COMPUTE!'s

COMPUTING TOGETHER

A Parents and Teachers Guide to Using Computers with Young Children

Fred D'Ignazio

A COMPUTE! Books Publication

$12.95

COMPUTING TOGETHER

A Parents and Teachers Guide to Using Computers with Young Children

Fred D'Ignazio

COMPUTE!™ Publications, Inc.
One of the ABC Publishing Companies
Greensboro, North Carolina

The following articles were originally published in *COMPUTE!* magazine, copyright 1982, Small System Services, Inc.: "Santa Claus, Subways, and Penguins" (April); "Architects of the Micro World" (June); "The Computer Playground" (July); "Build Your Own Computer Friend" (August); "The Talking Head" (September); "The Story Game" (October); and "Software, Sexism, and Other Topics" (December).

The following articles were originally published in *COMPUTE!* magazine, copyright 1983, Small System Services, Inc.: "A Computer Language for Kids" (February); "Catie's Christmas Card" (April); "Software for Toddlers" (May); "Turning Logo Upside Down" (June); and "Twilight of the Computer Hacker" (June).

The following articles were originally published in *COMPUTE!* magazine, copyright 1983, COMPUTE! Publications, Inc.: "*Sesame Street* and Interactive TV" (July); "Superbaby Meets the Computer" (July); "Beyond Computer Literacy" (September); "Islands of Learning" (October); "How to Get Intimate with Your Computer" (November); "The Electronic Chalkboard" (December); and "Winnie the Pooh's Alphabet Adventures" (December).

The following articles were originally published in *COMPUTE!'s Gazette*, copyright 1983, COMPUTE! Publications, Inc.: "Computer Adventures" (July); "Revolution in the Nursery" (August); "A Robot Inside Your Computer" (September); "From Dinosaurs to Freckles" (October); "Giants and Dwarfs" (October); "Computer Graphics: The Age of Electronic Art" (November); "There's a Creature in My Computer!" (November); and "Your Wish Is My Command" (December).

The following articles were originally published in *COMPUTE!* magazine, copyright 1984, COMPUTE! Publications, Inc.: "Computer Popcorn" (January); "The Robot Teddy Bear" (January); "Computing to Read" (February); "New Directions for Computer Camps" (March); and "Catie's Grandmother Goes to Camp" (April).

The following articles were originally published in *COMPUTE!'s Gazette*, copyright 1984, COMPUTE! Publications, Inc: "New Family Learning Games" (January); "The New King of the Mountain" (February); "Computer Show and Tell" (March); "Albert Zap, Won't You Please Come Home?" (April); and "New Standards for Home Learning" (May, June).

The following article was originally published in *COMPUTE!'s PC & PCjr Magazine*, copyright 1984, COMPUTE! Publications, Inc.: "Follow the Leader" (April).

Copyright 1984, COMPUTE! Publications, Inc. All rights reserved.

Reproduction or translation of any part of this work beyond that permitted by Sections 107 and 108 of the United States Copyright Act without the permission of the copyright owner is unlawful.

Printed in the United States of America.

ISBN 0-942386-51-5

10 9 8 7 6 5 4 3 2 1

COMPUTE! Publications, Inc., Post Office Box 5406, Greensboro, NC 27403, (919) 275-9809, is one of the ABC Publishing Companies, and is not associated with any manufacturer of personal computers. PET, CBM, VIC-20, and Commodore 64 are trademarks of Commodore Business Machines, Inc. and/or Commodore Electronics Limited. Atari is a trademark of Atari, Inc. TI-99/4A is a trademark of Texas Instruments, Inc. TRS-80 I and II and Radio Shack Color Computer are trademarks of Tandy, Inc. Apple II and IIe are trademarks of Apple Computer Company. PC and PCjr are trademarks of International Business Machines, Inc. NEC is a trademark of NEC Home Electronics USA.

Contents

Foreword

Kids love computers. We've all heard that dozens of times. But sometimes adults are more hesitant to join the computer revolution. Perhaps you feel that way. How can you keep up with your children's computer experience, and most important, how can you turn computer learning into a family affair? There's so much to learn. How can you best decide how to use computers to help your children learn, be creative, and have fun?

Computing Together: A Parents and Teachers Guide to Using Computers with Young Children is a collection of columns written over a two-year period by COMPUTE! columnist Fred D'Ignazio. Fred answers the questions that parents and teachers ask about computing with children. In a light and easy-to-understand style he discusses topics that parents and teachers will find interesting and informative.

Fred knows what questions to ask and how to answer them. With two children, eighteen computers, and seven robots, he has a wealth of personal experience to draw upon. As early as 1974 he was toting a portable terminal and modem to kindergarten classes. In those ten years, he's written 15 books about computers, robots, and children.

Fred knows how to help you move into the computer age. After reading this book, you might soon feel like welcoming a computer into your family.

Chapter 1

Introduction

Santa Claus, Subways, and Penguins

Close your eyes and imagine the baseball diamond at Yankee Stadium. The diamond has four bases. This book also has four bases. What do the four bases represent? First base is Fun. Second base is Learning. Third base is Kids. Home plate is Computers.

In the middle of the baseball diamond is the pitcher's mound. As author of this book, I am like the pitcher. My ball isn't made out of cork or cowhide. It's pure fantasy. It is the world inside the computer.

This book will explore the many ways kids can use computers to learn and have fun at the same time, on their own and with their mom, dad, or teacher. It will focus on ways computers can be used to foster learning for the benefit and enjoyment of both children and adults.

Santa and the Penguins

Our society is feeling the impact of a computer *implosion*. It's as if Santa Claus' bag burst as he flew across the world, and the presents are tumbling to the earth, ending magically under everyone's Christmas tree. And all the presents are small computers. And it is Christmas all year long!

Has your computer just dropped down through the chimney and bounced into your living room? If so, prepare for a knock on your door. Answer it, and you will find that your front lawn is overrun with experts frantically trying to attract your attention. "We are ready to advise you," they say. "We can introduce you to guidebooks, cookbooks, checklists, disks, and cassettes — whatever you need to operate your home computer."

By all means, let the experts in. And listen to what they have to say.

But don't expect to find an expert here in this book. I'm no expert. I'm not an educator. I'm not a psychologist or a game whiz or a scientist.

Introduction

Who am I?

I'm a writer, and a storyteller. I'm also a parent — of a little boy (Eric) who's almost three, and a not-so-little girl (Catie) who's just turned six.

If I'm no expert, why do I think I can climb on my soapbox and tell you things about computers?

First, because of what I love. I love kids, fantasy, fun, learning, and computers. I think I have built a book around these five loves, a book that will be interesting to anyone who shares my affections.

Second, have you ever seen a penguin rookery? It's a small island or hunk of rock near the bottom of the world, on or near Antarctica. Its most distinguishing characteristic is that it is crowded, packed, jammed with penguins. Certain applications of small computers resemble a penguin rookery. There are experts and so-called experts crawling all over, bumping into each other, stealing each other's rocks (for nests), and occupying all the free space.

