complemented by prepared, easy-to-use software for a spectrum of users. (Several such machines are reputed to have arrived; the software has not.) Such complete turnkey packages will change the market emphasis from the present nuts-and-bolts "reality" of the computer to its software *virtuality*.

The new users will have little familiarity with hardware, or even with machine language, and will program in high-level languages or not even program at all, being happy to use the new programs that come packaged. (As in many other cultures, the newcomers with their lack of ruggedness will be greeted with disdain and resentment by the old-timers who arrived six months before.)

Third and finally, the mass market era will arrive when computers can be purchased as easily as cassette recorders and programs as easily as pre-recorded cassettes for them. Demand will abruptly rise into the millions, the stock market will go wild, and so on.

The explosion of little computers does not mean, of course, that there will be no external or centralized services. Quite the contrary. Despite the proliferation of small inexpensive computers, there will be several reasons for offering personal services external to the owned machine, or in complement to it.

¶ Cost. Particular hardware may cost too much.

¶ Other lacks. The hardware may be broken, or unavailable, or its acquisition postponed.

¶ Networking. For many applications one user wants to be in touch with others.

¶ Social aspects: the user may want to be in the same room, say, with other game participants.

¶ Equipment multiplexing. It may not make sense, or even be faintly desirable, to have what you only need part of the time: large disk, or tape, or printer. (The multiplex advantage has long been the excuse for time-sharing, that it supposedly did not "make sense" to have your own computer. This argument is still valid for other equipment, and some uses of time-sharing.)

It is quite likely, indeed, that the personal computer market will expand both in purchased equipment and in ancillary services; the personal services to be offered will in many cases parallel, and in many cases extend, the software to be offered for small machines. (However, individuals will probably show a much greater reluctance to spend a computer dollar on evanescent services when it could go for permanent equipment.)

Corresponding to the three market eras, we will see a succession of user services offered. All are simultaneously viable and will eventually coexist.

In the first stage, we can expect to see simple services for which individuals can't afford the equipment. These include, in particular, printout and storage. We may expect users, for instance, to bring data cassettes to their local computer store, to have their correspondence printed, much as people bring film to be developed at the drugstore.

Those users who cannot afford mass storage devices should be able to dial up their local computer store and load a program into their own computers over the phone. Or, after finishing some sort of work — a data set, a piece of writing, or a graphical work of art — send it over the phone for storage to their local service.

Lastly, low-cost time-sharing, offering restricted facilities and simple languages, should have a ready market for hobbyists — when the price comes down to reasonable levels, like two or three dollars an hour. Walk-in time-sharing parlors will appear, with terminals amid tasteful decor. Particular time-sharing services will be built around this market: for instance, programs sectioned to be downloaded rapidly from the central system, chain to one another, and accept data from the user.

The computer store, and later the time-sharing parlor, will at first have something of the atmosphere of the old country store, with comrades chatting around the pot-bellied stove; though this will change as the population of users expands and becomes less exclusive.

In the second era, user services will begin to appear that have nothing to do with the usual ideas of the computer.

Scheduling systems of the PERT and Planalog type will become easy to use. Advanced Thinkertoy systems will become available.

Advanced games of real complexity having a certain social profundity, some with graphics, will become available. Two examples of such games are *Diplomacy,* by Allen B. Calhamer, and *Dungeons and Dragons,* by Gary Gygax and associates. The former consists of complex non-zero-sum bargaining in an intense social contex, largely simulating World War I. The latter is a system for imagined adventuring in an imaginary castle, with the referee as semi-opponent: a group of players, representing a band of explorers, wander through the castle according to the rules of the referee. Each player may do anything he wishes; when a crisis occurs, such as an encounter with a dragon, the player chooses his weapon and the referee tells him the result. (While "D&D" has been implemented in a number of computerized forms, so far these have reduced it to a mechanistic game of exploration and conquest, without the intense social setting that gives the game its fire.)

In the third era, user services will proliferate and there will be a jungle of imitations. There will be considerable differentiation for users of different interests. There will be quiet time-sharing parlors and noisy ones, cool sophisticated parlors and those with a rowdy atmosphere devoted to more churlish activities. The close social interaction of the computer store and time-sharing parlor will become more distant, as patrons and proprietors come to have less in common.

More sophisticated applications will spring up. Low-priced library systems will appear, allowing users to read, annotate and anthologize whatever they want from an ever-growing body of instantly-available documents. Teleconferencing schemes will become widespread. Extremely sophisticated games will appear, but these should more properly be called "games, adventures and simulations" — libraries of programs ranging from talking personoids, such as *Eliza*, to comic strips in which the reader may take an active part. We will see large-scale social and international simulations through which users may experiment with the world's destiny.

The exact consequences of all this are impossible to delineate right now. But some of us, at least, are motivated by the belief that these developments will contribute significantly to human understanding and wisdom.