A penguin rookery is a good place to read about, but I have no intention of visiting one and pretending to be just another penguin (as Jacques Cousteau once did). The same goes for this book. In this book, we (you and I) will search for islands that are less crowded. We will look for computer applications which are vital, but which have not yet received a great deal of attention.

Uncrowded Islands

What are some areas we might explore?

First, *computers for little kids*. Let's say, arbitrarily, from age two up to age eight. We have all seen articles, games, and programs aimed at this group. But not so many as at other groups. As a father and lover of small children, I'd like to explore some new applications for computers here.

Second, *education at home*. Computers are zipping into schools like commuters riding a New York Subway Express — quickly and in great numbers. To really fit in, the computers have to adhere to the schools' curriculum. That means programmed learning, by the experts.

But how about education at home? Self-motivated and self-directed education? Education without a formal curriculum. Education without a game plan or an expert peering over the learner's shoulder. Education that is not just alphabets, multiplication, memorization, and drill. This is another area that fascinates me. I'd like to focus on it in this book.

Third, *fantasy*. The computer playground. A place for kids to act like monkeys and develop bulging muscles of the imagination.

Fantasy is the world of kids. It remains their world until they have heard enough facts and enough drivel to drive fantasy back — back into their mind's back burners and dark corners.

Computers are an immensely powerful tool of directed, solo, or group-oriented fantasy. Just witness the enormous popularity of electronic games.

Fantasy is natural for the child. It has a galvanizing, emotive, and energizing effect on whatever the child does.

A personal computer is like a wizard's staff, a magician's wand. It is a powerful tool for fantasy. It is a tool for the gods, and the gods within us. It is creativity unbounded. According to computer philosopher Greg Yob,

> If you can program your computer, here is a tiny universe in which you can be God. Within the realms of expression that the computer can provide, you can build a world, define its laws, and watch the universe unfold. As your whim dictates, you can intervene at any time, and if you desire, the history of the universe can be changed and rewritten at will. Such a power this is! *

**Gregory Yob, "The Computer as a Gun: Personal Computers and Personal Autonomy," NCC'79* Personal Computing *Proceedings, NY: American Federation of Information Processing Societies (AFIPS), p.9, 1979.*

Yet as they become increasingly popular, small computers are also becoming more focused, more regimented. They are becoming big business. They are becoming standardized and institutionalized.

But they are still ripe for fantasy. Safe in the confines of your own home, where you and your kids are the kings and queens of the realm.

A large dose of fantasy can make your child's use of the computer more playful, his or her learning more creative, more effective and long-lasting.

Fourth, and last, *games*. What kind of games? It doesn't matter, just so they're fun and produce a positive, constructive effect on the young game player.

Welcome to D'Ignazio's Game Arcade

I have eight programmable computers in my home. Running on those computers are over 200 different games.

I have opened the doors of my home to neighborhood youngsters. As a result, my home has become the local videogame arcade.

It's an offer the kids can't refuse: They get to play games, hour after hour, and it doesn't cost them a single quarter. Needless to say, the parents are delighted with this arrangement. And I get lots of contact with young people, which I enjoy and which helps me with my books.

On any afternoon, there are usually three or more young people present, playing "canned" games or inventing new ones. The young people range in age from two (my son) to seventeen. Most, however, are between eleven and fourteen.

Afternoons are noisy in the D'Ignazio Arcade. There are squeaks, giggles, beeps, and booms. In the midst of all the silliness and fierce competition, though, two things are apparent. First, the kids are having fun. Second, they are learning.

During the course of this book, I will write about the many ways kids can use computer games to learn and have fun. I think it will make interesting reading.

Adventure, Oracles, Picture Books, and Turtles

So now you have it. I've emptied my pockets and dumped the contents into your lap. As a result, you probably have some idea of this book's flavor.

Here are some of the topics we are going to explore:

If Your Teacher Were a Turtle

Using turtle robots that live inside the computer to teach young children reading, writing, programming, directionality, etc.

Alice in Computerland

A visit to the world inside the computer. How the youngest children can learn the basics of computer hardware and software. Computer literacy for toddlers. Computers as a second language. Computers as a new mythology.

Robots, Games, and Learning

A special chapter on using robots and robot games to teach things to kids. Kids love robots. I'll bet you do, too.

Toddler Adventure

How to wean your toddler from her blanket or bottle and turn her loose on a computer. How to launch your young children on their first adventure—an exciting, educational experience.

The Computer Picture Book

The electronic book is on its way. It will come in the form of a microchip, ready to plug into your *book player*. But until it arrives, you can create your own books on your home computer. This book will show how you and your kids can create electronic picture books.

Building Models

Kids can fire questions endlessly at you. You try to answer them, but you suspect that your answers flit like butterflies into their ears, ricochet around a bit, then flit back out, only a moment later.

We all learn things best by doing them. How did you first learn about people and their bodies? Did you have a doll or a teddy bear? How did you first learn about automobiles, monsters, trains, airplanes, and spaceships? The way I learned was by building models.

You can answer your kids' questions with models — *computer* models. You don't build them using paper, plastic, or glue, but by creating simulations — miniature replicas of creatures, things, processes, or events pulled from the real world.

But why stick to the real world? Why not copy something directly from your child's dreams or imagination? With a little ingenuity, you can probably build a model of it on your computer. Then your child can run the model, change it, or add to it. Or replace it with something else.

A New Member of Your Family

Our family computer (the oldest one) is named Ged, after the wizard hero of Ursula LeGuin's wonderful *Earthsea Trilogy*.

When Ged first arrived on our doorstep, he was a dull and simple-minded character. He knew how to edit text, save and copy files, things like that. But that was about all.

So my kids began to teach him. They imagined what kind of personality he ought to have (wise but mischievous and tricky), and we gradually breathed life into what was once a dry and pedestrian computer.

Now we treat Ged like a member of the family. He has his own jokes, his own riddles, tricks, favorite expressions, and peculiarities. He is very much like a real person. That means he is constantly learning — and my children are his teachers.

This book will offer suggestions on how to turn your computer into a member of your family. Or several members. After all, just like a real person, he or she can have many faces.

On the way, I guarantee you, your children will learn many things about computers, programming, intelligence, and personality. And about themselves. In a sense the computer becomes a mirror — it reflects the kids, you, and your entire family.

Making Your Computer More Real

Have you ever read *The Velveteen Rabbit*? It's a story about a nursery toy — a little stuffed bunny — who becomes real to the child who loves him.

Kids tend to anthropomorphize everything. They see a person, a spirit, a gremlin, or creature inside or behind everything that exists. Ironically, this fantasy image of the world seems to make the things in it more real.

Computers can easily become more humanlike and more real. We can program a personality into them. We can add a voice synthesizer. We can attach a speech-recognition device. There are many other options.

Software and Reviews

Some of the chapters contain some original software, written in BASIC, and written so you can use it on one of the popular, low-cost computers, such as the Commodore VIC and Commodore 64, the Atari-XL computers or the TI-99/4A.

In addition, I will have reviews of books, magazines, and software that will awaken your curiosity and spark your imagination. I want to startle you and surprise you. But I also want to inform you.

A Word of Explanation

This book was written over a period of two years. This explains the conflicting references to the ages of my daughter Catie and my son Eric. When I started writing the book Catie was six and Eric was two. Now Catie is eight and Eric is five.

I left the kids' original ages in the chapters since many of the chapters were written about Catie and Eric as they behaved at a specific age. This broadens the frame of reference for the book.

Chapter 2

Computers and Young Children

The Computer Playground

Eric ran over to the royal-blue sliding board, standing like a petrified monolith in the corner of his school playground. He looked up, searching for the top of the sliding board. It was lost somewhere in the drifting clouds. The metal rail glinted in the sun, resembling the arm of a giant crane. The dirty blue stairs seemed to go up and up, maybe ending somewhere in heaven.

Although Eric was only three, he confidently grabbed onto the sliding board's steel handrail and began climbing the steep metal stairs. After he got to the top, Eric stopped, panted like a puppy, and surveyed his playmates, screeching and dashing around the playground. His friends looked strangely small, since Eric was so high up.

Eric stretched on his tiptoes and grabbed the chinning bar above the sliding board. He swung back and forth wildly.

"Karla, watch!" he yelled to his teacher.

Halfway through each arc, he smacked his feet against the top of the sliding board. Loud percussion noises echoed across the playground.

Suddenly, Eric let go and went flying down the sliding board. He slid dizzily round and round the board's corkscrew spirals. He flew off the end of the board, sailed across a mud puddle, and demolished a sand castle being built by two of his friends. His friends went crying to their teacher. Eric howled. When he'd landed on the sandpile, he had scraped his bottom on the tip of an upturned shovel.

Moments later, the two kids were at work on another sand castle, undisturbed about the prospect of new children falling from the sky.

Eric was far away, on the opposite side of the playground. His cheeks were splashed with gritty tears, and his ears were filled with sand, but he was smiling. He had a new project. He was crab-crawling across the green-shingled roof of the school's playhouse, trying to work up his courage to jump into the old rowboat beached under the house's eaves. Eric thought maybe he could do it. After all, what's jumping off a roof to a three-year-old who can climb to heaven and drop from clouds and clobber castles?

The Computer Alchemists

The dark-suited men were frowning and fiercely intent. They spoke a cryptic, arcane language. "Disk error in sector five," said one. "Uh-huh," said another. "We'll have to bleed the accumulator." A third stared at a wide piece of paper he had just ripped off a green-and-white sheet of paper so long it could have covered a banquet table. "The OS slave program keeps stripping the control bits!" he said angrily. He crumpled up the paper and tossed it into a navy-green wastebasket under his spotless desk.

The men worked in a cool, almost Arctic room, drained of color by the overwhelming bath of white light emanating from the banks of fluorescent lamps fastened to the ceiling. Tall panels of lights winked like a city's streetlights viewed from a distant hill. Rows of metal boxes were filled with rotating platters and reels of gleaming tape that spun and whirred.

Staccato, machine-gun noises filled the room, as three of the boxes disgorged endless sheets of paper. The paper looked like a giant, jerking tongue or an impossibly long scarf pulled from a magician's sleeve.

The men were only vaguely aware of the tap-dancing printer and the clacking tape drives. These sounds were muffled by a loud whooshing noise that seemed to cover everything in the room like an invisible volcanic ash. The noise came from air pouring through the wall grates from huge cooling and filtering fans. After several hours in the room, the noise could numb a person's whole body and make it feel wrapped in a cocoon of cotton candy.

When Alien Worlds Collide

Eric on the sliding board and the men in the cotton-candy room seem to be worlds apart. Eric is a toddler having fun on the playground. The men are high-level programmers and analysts designing a new computer system.

Eric is outdoors in the sun. The men are in the Pentagon, in a guarded, windowless computer room.

Eric is covered with dirt and sand, and sweating from his exertions. The men are covered with plastic badges that proclaim their dedication and their utter seriousness about their craft. Dust and dirt are not permitted in their room. Only sweat — cold sweat.

I used to be one of those men. And Eric is my son. Now Eric's world and the world where I once worked have intersected, like alien worlds colliding.

When did the collision take place? It occurred when I first turned Eric loose on my home computer.

The shock was antic. It was also profound.

Eric is a person of arbitrary, random action. He is driven by fantasy, by enormous appetites and desires. His energy seems boundless. His temper is formidable. He vibrates to the tune of unchecked emotions and enthusiasms. He cringes from invisible monsters that scare him half silly, then walks unconcerned past real monsters he still hasn't learned to fear. To my most careful, clearly put instructions, he answers, "What?"

Computers, on the other hand, are creatures of reason and precision. They are powerful tools, the product of advanced technology. Computers are predictable. If we put in the right command, we get the right answer. But if we feed the computer garbage, we get garbage back.

Computers and three-year-olds seem worlds apart. Just imagine a computer that acted like a three-year-old. Or imagine a three-year-old that acted like a computer.

Three-year-olds are hardy and tough. Computers are finicky and delicate. Three-year-olds revel in dirt, mud, and grime. Computers like to be clean. They work better that way.

Knives, Forks, and Atilla the Hun

Eric has lots of nicknames: Little Hulk, Commando Kid, Tank, and the Lone Ranger, among others. My favorite, though, is Attila the Hun. This is an apt nickname because Eric, at heart, is a barbarian.

Can a barbarian ever learn to eat with a knife and fork? Can he be trusted in the same room as a computer?

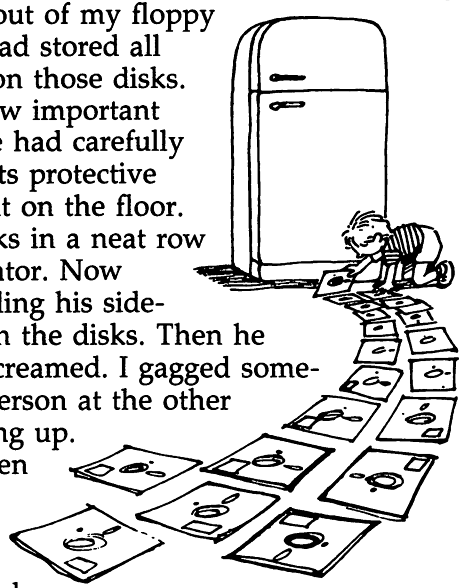
Sometime, over two years ago, I was on the phone in the kitchen talking with someone in California. I was trying to obtain some photos for a book I was writing.

Eric was home with me all day, back then, while his mommy went to work and his sister went to school. I didn't worry about Eric while I talked on the phone, since Eric was busy crayoning on a piece of scrap paper on the floor of my study. Or so I thought.

I was deeply engrossed in the conversation, trying hard to visualize the Californian's robot turtle, which I had never seen. Then I happened to look up, and my heart stopped.

Computers and Young Children

There, on the kitchen floor, Eric was hard at work, creating a flagstone sidewalk made out of my floppy disks. My *floppy disks.* I had stored all my books and programs on those disks. Eric must have known how important the disks were because he had carefully removed each disk from its protective case and gently dropped it on the floor. He had placed all the disks in a neat row that ended at the refrigerator. Now that he was finished building his sidewalk, he began to walk on the disks. Then he started to hop! "Eric!" I screamed. I gagged something inarticulate to the person at the other end of the phone and hung up. I charged across the kitchen floor. Eric fled to his bedroom.