BIBLIOGRAPHY

Nelson, Theodor H., *Computer Lib*. Ted Nelson, Publisher; distributed through Hugo's Book Service, Box 2622, Chicago, IL 60690.

THOSE UNFORGETTABLE NEXT TWO YEARS

Here we are, at the brink of a new world. Small computers are about to remake our society, and you know it. I am supposed to tell you what is about to happen in the near future.

But to understand the future we must understand the past: most people don't realize what *has* happened. What is astonishing to me is not so much the future as the past, and the things that are going to surprise people — the sudden appearance of little helpful interactive computers everywhere — should be less surprising than the past circumstances that have delayed all this till now.

This is the "official" text of a banquet address presented at the West Coast Computer Faire, April 16, 1977; as prepared beforehand for the Proceedings of the West Coast Computer Faire. Tapes of the talk are available from Butterfly Media Dimensions, San Francisco.

We have inherited some silly myths about the nature of the computer; and these myths are miraculously shared by both computer-lovers and computer-haters. They are the silly and cruel myth of depersonalization — that computers demand impersonal behavior and rigid unpleasant rules, not to mention punch cards; the myth of efficiency, that using computers in depersonalized ways is somehow good; and the myth of technicality, that to criticize to try to appreciate computer systems is beyond the competence of the layman. These vicious myths, and more, imprison us all.

Many computer people seem content with these myths and — intentionally or not — keep them going, dealing with non-computer people ominously and unhelpfully, with hardheartedness, scientific pretension and scary lingo.

We all know that there are those in the computer profession who enjoy pushing other people around, who hide behind the computer and say the pushing around is the computer's fault. The computer is functioning as a mask. There are even those with long hair who teach computer science and talk about computer liberation, who yet enjoy mystifying and confusing people and leaving them helpless. Worst, I regret to report that there are actually people who enjoy pushing people around, and intimidating them, in the amateur computer field.

C.P. Snow, the English author, has spoken of what he calls "the two cultures" of educated people, one being those scientifically trained and the other being the humanists. Ordinarily these two factions do not speak to each other, each regarding the other side with distrust and disgust. Computer people continue this tradition. Many computer people, indeed, seem to half-desire some kind of a showdown fight, where they are going to rub the humanists' nose in computer jargon, possibly to get even for having had their spelling corrected.

Do you think it's not as bad as that? Consider the problem of lower-case lettering for display and printout on computer terminals. Some computer people feel there is no need for lower case. This is a slap to all lovers of the written word. If you have now lower case on your terminal, it assures that no person with a literary background will ever use your text editor.

Or consider an argument that recently occurred in one of my seminars. A student of mine is planning to implement a text system along certain lines I advocate. But he insists that the user, typing in text, end each typed sentence with a special character, such as the caret. When I pointed out that standard practice for both stenographers and authors is to delimit a sentence with two spaces, he was indignant. "I don't want to have to type two spaces after a sentence!" he declared, preferring instead his caret; it did not matter to him that literary people would hate the caret more than he himself hated the two spaces, and so they wouldn't use his system.

With these points I have intended to highlight several problems. Computer people are not all of them prepared to honor or appreciate the wants and needs of others, especially those with different points of view. And this has had, and will continue to have, curious repercussions. Old style computers are about to collide with new style computers, and many people will get caught in the middle.

Here is what has happened so far. In January of 1975, MITS announced the Altair computer kit for $400. That seems a long time ago: in the two years since then, so much has happened. To the surprise of some people, the Altair computer took off like a rocket. Certain other people, who had been almost ready to do the same thing (they say), were galvanized into action. More brands appeared.

The next major event was Lois and Dick Heiser's opening of their Computer Store in Los Angeles. "Now why didn't I think of that?" was everybody's first reaction, followed instantly by, "I still *could*."

So here we are. Perhaps twenty thousand dinky computers, perhaps more, are in the hands of small businesses and hobbyists. (Many hobbyists, too, have visions of becoming small businesses.) Some three to five hundred computer stores are open, or on the verge of opening, or in some stage of near-existence. And throughout the country are the cottage computer industries: people making accessories, software, teeny mainframes. Catering to these mad folk are half a dozen — probably a dozen by the time this comes out — counterculture and hobbyist computer magazines.

Now, as far as I am concerned these developments are no surprise at all. The surprise is that it took so long to get started.

Certainly the computer-on-a-chip could be seen coming ten or twelve years ago. Now some people call them "microcomputers." But nothing could be more absurd than calling them "microcomputers:" a radio is still a radio if it is built from integrated circuits, not a "micro-radio;" a tape recorder or a hearing aid or a telephone is not a "micro-tape-recorder" or "micro-hearing-aid" or "micro-telephone" even if it is built from integrated circuits; and a computer is a computer, whether or not it is built from integrated circuits. *The real meaning* of calling it a "microcomputer" is this: the word is a face-saving device for people who did not see any of this coming, who had their heads in the sand. Calling these things "microcomputers" implies that they are something new under the sun, a technological surprise which could not possibly have been predicted, so the term absolves the speakers of having been fatheaded in expecting the status quo to continue. The surprise is all in their minds.