I knelt beside the floppy disks. I was in shock. I was certain that the information they contained was squished, trampled, ruined. Several of the disks were covered with small dusty footprints.

Did I flog Eric? Was my information destroyed?

Happily, no.

But that was only the first of a brief, but emotional, series of incidents involving Eric and the floppy disks. The problem was he couldn't keep his hands off them. Something perverse inside him told him he had me by the throat. All he had to do was point to a disk, and I flew into a tizzy. Every time I left my study, he would sneak in and rob several disks and hide them under his bed. For a while, things got so bad I began sealing my desk drawer shut with strapping tape, just to thwart Eric.

Then I had an inspiration. I gave Eric a disk of his own. It was an old disk that was no longer reliable. I wrote Eric's name on the disk in big green, magic-marker letters. I wrapped the disk up and gave it to him as a present.

He was delighted. He still keeps the disk in a place of honor on his bookshelf.

And he stopped robbing my disks.

Now You See It, Now You Don't!

I have two kids. Eric, of course, and Catie, who is six.

I have several computers, and I have turned my study into a videogame arcade and a computer programming workshop for neighborhood kids.

The kids can use all the computers in the room except one, my writing computer. That computer is supposed to be off limits.

One day, while the kids were banging away at the other computers, I got up from my computer to stretch and make a snack in the kitchen. On my way back into the room, five minutes later, I happened to glance at my computer screen. Before I left, the screen had been filled with words — a section of a new book chapter I was writing. Now the screen was empty.

I panicked. I ran into the room yelling at the kids. "Who messed with my computer?" I hollered. "My chapter's gone. I've lost hours of work."

Then I checked the computer. I was sure it had been turned off, but it was still on. I did some more checking. Finally, I noticed the screen brightness switch. I turned it. Magically, my words reappeared. My chapter was untouched. Nothing was lost.

Just then, a very contrite five-year-old boy came up to me with his head drooping. "I did it, Mr. D'Ignazio," he said. "I didn't mean to kill your computer."

Joystick Tug of War

Another time, I was in the kitchen eating dinner with my family. Several children were still playing computer games in the study. All of a sudden, from the study came shouts and scuffling noises.

"It's my turn to blast them!" one child cried.

"You just blasted them!" cried another child. "Now it's my turn!"

I ran into the study and found a six-year-old and a nine-year-old (brother and sister) doing their best to rip two joysticks out of a computer.

"What are you two doing?" I shouted.

The kids turned toward me. Frightened, they dropped their joysticks. One broke open on the study's hardwood floor.

"Out! Out!" I yelled.

They dashed past me and ran out the front door of the house.

How Come This Doesn't Fit

Another time, I was in the kitchen preparing dinner for my family. It was close to five-thirty, and my wife, Janet, would be home soon from the office. As usual, the study was filled with kids playing with the computers.

I was slicing up some carrots for the salad, when I heard strange grunting noises coming from the study. Were the kids in there wrestling among the computers?

Hurriedly I dried my hands and went to see what was the matter.

When I got to the study door, I couldn't believe what I saw. A five-year-old boy was standing in front of one of the floppy disk drives. That was okay. I had taught all the kids — even Eric — how to insert disks properly into the drive.

But this kid wasn't inserting a disk. He was trying to shove a game cartridge into the drive. He was trying to squeeze an inch-thick plastic box into a hole designed for a disk only a sixteenth of an inch thick. He grunted loudly because he was trying so hard. He knew it was a tough thing to do, and he was giving it his best shot.

Strategies for Survival

Given these tales of abused and violated computers, what is the best strategy for mixing young people and computers?

You could put a lock on the computer room door until the kids all turn sixteen. Or until they leave home.

You could keep the computer in the car trunk and only bring it in the house after the kids are asleep.

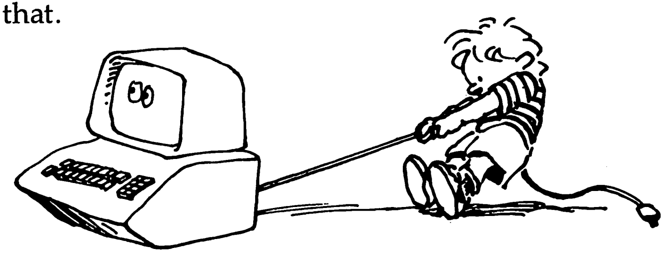
You could even fortify the computer, beef it up, and make it indestructible, the way the military does with computers on board tanks, guided missiles, and submarines.

You could do these things, but I would advise against it. In their place, I would advise a "Computer Orientation Session," followed by frequent "refresher" courses. I would also recommend a few simple precautions. And trust — lots of trust.

From my experience, I know that even toddlers can be taught to use computers properly.

Sure, there are lapses. (Oh, boy, are there lapses.) But for the most part, Eric and his little friends do an amazing job with the computers. They turn them on and off. They adjust the TV screen. They insert and remove disks and cartridges, even tapes. They know how to call up programs and operate the keyboard, joysticks, and other controls.

But they must be taught first. You need to sit them down and go over the etiquette of using a computer. Compare it to the family pet. You shouldn't pull its tail or try to ride on it. It wouldn't like that.



Instead, you need to be gentle with it and take care of it. If you do the right things, it will be nice to you and give you pleasure.

Frequent drills on specific rules help a lot, here.

It would help if you give the kids some responsibilities for the computer's care that are just their own. Like keeping the keyboard free of cookie crumbs. Keeping the screen wiped clean of fingerprints and all sorts of unmentionable objects. Or reporting any irregularities such as wires in the wrong place, or anything that looks dangerous.

Remind the kids frequently that computers run on electricity, and that electricity, like fire, can serve them or hurt them.

Also, if you can afford it, give them their own floppy disk, cartridge, or tape. Help them store their favorite programs and games. Give them a small case to store their printouts, game instructions, and other computer-related materials. This way, the computer will become partly theirs. As its owners, they will try harder to care for the computer and protect it.

The following precautions, like the advice above, are just common sense. For example, I learned to put rugs on my study floor. Then, when computer objects dropped, there was a smaller chance that they'd actually break.

Also, back up *all* your important programs and files. Then keep those backup copies in your office, your bedroom, or in a locked safe. I can tell you, it's a lot easier to keep your cool when your child puts your floppy disk in the toaster, if you have a backup to the disk.

Keep all plugs, wires, and cables (anything electric) hidden and out of the line of fire. My kids could trip on a paper clip. How about yours?

Put all your equipment on top of large, sturdy tables, surrounded by lots of sturdy chairs.

Keep your cartridges, disks, tapes, manuals, programs, etc., organized. Make everything accessible to your children. But make them respect and adhere to your organization.

Teach them standards. All devices must be shut off after use. Disks and tapes must not be left out after use. Everything has a place. (Nag them about this rule, if only to preserve your sanity.)

Two last elements of strategy. First, try hard not to get upset when it looks like your kids have done something to the computer. Most of the time things are a lot better than they seem at first. We've had dozens of accidents with computers here at the D'Ignazio Arcade, yet we've never broken a computer. At least not completely.

Second, don't push your kids to use the computers. Your three-year-old doesn't have to use educational programs to benefit from the computer. Your six-year-old doesn't have to learn to program.

Just the exposure to computers is doing your kids a world of good. Just switching buttons off and on. Or typing letters and shapes on the keyboard and watching them appear on the screen.

If you make kids' computer time synonymous with drudgery, with work, with tension, or with pressure, the kids won't like computers.

Don't push the kids. Just open the door, and sit back and watch. Computers fascinate kids, and that fascination will motivate them to learn on their own.

And congratulate yourself on your accomplishment. You are bringing together two very different worlds. And the results? The results are anybody's guess.

The New King of the Mountain

My daughter Catie was born in December 1975. My son Eric came along in April 1979, over three years later.

Today Eric is four, and Catie is eight. When I look at Catie, she always looks bigger than I remembered. When I look at Eric, he always looks smaller.

Catie amazes me because she is growing up so fast. Eric amazes me because he is staying little so long.

I don't know how many times I have wished that, somehow, Eric would catch up to Catie. It's not that I wish Eric would physically grow as big as Catie. It's just that I wish he would be as *able* as Catie at lots of different things — things like reading, writing, talking, listening, walking, running, minding his parents, drawing, painting. You name it.

This is a big secret that I'm telling you. I've never even shared it with my wife Janet.

But I'm not the only one in our family who feels this way. Eric feels this way, too. I can tell just by watching him struggling to keep up with his big sister. No matter what Catie does, Eric is there, too, trying to do it. But he is always a little slower than Catie, a little less able.

That doesn't stop Eric from trying. In fact, I think it makes him try even harder. And it has made him pick up the habit of jumping into any situation, no matter how difficult and complex, with the expression: "I know how to do it. Let *me* do it."

Unfortunately, in most cases Eric *doesn't* know how. But that doesn't stop him from trying.

And it doesn't stop me from admiring him.

Eric makes me think of other four-year-olds, especially four-year-olds with older siblings. They must be a pretty hardy bunch. They are at the bottom of the family totem pole no matter what is going on. They always come in last. Yet they never stop trying. I think that's pretty amazing. I know I couldn't do it. It takes a lot of spunk.

Computers and Young Children

The Great Equalizer

Last week we got a new computer product for our Commodore 64 — the KoalaPad from Koala Technologies. The KoalaPad comes with a black plastic stylus (a pencil without lead) and a software package, *KoalaPainter* from Audio Light, all for $125 (prices for the KoalaPad range from $110 to $150).

KoalaPainter is a do-it-yourself, create-your-own-pictures kit. It is also the great equalizer that has reversed Eric's position in the family. He used to be the least accomplished artist in the group (with the possible exception of our fat black cat, Mowie). But now he is the best artist in the family (the best *video* artist). He is the king of the mountain. And he's loving his new position.

The Acid Test

The acid test for any new computer product is whether it lets people do something on the computer that either (1) they could not do without the computer, or (2) they could not do as well without the computer.

The KoalaPad and *KoalaPainter* have dramatically passed this test. Eric can do things now on the computer that he could never duplicate on paper. In fact, Eric can do things on the computer that I can't duplicate on paper or the computer. And I'm 34.

For a four-year-old, Eric is a pretty good artist. But, using crayons, magic markers, and paper, he is no match for his big sister, for that matter, Janet or me. However, using *KoalaPainter* Eric is more than our match. Eric is now the reigning video-art champ of our family.

When Eric first boots up the *KoalaPainter* disk, he sees a bunch of menu boxes on the display screen. When he presses the point of the plastic stylus against the KoalaPad, he sees a cross-hairs drawing cursor on the picture screen. By moving the stylus around on the pad, he moves the cursor on the picture screen from box to box.

The boxes let Eric choose the size of his paintbrush, the color of paint, and the type of drawing he wants to do. Here are some of the boxes that hold Eric's drawing tools. With these tools Eric can:

***DRAW**	Draw freehand.
***LINE**	Create "rubber band" lines that stretch across the screen.
***LINES**	Connect rubber bands, end to end.
***OOPS**	Undo his most recent drawing command.
***FRAME**	Make rectangular frames.
***BOX**	Draw a frame filled in with a particular color.
***RAYS**	Draw lines that radiate from a central point.

***XCOLOR** Change one color on the screen to a new color.
***COPY** Copy a picture or portion of a picture onto a new screen location.
***MIRROR** Create mirrored images simultaneously on the picture screen.
***CIRCLE** Draw circles.
***DISC** Draw circles filled with a particular color.
***FILL** Fill in any shape he creates with any color he chooses.
***ZOOM** Magnify a picture for detailed drawing, erasing, or changing.
***SWAP** View two pictures at the same time. Using the COPY command, Eric can copy portions of one picture onto the other picture.
***STORAGE** Store his pictures on disk.
***ERASE** Erase the entire drawing area.

This sounds like an overly powerful array of tools for a four-year-old. Don't believe it. They boggle Janet and me, but they do not boggle Eric. He attacks *KoalaPainter* the same way he charges down the street on his Big Wheel bike — ZOOM!

Rough Drafts

Once, a couple of years ago, I had a conversation with Alan Kay, Atari's chief scientist for research and development. Kay is also one of the inventors of Smalltalk.

We were talking about the difference between a novice doing a task and an expert doing the same task. Kay said the key difference was that the novice was happy to do the task once, then go on to something else. The expert, on the other hand, did the task, then did it again and again to do it better. Each time the expert did the task it was like producing a rough draft of a writer's manuscript. The expert kept churning out new drafts. Along the way the expert polished, embellished, corrected, adjusted, and fine-tuned the product of his or her labors until it was done right. Then the expert went on to something else.

Why don't novices do rough drafts like experts?

They don't because usually it is too hard. They barely know their craft — of writing, painting, bridge building, cartwheel turning, driving, or whatever — and they usually work with primitive, amateurish tools. The expert, on the other hand, works with the finest tools technology can produce and is competent, disciplined, and experienced in the craft. The mechanical part of the work comes naturally, intuitively, and effortlessly. With the right tools and skill, the expert can reel off several drafts in the time it takes the novice to complete just one.

Computer Elevator Shoes

What happens, though, when you put a tool into a novice's hands that compensates for his lack of experience, his undeveloped motor and cognitive abilities, and his dearth of skill and craft?

KoalaPad's *KoalaPainter* is such a tool, and Eric is such a novice.

When I watch the beautiful pictures that Eric produces on the KoalaPad, it makes me think of a seesaw. Let's say I'm on one side and Eric is on the other. Usually, my greater size and weight makes the seesaw unbalanced. I drop like a stone, and Eric flies up in the air.

With *KoalaPainter* on his side, Eric goes down and I go up. What happened? It seems that *KoalaPainter* is a great equalizer. It amplifies and extends Eric's limited cognitive and motor skills beyond my own. It also taps his unbridled energy and imagination. It harnesses his tremendous curiosity and his tireless capacity for exploration and discovery.