The dinky computers, as I prefer to call them, are widely belittled in the regular computer industry. But our fundamental 8080 chip, suitably supported, is approximately comparable to the 1401, which was the workhorse computer of American business in the early sixties. For many purposes the 8080 is more than adequate. It's always nice to have more computing power, but then it's *always* nice to have more computing power. There will come a day when the power of a big PDP-10 will seem inadequate for a personal computer.

For now, though, the dinky computers are working magic enough. They will bring about changes in the society as radical as those brought about by the telephone or the automobile. The little computers are here, you can buy them on your plastic charge card, and the available accessories include disk storage, graphic displays, interactive games, programmable turtles that draw pictures on butcher paper, and goodness knows what else. Here we have all the makings of a fad, it is fast blossoming into a cult, and soon it will mature into a full-blown consumer market.

FAD! CULT! CONSUMER MARKET! The rush will be on. The American manufacturing industry will go ape. The American publicity machine will go ape. American society will go out of its gourd. And the next two years will be unforgettable.

The exact sequence of events is of course hard to predict. The general outline, however, is not mysterious. Let me try to fill in some of the basic things I think are going to happen.

In 1977, tens of thousands of computers will be sold to individuals; the current demand level, with machines sold as fast as they come into stock, will probably continue for some time, possibly several years for the better machines.

The sales of the S-100 machines, like Altair, Imsai, SOL and Polymorphic, will continue strongly. If you've followed the computer field for any length of time, you know that standardization is generally established on a defacto basis by whoever gets someplace first. It should have surprised no one that the original Altair system of interconnection at once became a standard. Many engineers have said, "Aw, I could do better," and proceeded to try, but that has been quite beside the point. When there is a standard way to do something, the possibility of a better way is often academic.

So we have seen a mass movement: Processor Technology, Cromemco, et al. have built boards for the Altair, and then their own computers on the same system of interconnection. Whereas little computers like the Sphere, the Jupiter Wave Mate, the Digital Group, indeed MITS' own 6800 machine — and conceivably the Heathkit computer that is to come — have ignored this standard at their peril, attempting to create a non-interchangeable market.

So far these little computers, from little manufacturers, comprise the "amateur" market. But next we will see the bigger manufacturers come rolling. Large-scale manufacturers will enter the game; some are already talking about building millions of units. And already there are "video games" actually containing a computer, such as the Fairchild for $150.

Soon the media will hear about it, especially magazines and TV news. They will go out of their minds, and give home computers the full-scale "mysterious phenomenon" treatment. They will make all those incredibly stupid jokes over and over, about "thinking machines," and about *1984*. But few of the journalists will bother to find out what computers are really about until everybody else knows already.

The funniest aspect of the press coverage will be, of course, that everyone they talk to will have a different story; it will be impossible for reporters to converge or focus. Some of the people interviewed will tell the press that home computers are an extension of ham radio (this has actually happened); others will seem to say it is an outgrowth of the space program or of video games, or of the New Math, or of pocket calculators, or who knows what. And they'll all be wrong because they didn't see that computers were for people in the first place.

After the media, of course, the stock market will react. Wall Street will behave as Wall Street always behaves in the face of a new consumer fad, and public corporations involved in home computing will lurch upward in price.

About this time it will become clear that home computers are actually as important as people thought video disks would be. Many people will try to get on the little-computer bandwagon in inappropriate ways. The cable TV operators, for instance, will blunderingly keep trying to set up some sort of a connection to home computers. There isn't any — except that they could use dinky machines to generate little animated cartoons and logos.

Now the really big manufacturers will come in. Philco and Zenith, Sylvania and Magnavox and the Japanese TV concerns, will bring out their home computers. Then will come the me-too electronics packagers, with a dazzling array of brand names offering small silly variations.

One problem for the larger organizations is that their decision processes are perilously slow: since the home computer market will change drastically every six months, the big slow companies may never get off the ground. They will be continually deliberating over how to compete with last month's product, and being dumfounded by the next development. However, some big manufacturers will swallow the little companies that are off to a good start, paying lots of money for the privilege.

While there will be many more small accessory and software houses, it will be much harder for the second generation to make it big; their best strategy will be to target and specialize.