With *KoalaPainter,* Eric does *lots* of rough drafts. And he does them fast!

The Brick Wall

Watching Eric use the KoalaPad and *KoalaPainter* can be sheer delight. It is not sheer delight at all times, however. Eric creates new *KoalaPainter* pictures at 90 miles an hour. Sometimes he takes a wrong turn and runs, SMACK!, into a brick wall.

Not surprisingly, Eric does not like to run into brick walls. So what does he do?

Sometimes when he runs into a wall he backs up and tries a new path. Sometimes he tries climbing over the wall. Sometimes he tries to knock the wall down.

And sometimes he just sits there and howls.

Leonardo the Little

When Eric boots up the *KoalaPainter* disk, he sees the menu with all the little pictures of brush sizes, paint colors, and activities. He never pauses to study this menu. Instead he presses the KoalaPad with his stylus, chooses a color, a paintbrush, an activity, and dives right into making a new picture.

I often wonder: Does he think that fast? Or is he in playground mode where he races from swings to monkey bars to merry-go-round randomly and at top speed?

In any case, within seconds after turning on the computer, Eric has a new picture under way. And more often than not, the picture is stupendous.

This is because, to create a picture, Eric has usually employed all the tools that *KoalaPainter* provides. All the tools. It may not be the most efficient way to make the picture, but it's Eric's way.

He uses the CIRCLE command to grow concentric circles of different colors around the screen.

He uses the MIRROR command and the RAYS command to create prickly sea urchins with purple and green spikes.

He uses the MIRROR command and the LINE command to create beautiful, layered tiles of multiple colors.

He uses the COPY command and the ZOOM command to create lots of tiny ERICs inside boxes, triangles, and circles, all over the screen.

He uses the ZOOM command and the BOX command to erase mistakes. The ZOOM command is good for erasing little mistakes. The BOX command is perfect for the great big mistakes.

Electronic Scribble

When Eric draws on a piece of paper on the kitchen table, he often just runs the pencil back and forth across the paper. To Eric this is great fun. To me it looks like scribble.

I think that what Eric is doing on the computer is scribbling, too. He is using the powerful tools made available to him by *KoalaPainter* and the KoalaPad to do advanced (superadvanced) scribbling. This is scribbling at a new plane — *hyper scribbling*.

Maybe the reason Eric is the family champ at video art is that he is also the family's best scribbler. The other family members just can't compete. As Catie grows older, she is losing her ability to scribble. And with Janet and me, it's a lost art.

But Eric is a master scribbler. And with the Koala tools in his hand, he is also a budding video artist. Maybe these tools will even arrest his development. Maybe he'll keep scribbling forever. And his scribbles will just keep getting more and more complex, and more and more beautiful.

Koala Technologies
3100 Patrick Henry Dr.
Santa Clara, CA 95050

Giants and Dwarfs

Yesterday my seven-year-old daughter Catie went off to computer camp at Hollins College, here in Roanoke, Virginia. When she came home she could hardly wait to tell me about her day. ''Computer camp was fun!'' she exclaimed. ''I thought we'd have to program all day, but we didn't. We did *lots* of things!''

One thing Catie did was send electronic mail back and forth to her campmate, Ashley Bell. The girls used the Minerva terminals on Hollins College's DEC (Digital Equipment Corporation) computer system.

Using a *big* computer was a new experience for Catie. The only computers she has ever seen are personal computers. Until yesterday she didn't even know what a big computer was. She thought it was really neat that she and her campmates were all using a computer—the same computer! She has grown up in a world of truly personal computing; *group* computing was a new experience for her. And she loved it.

For example, last night she wanted us to wire all our little computers together. Catie wanted to send games and electronic mail from her bedroom to her brother Eric's bedroom, and from the kids' bedrooms to my study, to the kitchen, the basement (kid's playroom), and even to the bathroom (''In case I want to tell you something while I'm taking a bath,'' Catie said).

I had to explain to Catie that the computers she was using at computer camp were really just terminals — ''octopus arms'' to a single giant computer. The computers in our house, I said, were different. They were like little dwarf computers — dwarfs of every sort, color, and personality. The dwarfs were all different from each other. Most were not even on speaking terms. Getting them to talk to each other was not ''just a quick project'' as Catie called it.

Different Perceptions

All this talk about giants and dwarfs got me thinking. When it comes to computers, Catie and Eric's generation sees things just the opposite of my generation (adults in their 20s and 30s, and above). When I close my eyes and try to picture computers, I still see the big IBMs, UNIVACs, Honeywells, Burroughs, and other machines.

Computers and Young Children

When Catie closes her eyes, she sees VICs, Commodore 64s, Apples, Ataris, and small lap computers such as the Epson HX-20 and TI's CC-40.

Now that small computers are popping up everywhere in millions of people's homes, classrooms, and offices, we adults are beginning to change our impressions. More and more, we acknowledge that there are two kinds of computers — giants *and* dwarfs.

But kids still see only one kind of computer. And that's all they'll ever see, until they grow up and try to get a job. Then, *BOOM!*, they'll come face to face with their first computer giant. And they'll be ill-prepared for that experience.

As far as most kids know, the Computer Age began in 1977 and 1978 with the appearance of the first Apples, Commodore PETs, and Radio Shack TRS-80s. The memories of younger kids don't go back even that far.

For most kids the giant computers have disappeared. Or they never existed. Giant computers are creatures from adult fairy tales that filter home from the office or workplace. Little kids see them occasionally on TV or in the movies because they are stereotypes. But they are vague, hazy creatures. Based on movies like *Superman III* and *WarGames*, kids' images of big computers are certain to be extremely stilted and unrealistic.

Giant computers are not a tangible part of a child's everyday world. But they have not disappeared. They have faded from the popular consciousness because they are not as trendy, fun, or cheap as the little personal computers. And they are not poking their terminals into all of our homes.

Giant computers are not mass-media superstars. But they still run the world. Personal computers are the front men for the computer revolution, but the big mainframe computers are still there, locked in the back room, chugging away, churning out most of the world's work.

An Emerging Network

Pretty soon, mainframe computers will play a larger part in the life of our families. When we dial up an information network such as CompuServe or The Source, we are linking our personal computer with a mainframe computer. Information networks, libraries, mail systems, banks, and shopping catalogs will soon become a major adjunct to personal computing.

In the next couple of years, small computers will cease to be little isolated atoms in the electronic universe. Instead they will become *nodes* linked in neighborhood, professional, and national networks. Personal computers are now autistic, shut off from each other. This is no way to gather information, learn new things, and get work done. Personal computers must communicate. And the big computers will act as the middlemen.

What's more, big computers have advanced computational abilities still sadly lacking in most smaller computers. As part of their computer education, our kids (even our smallest kids) should learn about these powerful machines.