Certainly the maturing markets for dinky computers will become ever more differentiated. *Beginners'* systems, whatever the chip, will usually come with Basic and game cassettes. *Office-standard* systems will probably become stablized on the computer-with-screen, like the SOL, Sphere or Intecolor; but interchangeability will be an important feature, possibly favoring S-100 machines. *Graphic* systems will become popular among artists and doodlers and playful people, and indeed among the well-to-do these may become a sort of animated decoration. (We will see them turning up all over retail stores, with sales messages and animated cartoons.) *Literary* systems for readers and writers will include disk, high-capacity display, and high-power text manipulation and filing programs — of a type, I believe, that nobody has seen yet. *Baroque* and *snazzy* systems will come out for technical showoffs: like the computers that have one of every processor chip, or multiple Z80s; some with lights and switches by the yard, others with lots of "computerish" lettering, military styling, futuristic spheroidal boxes and so on. Finally, perhaps most important, we will see the "nothing is too good for my kid" configuration, with graphics, extensive text handling, programmability and large memory; possibly musical input and output as well. (Kay's Dynabook, at Xerox Palo Alto Research Center, is the exemplar of this class — even though it may never appear commercially.)

As to manner of interconnection, alas, there will be all too many separate hardware worlds. There will be the ever-growing S-100 world, of Altair, Imsai and so on. There will be the nonstandard world. Some individuals will actually buy high-class hardware out of the professional computer world, which will be nonstandard as far as the hobby world is concerned. (In the next two years it is not clear that the entrenched minicomputer manufacturers can lower their prices, or adjust their thinking, enough to compete.) Then there will be the other chip computers, packaged with incompatible accessories, such as the Intecolor. Then there will be the discount packaging world, probably with more systems of incompatible accessories. The incompatibilities will be regrettable, but the marketplace will eventually punish this approach.

There will of course be a variety of other oddly-interconnected computer configurations. Certainly we may soon expect the TV with built-in computer. We may see the "smart hi-fi," a multi-room, multi-tuner, bus-oriented programmable system. And don't rule out the computer-in-a-van, with one monthly payment for both transportation and hobby.

Though it would be a good thing, it is doubtful whether any later and better chip computer will attain, in the next two years, the universality that the 8080 (and the Altair bus) have enjoyed. We *ought* to have a 16-bit amateur machine, or at least one allowing relocatable programs, which the 8080 does not. Indeed, there are several decent candidates, like the LSI-11, chip Nova and TI 990. But the central position of the 8080 will be hard indeed for another unit to capture.

In the software area, meanwhile, we will go from confusion to chaos; but perhaps after about two years it will begin to straighten out.

Utterly disparate software packages will appear, all proclaiming themselves to be Everything You Need. (There is one advocate of Tiny Basic who says numerical input is sufficient for "all software.") Programs will appear embodying the greatest possible variety of viewpoints.

There are many ways of thinking of software, and in the hobby world some of them are weirdly simpleminded. Many hobbyists come out of electronics, and insist on thinking of a computer as "a collection of switches." This is true in the same sense that a human being is "a bag of chemicals." But some hobbyists think, on the basis of this impression, that programs are little things, like ten or fifty lines long; like a binary-to-decimal conversion routine. From that follows the hobbyists' mystification by, and in some cases objection to, the copyrighting of programs.

It would seem peculair that people who claim to have compunctions about shoplifting, or the looting of parked cars, nevertheless proclaim — as some computer hobbyists do — that they have a perfect right to violate others' property rights in respect to copyrighted software. But we can understand this view slightly better if we consider that many electronics-oriented amateurs think programs are small and insignificant — of the level of complexity, say, of a limerick.

Many programs are short and simple. And many enjoyable computer games run in less than a hundred lines of BASIC. But tomorrow's home programs are not going to be just little programs. The important software of tomorrow will be immense turnkey systems of programs, comprehensive and interactive, for a broad spectrum of business and personal uses.

New business systems will wholly reverse the computer business systems of the past. The business systems of old, based on batch processing, created strange work procedures: transactions had to be keypunched and sent into the computer in long trains, to be processed one at a time by the same program. This created dehumanizing job definitions, such as keypuncher. Now, however, clerks will be able to process business paper one piece at a time, in the order of its arrival, and businessmen will discover themselves in control of their firms once again. But the programs will be big.

There will also be large-scale programs for the new three Rs — Readin', Ritin' and Retrieval. These will be immense, overlaying from disk.

Whole new big systems of programs for gaming and simulation and animation will likewise be swapped and overlaid from disk.

The new big software packages will have to do a lot of storage management — overlays, swapping and housekeeping. Once such a system is started, it will not need reloading; it will roll and undulate from one function to another at the user's whim, swapping and storing (fail-safe) and displaying interactively. These features do not come easy; they involve a lot of programming.

Now not everybody may want or need such extensive services. Some hobbyists, indeed, may also enjoy the ritual of "running separate programs" — toggling in bootstraps and running in loaders and programs for each new activity, in an anachronistic and playful simulation of batch processing.