Computer camps at universities are great places to learn more about large computers. Another place is Walt Disney's EPCOT Center (Experimental Prototype Community of Tomorrow), in Florida, near Disney World. When I found out that EPCOT did not feature personal computers in a "community of the future," I called to ask why. An EPCOT official explained that the designers of EPCOT had studied small computers, but they had decided that small computers were still not friendly enough. "Big computers can be much friendlier than small computers," she said. "Our big computers at EPCOT are fast, have gigantic memories to store millions of facts, and are capable of displaying beautiful, high-resolution pictures."

As I listened to her talk, I remembered the big computers I had grown up with. Maybe they were big and fast and were magicians with pictures, but they were also cold, unfriendly, and aloof. As a college student, the closest I ever came to an IBM 370 was when I pressed my nose up against the heavy-duty glass surrounding the computer room. I told this to the EPCOT woman.

She explained that their big computers were hidden away, but their terminals (like the octopus arms of the computer at Catie's computer camp) were all over EPCOT for everyone to use. "And we don't put up a barrier between people and our computers by making people use a keyboard," she said. "People can interact with our computers by operating simple controls, by touching the picture screens, and by talking to them. And the computers talk back."

Family Word Processing

I make my living by writing. I prefer to use *WordStar* on a personal computer to write faster and better. This has made me the default champion of "family word processing" here in our home in Roanoke.

I may be family champion for now, but there are several challengers on the horizon.

First, there is my wife Janet. Janet doesn't like *WordStar*. "It's too complicated," she says. "Using *WordStar*, you don't write a letter, you *program* it!"

Janet isn't an ex-computer programmer like me. Instead of running programs, she runs buses. And when she sits down to type a letter, she doesn't want to program. She does not want to go through an elaborate regimen of switch-flipping and button-pushing, as if she were Sally Ride preparing for blast-off on the space shuttle *Challenger*. She just wants to type a letter.

So Janet has abandoned *WordStar* in favor of another word processing program, *Text Wizard*. And now she's turning out whole piles of papers, résumés, and correspondence.

Then there's Catie. Catie does her word processing lying in bed.

Catie's computer is right beside her bed on a card table. Catie has a perfectly good chair to sit in, but she has swiveled her computer around to face the bed, and she types lying down, with her fat black cat draped across her back.

Catie has a simple explanation for the unusual way she has arranged her word processing workstation. "Mowie (the cat) wouldn't fit on the chair," she says.

Catie doesn't use *Text Wizard* or *WordStar*. Her favorite is *Bank Street Writer*. She cranks out page after page of very short stories, and notes to her parents, Eric, and Mowie. According to Catie, *Bank Street Writer* is the preferred word processor for seven-year-old girls, "because it's easy to fill up the little box on the screen, and it makes pretty letters."

Gobbledygook Processing

Is that it? Are there any other family members who are challenging me for my position as the family's number-one word processor?

There couldn't be any body else, really. There's Eric and Mowie, but neither of them can read or write.

Mowie is truly out of the running. She's content to lazily nap on Catie's back. But Eric is another story.

Every morning before nursery school and every night right at his bedtime, I find him in the study pounding away on a computer. I'm not certain what Eric is typing, but the paper flies! As far as I can tell, Eric may not even be competing with the other members of the family. He may have created his own category — *gobbledygook processing*.

Eric's workstation is even more unusual than Catie's. He achieves maximum productivity when he is standing on the creaky wooden chair in my study and leaning over the computer. He usually does his typing in his bare feet, and this morning, in his underwear, too. (The increased ventilation probably keeps the ideas flowing.)

I left him alone this morning for about 15 minutes. When I came back, he had a streamer of printer paper about six feet long sticking out of the computer. "Look at this, Daddy!" he exclaimed proudly. "Look at all the work I did!"

Eric doesn't have his own briefcase yet. So when he went off to school this morning, he had his work rolled up, fastened with a green rubber band, and dangling inside a plastic bag.

As soon as he got to his class, he was besieged by other kids asking to see the bag's mysterious contents. He pulled out the paper, unrolled it, and carefully explained how he'd made all the hundreds of random letters, numbers, punctuation symbols, and graphic characters that he and the computer had processed.

The kids were impressed. They'll probably go home and tell their families. And who knows, with Eric around, advanced gobbledygook processing might become very popular in Roanoke. Soon all the preschoolers could be doing it.

The Robot Teddy Bear

Just about this time last year my three-year-old son Eric and I went to the World Science Fiction Convention in Chicago. It was an experience neither of us will ever forget.

The hotel where the convention took place was filled with over 7000 science fiction movie-makers, writers, hucksters, and fans. Most of the fans were in costume.

Since the fans were in costume, Eric and I decided to wear costumes, too. We went with three friends we were staying with in Chicago. Hope (8 years old) dressed as a bride, Felicity (10) as a princess, Hugh (6) as an Indian, Eric was the Lone Ranger, and I went in the most bizarre costume of all — a business suit with a narrow tie, dark shoes, and a briefcase.

The kids' costumes fit right in, but my costume got a lot of surprised and baffled stares. Each time someone stared at me in wonder, I secretly patted myself on the back for my originality.

You Will Always Be in My Memory Bank

Eric and I returned to the convention on another day by ourselves. That's when Eric met Denby, a show robot from the International Robotics Corporation in Dearborn, Michigan. Denby was about six feet tall and all white except for a billboard advertisement on the front of his cylindrical body that advertised two of the leading science fiction magazines.

Denby was a real character. When he spotted Eric, he rolled over and greeted him. "What's your name, young man?" he asked.

Eric told him his name. He also told Denby about his mother, his sister, and his black cat, Mowie. He told Denby he had seen Darth Vader and Yoda at the convention, and that he had worn his Lone Ranger outfit last time he was there.

Denby told Eric that he was the nicest little boy he had seen at the entire convention.

Eric shook Denby's hand and gave him a big hug. Denby got so excited he started bouncing around the floor, spinning his head, and blinking his baby-blue eyes. "Whooweee!" he said.

Denby told Eric good-bye and rolled off across the convention floor. That didn't shake Eric. He followed Denby around the convention, up an elevator, and into a conference room. He didn't miss an opportunity to engage Denby in further conversation, shake his "gripper" hand, and give him kisses and hugs. (Eric couldn't reach more than a third of the way around Denby's barrel waist, so he hugged Denby's leg.)

Denby was a nice robot. Every time Eric appeared he acted really happy to see him. I think he must have realized that he had stolen Eric's heart.

Eric finally said good-bye to Denby, but not before he had collected a Polaroid photo of himself and Denby in front of the *OMNI* magazine booth, and another 8 x 10 color photo of Denby, complete with Denby's personalized autograph. On the photo Denby wrote: "To Eric, You will always be in my memory bank."

Now, a year later, the photos are still among Eric's prized possessions. One hangs on his bedroom wall; the other sits on his dresser and often gets taken to bed.

Eric Meets Little Denby

Big Denby made such an impression on Eric that when I saw a little toy robot at one of the booths at the convention, I immediately picked it up.

The new robot became known as Little Denby, then simply as Denby.

From the first night he got him, Eric began taking Denby to bed with him, like a mechanical teddy bear.