But the big and sophisticated software will be to the little toy programs as *War and Peace* is to the limerick. The really good systems will be unique and distinguished, even works of art. People will devote months and years to their creation, and they will have considerable commercial value — being leased to businesses, say, for hundreds of dollars a month.

Speaking as a prospective supplier of such advanced software, the amateur market tends to interest me not at all. I see little reason to sell to hobbyists, and considerable reason not to. For the revenue of one well-behaved business customer, it might be necessary to deal with ten belligerent and disrespectful amateurs, some of whom have given notice of their intention to break any software contract.

In the ordinary computer world, the most emotional isuue is people's preferences with regard to computing languages; copyright is accepted as an ordinary part of life. In the amateur world the most emotional issue is that of copyright, with languages second. My own feelings about copyright are very strong: copyright is one of the only ways the little guy can get a leg up — you will excuse me for oversimplifyıng — against the Big Guys and the Bad Guys. I publish my own book, *Computer Lib,* and derive a modest income from it. Now if there were no copyright protection, anyone could print *Computer Lib* in Taiwan, and sell it for less than I get, and I would get no benefit for the years of work I've put into it. But the law of copyright, a Federal law, says the book is mine, and that people can only make copies with my permission; I like it like that. The same law, now, is extended to computer programs as a form o writing; and the same protection is available to you, practically free of charge, for the work that you put in on programming.

Anyone is of course free to give away a program he has made, since the copyright presumably belongs to him. And people to whom software means *small* programs of their own concoction will be happily giving them away.

But people who are working on big programs — programs involving tens of thousands of lines of code, programs that take months or years to create, will be very concerned for their copyright protection. To further this protection, we will begin to see software disguised as plug-in hardware, software coming on mystery tapes in unknown formats, software disguised in every possible way. This cloak-and-dagger approach is quite regrettable, but may be the only way to create an orderly market. I think there will be a period during which the good software, the big and serious systems, are held off the amateur market and made available only to commercial customers. But eventually, perhaps at about the end of the next two years, we will see bargains struck up with well-behaved individual users, and a system of safeguards hammered out to make an orderly market possible. These safeguards may even include program-readable hardware serial numbers, just as used in the bigger systems.

Languages. Just as in the big computer world, in the dinky world we also hear hotheaded arguments about programming languages. Computer people always become very pugnacious about their favorite programming languages — especially if they only know one or two. But the dinky-world arguments about language are a travesty on those same arguments in the straight computer world. The same honors are here attributed to BASIC that more knowledgeable computer professionals attribute to Algol, or SNOBOL, or PASCAL. (In the hobby world, one actually hears people with Ph.D.s, who certainly ought to know better, arguing very angrily that Basic is the ultimate.)

By two years from now, after a lot of frustrated effort and experience with big systems shoehorned into little computers, people will have realized that for powerful software we need good and powerful languages. They will also have found out that these must be *structured* languages, in the Dijkstra sense, which make programming so much more manageable; and they must be *extensible,* allowing the rapid creation of new commands for particular ranges of purpose. This means that interpretive languages like LOGO, TRAC Language* and SMALLTALK will assume new importance, as will such compiling languages as FORTH and "C".

One of the great strengths of the extensible interpretive languages is the degree to which they simplify *big* programs. My colleague William Barus and I recently created an animation program for the VDM board using TRAC Language; the program is about 16K in size. TRAC

*TRAC is a registered trade and service mark of Rockford Research, Inc., Cambridge, Massachusetts.

is not as easy to learn as Basic, and for small programs may seem rather clumsy. But as programs grow larger, its intrinsic orderliness keeps programs manageable. (The 16K program was written in about three weeks, and seems to work quite nicely, allowing both frame-by-frame animation and subpicture overlays.) With unstructured languages like Basic, programming becomes exponentially more difficult as the size of the program grows; with structured languages, the growth in difficulty seems to be logarithmic.

Basic will remain the language for beginners, unfortunately; no standard Dijkstra-structured upgrade is presently defined for it, although a "structured Basic" has been under design where Basic began, at Dartmouth. (But for those who want to sell language processors, there is a simple secret; hobbyists want any language as long as it's *called* Basic.)

It is a great pity that hobbyists are so insistent on talking about the physical box of the computer, and so resistant to learning about the issues and depths of programming languages — even though all their programming difficulties stem from these issues and depths. Meeting frequently with computer hobbyists, it is astonishing, and depressing, how often one must repeatedly answer such questions as "What is a computer language, anyway?" and "What is structured programming?" and "I don't see why it can't all be done in Basic." There has got to be some form of general consciousness-raising in this area.

In any case, when the smoke clears, about two years from now, hobbyists will begin to realize that the more advanced languages — at least the ones that fit on the dinkies — make programming much, much easier.

Such languages — like LOGO and SMALLTALK and TRAC Language — will gradually assume primacy. But we may hope that their being "advanced" languages will not make people think they are unsuited to beginners. On the contrary, structured extensible languages are the right way to learn programming. For instance, at MIT's LOGO project, they tried teaching programming to kids two ways: with BASIC and with LOGO. After a few weeks, I have been told, the kids who had started on LOGO were hopelessly far ahead, and, it is said, did not want to speak to the BASIC-trained kids because they thought the latter so ignorant and incapable.

No one who is really interested in computing languages would claim that the "ultimate" language exists; but there is a new generation of languages shaping up now, the "actor" and "agent" languages, whose proponents expect them to be as far ahead of LOGO, TRAC Language and SMALLTALK as these are ahead of Basic. Which of these advanced languages will fit on little machines is another question.

The merchandising of little computers will gradually break out into a number of separate sales approaches, much like the merchandising of anything else. The mass-market computer will of course be sold by the corner electronics discount house, in a fancy package. You'll get the instruction book and an impatient rundown, but no personal help. In the box you'll find the computer, an instruction book, warranty card, and list of repair centers. Perhaps also one free game cassette.

This mass-market approach will probably go two ways: toward the non-compatible, as with the Fairchild computer-game tapes, which will be a dead end; and toward compatibility, as with systems offering Basic and a standard cassette interface.

For more serious users, and for the S-100 world, the computer stores will continue; but they will become a cross between the hi-fi store, camera rental house and laundromat.

A viable take-home rental market will develop, presumably as a part of the computer stores. There you will be able to rent simple standard units (as you might rent a standard car), or, later, more esoteric units (in the way that professional motion-picture equipment is rented).

Repair steups will blossom. There may appear franchised computer-repair chains, similar to the automobile-transmission repair centers. But these will be walk-in centers; cheap computers will not be eligible for house calls. (Many computer-center people, who are used to repairmen coming *in,* cannot visualize throwing the computer on the front seat of the car and driving it to a repair center, as you do with an amplifier. but that's how it'll be.)

The amateur used-computer market will come into full swing. We will see the more fanatical hobbyists buying not just used Altairs but commercial minis and old "big" machines — real 360s and 7090s — and real antiques and fond replicas, such as Eniac and TX-2. (We may yet see a fully-loaded Imsai go for more than a small 360, just the way in 1974 you saw small used cars selling for more than big ones.)

A variety of personal services will appear, probably in conjunction with the expanded computer store. Such services will include program storage and printout; soon, time-sharing parlors, having clusters of terminals which may be rented by the hour. (This is the laundromat analogy — renting a terminal by the hour is not unlike renting a washing machine.) In the next two years, we should see services grow from the simple ones to a dazzling variety of new ones, such as advanced text services and retrieval, digital music synthesis and movie-making, and great libraries you can reach through your home screen.

The prospect of great on-line libraries raises great questions about truth and freedom — for which there is no room in the present talk, but about which much more will be heard.

I have elsewhere predicted that there will be ten million computers in American homes by 1980. This may be a considerable under-estimate, in the light of American consumer contagion. It will not be long, in my opinion, before the home computer field becomes most of the computer field in absolute dollar volume. This will take longer than two years. But I would predict a hundred dollars per capita per year within a decade.

Pretty good so far, right? But I'm afraid it will not be all sweetness and light. We can expect the dinky revolution to have a convulsive effect on the computer industry, and on the society as well. First let us consider the industry.

IBM will be in disarray. Fine-tuned to a captive market having certain kinds of submissiveness, it is hard to suppose that their kind of sales, let alone their kind of computer product, will give them a ready entrance to the kinds of markets now opening. (Supposedly as a personal computer, they will push the 5100, actually a 360 in disguise; but they will probably not know *what* to do if someone puts out the S-100 adapter for the 5100, effectively offering a 360 with Altair accessories.) The jolt to IBM's product line of all these developments will be considerable. A loss of revenue for IBM, or at the very least a slowing of its growth rate, seems to me inevitable. This could be traumatic to the stock market and to other true believers.

Most of commercial time-sharing will be down the drain, being priced at the top-managerial level but only usable by technicians.

Organizations will be in internal turmoil, as they see kids doing with computers what their programming teams can't. And as the new availabilities reverberate, many internal goals and budgets will be shot to hell.

There will be mass layoffs in programming staffs, and this will be only a beginning. The skills involved in programming are comparable to those involved in automobile mechanics, and as the number of capable individuals increases, the pay levels will go down, perhaps dramatically. Indeed, the number of salaried positions will go down, because systems will need less and less tinkering to be made to do what you want. It will gradually be revealed that most programming has been make-work around ill-designed systems.

Between laymen and the old-style computer people there will be a lot of questions asked. "Why weren't computers easy to use before? Why couldn't I have had this sooner?" These are not pleasant questions to answer.

In the industry there will be finger-pointing. "Why didn't we realize this would happen?" A very good question indeed.

For computer professionals will be the last to see this coming. Professional programmers have lived in a world of account numbers and job submissions, core declarations and runs aborted by the operating system — silly complications that evaporate as soon as everybody has his own computer. Theirs is an awkward and inhospitable world that bears as much relation to the real nature of computers as a string of frankfurters bears to a living cow. The "Nature of the Computer," for many who have worked with it, has really been the system of bizarre bureaucratic project management under conditions of monopolistic mysitifcation. And so it may come to be revealed that the people who have worked with computers most have understood them least.

In some areas of the society there will be pandemonium, as the personal lives of a vast number of people — *especially* computer people, who thought change was their friend — are uprooted and retreched. These more general effects are hardest to predict. I suspect that the main public reaction to little computers will be a mixture of delight and anger — delight at the new fun and facilitation, anger that computers were ever allowed to create the harm and inconvience that they have till now.

Here is the final surprise. We the computer folk are not going to be converting the non-computer world to be like us, as some of you may have been hoping. Quite the contrary. We are going to be repudiating the computer past, resolving not to push people around any more, and finding out how to make computers easy and helpful to everybody.

MUSINGS

The true use of computers is, and should long ago have been, personal. Misled and perhaps occasionally malicious, the industry has ignored this as a possibility for thirty years; computers have been held back from a personal market for perhaps thirteen. But the movement cannot now be stopped.

Why has individual computer use been so retarded? Partly because of the nature of our society, and partly because of the trickery by which computers have been sold: *not* because of the nature of the beasties themselves.

The 360 drastically postponed personal usability. And in an unholy alliance with the big manufacturer, the computer centers long ago became entrenched power units especially concerned with preventing maverick applications, and with preventing any other departments — let along individuals — from getting a computer. Underhanded methods have been used, it is sometimes said, to get rid of people who advocated intereactive systems. The computer field was structured to lock out both the interactive applications and the people that wanted them, or might have.

Now the computer centers are on the defensive, publishing articles saying there is *still* a place for the centralized computer facility. They may be right, but they have an increasing burden of proof. As dinky computers and interactive styles of use take over more and more, little sym pathy is due to those who strove to hold them back.

But let us lift our eyes from the past, from the limitations and trappings of how you've had to do it. Rather than being obsessed with the styles of computers as they have been used, we need to consider, and design, the environments that we really want. I call this the Higher Virtuality. By virtuality I mean the effective environment we create with the computer, as distinct from the computer's "reality," the physical parts and program techniques that make things come out that way. The higher virtuality, then, is the computer environment we *should* strive to create. It is hard to make the choices and designs that this entails.

The virtuality that calls to us now, I think, is the new world of highly interactive systems. Cursors and panels, lightbuttons and menuplexes, maps and graphs and fast-changing screen layouts, are the pieces from which we will construct our new experience-spaces. Now, we tend to think of highly interactive computer systems — for text editing or retrieval or movie-making or whatever — as the ones with fancy screens and light-pens. But this brings up an interesting general point. The fancier super hardware isn't as important as the imaginative and artistic use of whatever we have.

Let's consider just the problem of pointing at things and controlling what's on the screen. Most of us don't have a light-pen or Engelbart mouse on our home computers; but if we are clever we can make systems highly interactive even if we just use keyboards. We can zip cursors around with the keys to select from menus. We can even whiz between environments. Suppose the interactive program can change the meanings of the separate keys dynamically: at one instant, specific keys can be arrow keys, to move a cursor about the screen; at another instant, they can type musical notes, or special picture-symbols; or anything.

This is not an unimportant matter. Some of the snazziest research setups have highly interactive keyboards of this type. Bitzer's PLATO system, for instance — a large special-purpose graphics time-sharing network run out of the University of Illinois. The main program can react to every key-press, and the keys can be automatically redefined under program control. The letter "D" on the keyboard can be an arrow when pressed at one instant, create a whole picture of a dog when pressed in the next.

This feature makes PLATO one of the most dramatic introductions to interactive computer graphics. Kay's Dynabook system, at Xerox, likewise allows a key-by-key dynamic reaction to what the user does. There is even a research setup with a "phantom keyboard" — its keytops have colored changing labels that you can read right through your fingertips. (This is done with a color video display and a semi-silvered mirror.)

Such dynamic keyboard capabilities, together with our screens, make highly interactive systems easy and exciting to create. (It is regrettable that most if not all IBM systems do not allow this dynamic key-by-key redefinability. IBM systems will not ordinarily respond to single arbitrary key presses; you have to hit either the Carriage Return or an "Attention" button to make the systems respond. Again, the interdiction of highly interactive systems. Was this malicious design?)

This key-by-key arbitrary response leads us, as do many other paths, to the real question. Given that we can make each key have any effect, then what? How should our systems behave? How do we make the interactive system make sense?

On the interactive screen we deal with a special new virtuality: architecture in virtual space, where anything can happen based on anything you do. The space has to be invented. This architecture of conceptual space, this conceptual architecture of screen-space — is a new realm where we are combining feelings and effects, as in a movie, and the need to be clear, as on a map.

The problem really becomes, I think, giving the system conceptual unity, which is just a more general example of the problem of giving conceptual unity to the interactive keyboard. How do we give the key assignments, and how do we give the larger system, conceptual unity? "Conceptual unity" is an extremely pliable concept. What many programmers think is conceptual unity, clear and simple, is not necessarily conceptual unity to the non-computer people.

The problem is very like architecture. Now, I'm no architect, but I have spent some time in buildings — indeed lived in them! — and I think there are several clear criteria for what makes buildings good. It's nice for them to look good, sure. It's nice for them to have good cost ratios for their builders. BUT I submit that one of the most desirable things about good buildings is for people to be able to get from place to place simply, and know where they are.

I recently taught at a great urban campus filled with grand pretentious buildings. The architect gave no mind to helping people find their way around. The buildings looked very Futuristic, but were incredibly confusing and inconvenient.

One evening, in the depths of one of the incomprehensible buildings, I had to mail a letter and make sure it went out in the morning. I went to the mailbox on the main floor, and there was a sign on it saying that after hours one was to use the mailbox on the *first* floor. (This was my first inkling that I was not already on the first floor.) Diagonally to my right was an elevator. I stepped in, pressed "1", and prepared to retrace my steps, walking diagonally back to my left to the first-floor mailbox.

I was faced with a cinder-block wall.

I walked around to the right, then finally found a back corridor going in the direction the mailbox should have been in. But now there was a succession of cinder-block cubicles and rooms. In none did I see the mailbox.

Eventually I was able to find the mailbox by a system of triangulation. By counting paces, and making a map with paces marked and a combination of ninety-degree and sixty-degree angles (of which the architect was fond), at last I found the mailbox in one of the rooms of the cinderblock environment.

Here is an example where an architecture of *real* space was created with no concern for personal clarity or orientation.

The problem is the same for creating interactive systems. Creating interactive systems, and their virtual spaces, should be an *art*. And the real kicker, I believe, is this: *all computer systems should be highly interactive.*

The confusions and oppressions of yesterday's computer systems vanish where the user can see his alternatives on a screen, get quick explanations, see maps of what he is doing, and get all the other helps that the interactive screens can provide. This means the continuing oppressions of yesterday's computer systems — even today's — are the opposite of the way computers ought to have been used all along.

We shall see. Time, and the marketplace, will decide. When people get to use interactive systems, they will be much less interested in either manual methods or batch.

But the central concern of highly interactive systems will be making things clear and simple. Now, making things clear and simple and easy to use is, I'm afraid, the opposite of what some computer people, perhaps some of you, want to do. So in an important sense, it is *not* the laymen who have to learn Computerese; in the greater sense, WE MUST ALL LEARN COMPUTER EASE.

The next two years will wreak extraordinary changes, and we will "computerize" society for fair. But it may not be what you computer-smart people expect.

What some of you have been considering "the new era" of home computers may correspond to the tin-can and crystal era of radio; and the convivial hobby you are part of right now may vanish like the crowd that welcomed Lindberg at Orly. They don't come out to meet the planes any more. Today's summer-camp camaraderie won't last forever, and the computer will probably become a home appliance, as glamorous as a canopener, within a couple of short years.

What then is there left to believe in . . .? To hope for . . .?

I suggest that we look to simplicity and clarity (to make people's lives easier) and to truth and freedom, for their own sake.

Thank you.

Ted Nelson holds degrees from Swarthmore College and Harvard. He has taught at Vassar College, The University of Illinois, and Swarthmore.

He is also the author of Computer Lib, *now in its fifth printing.*

A frequent lecturer, banquet speaker and consultant, Mr. Nelson is thought by some to be a spokesman for the coming social revolution of home computers and computer graphics.

$2.

". . . The question is not, Why have home computers come? The question is, What has held them back?"

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

"The sooner we understand it, the better."

"Computer screens on the kitchen table. Computer screens by the bedside. Computer screens on the office desk. Computer screens on the school desk. Computer screens on automobile dashboards, Coke machines, ticket dispensers..."

"The true meaning of interactive screen-systems is that people can do things easily and without confusion. Sitting at screens at home or the office, they will type — or point — and the system will respond clearly, with a clear virtuality."

". . . 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 book! 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 a new world, I have to fumble with words. . . ."

". . . Those who argue that things should be made difficult for some moralistic reason are not welcome in this book. May their corridors never end and their forks be too heavy to lift."

". . . So in an important sense, it is not *the laymen who have to learn Computerese; in the greater sense, WE MUST ALL LEARN COMPUTER EASE."*