Denby does not look like a teddy bear. He is made of hard, black plastic. And his eyes flash when he is turned on. But to Eric, Little Denby is like a teddy bear. He is Eric's link to his friend Big Denby. And few people — biological or mechanical — ever made as deep an impression on Eric as Big Denby.

My Best Wishes to Everybody!

Learning to live intimately with a robot has not exactly been easy.

When Eric and I returned from Chicago, he continued taking Little Denby to bed with him every night. I remember nights when I would wake up and hear Eric crying in his bedroom. I would rush in and Eric would sob and tell me "Denby hit me" or "Denby stuck me." Eric had rolled over on Denby in his sleep. Denby is hard with lots of angles and bumps. He is not the kind of robot you can snuggle with and escape unbruised.

One night shortly after Eric and I returned from Chicago, my wife Janet and I were sound asleep in our room when I heard a loud *clunk!* come from Eric's bedroom.

Then came a loud, shrill air-raid siren.

Janet and I sat up in bed, alarmed and confused. We began climbing out of bed, and the siren stopped. Then, real loud, a buzzing, mechanical voice announced, "I am the atomic robot! My best wishes to *everybody*!"

It was Denby. He had fallen out of Eric's bed and landed on his head. On Denby's head is a yellow button. When you press the button (even when Denby is turned off), Denby makes an air-raid siren noise and tells every one who he is and wishes them his best. And Denby doesn't just say these things. He blasts them out like a bullhorn.

This is an okay feature for a robot to have during the daylight hours, but when a robot does this at two in the morning it can make you come unglued.

One further qualificaton: Robots should only be seen and not heard in the middle of the night or *in a car*. I don't know how many car trips we've taken where we have had to confiscate Denby from Eric and my daughter Catie. When Denby shouts out his greetings from the back seat, and, worse, when his alarm goes off unexpectedly, it takes all the self-control I've got to keep from swerving the car into a tree.

The One-Armed Robot

When children fall in love with dolls, blankets, and stuffed animals, they carry them around everywhere. Eventually the object of the child's affection begins to take on a very different look. It looks more and more faded, pummelled, mauled, unpleasant, and unhygienic.

My daughter, for example, had a mouse ("Mousie") that had been hugged and carried so much its skin grew so thin that its stuffing started leaking through. And she had a blue blanket ("NiNi") that, despite frequent washings, took on a greasy gray color and looked more like a shredded garbage bag than a child's blanket.

The same is true of robots. Except they're more sturdy. Eventually, however, all that love and affection begins to get to them. After a while they begin to look as ragged as a favorite stuffed doll or blanket.

Denby is a very tough little robot. For several months he continued to look as good as new. Then, one day, Eric decided to "walk" Denby down the basement stairs. Denby is only ten inches high, and he has no legs — just wheels. He made it to the top of the first stair, then he turned into a robotic pogo stick and bounced his way down the remaining stairs.

That was how Denby lost his arm.

The arm still sits, forlorn looking, in a small demitasse coffee cup in the kitchen. Janet has performed several surgical operations with Krazy Glue and Miracle Glue to try to reattach the arm to Denby. The arm stays on Denby for a short time, to Eric's acute joy and pleasure. Eventually, however, the arm ends up back in the coffee cup.

On that same trip down the basement stairs the little door on Denby's bottom burst open and spilled Denby's batteries all over the basement floor.

Today Denby wears a truss — three arcade-store game tokens underneath four layers of masking tape fastened to his bottom. The tokens and the tape keep Denby's batteries inside his body where they belong. But they don't always work, and this makes Denby sort of cranky and unpredictable. Sometimes he races around the kitchen floor, but sometimes he just sits on the floor and makes his air-raid siren in slow motion. It sounds a lot like a whine.

TOPO, the Bag Lady

We recently acquired a new member of our family — TOPO the robot from Androbot (101 East Daggett Drive, San Jose, CA 95134, 408/BOB-TOPO). Now we are a three-robot family (including the robot turtle who lives in the piano room).

I think TOPO looks fine just the way he came out of the packing crate from the factory — like a little white snowman. But my kids think differently. He must look naked to them, because ever since we first got him they have been dressing him up.

At different times TOPO has worn capes, shawls, cowboy guns, hats, flags, bracelets, and rainbow-colored Smurf belts. But my favorite is the time my kids dressed TOPO as a bag lady.

One night, very late, I was going around the house turning off lights and making sure all the doors were locked. I went into my daughter Catie's room. She was sleeping soundly. Then I went into Eric's room.

I got the shock of my life!

Looming over Eric's bed was a small figure dressed in a shawl, a scarf, and a faded purple skirt. It looked like a pygmy bag lady. The bag lady carried a bulging paper sack in each arm. Large, tacky, plastic bracelets dangled from her wrists.

TOPO the robot flanked by D'Ignazio family (from left): Catie, Fred, Janet, Eric.
Credit: Roanoke Times & World-News. Photo by Wayne Deel

And there was more. In the darkened bedroom she seemed somehow ominous and threatening. I think it must have been the white plastic Dracula teeth taped to her mouth.

I was relieved when I finally realized that the creature in my son's room was TOPO the robot. Then I grew amused. It was that "Look what I've gotten myself into" feeling that I often get when I hang around Catie and Eric. You see, when we got TOPO I didn't realize what we were doing. I thought we were acquiring a robot. But we weren't acquiring a robot, we were adopting a pygmy bag lady — a pygmy bag lady *vampire*.

Just what every family needs.

Now You Can Be Real to Everyone

When I was a kid one of my favorite stories was *The Velveteen Rabbit* by Margery Williams. The story is fairly well-known, but the subtitle is less familiar: *How Toys Become Real.*

Denby and TOPO remind me of the velveteen rabbit. When they first arrived they were just "things." But before long they became vital members of our family. Now we talk about them as if they have personalities, ideas, and feelings. We act as if they are *real.*

On ABC-TV's *World News* program last night, Peter Jennings, the show's anchor person, went to a teddy-bear convention. The title of the piece was "America Is Bullish on Bears." Hundreds of people had come to this convention with their favorite teddy bears. There were fat bears, beautiful bears, dumpy bears, big bears, and bears the size of pins and match sticks. There were wise bears, silly bears, watch bears, and guard bears.

The people who own teddy bears love them as much as we love our robots, maybe even more. To those people, the bears are alive. They are real.

How do robots and bears become members of your family? How do toys become real?

They become real when we project our ideas, thoughts, personalities, and feelings into them. It's the same thing novelists do when they create characters with words on paper. They create lifelike beings who inhabit the pages of their books.

And, almost as soon as we project lifelike traits into them, our toys become independent from their creators. They seem to have an identity all their own. They seem to exist whether or not we are around to project life into them. We never know what to expect from them. Their thoughts, feelings, and imaginary actions are always a surprise. We can't predict what they'll do next. All their actions are consistent with the personality that they have evolved, but they are not preprogrammed or "mechanical."

The reality of the teddy bear or robot is greatly heightened when its personality becomes a shared fantasy among several family members or friends. Then it becomes an ongoing "joint invention" of several people. When we hear other people talk about these creatures as if they were real, we come to accept their reality even more than before.

Robot Maids and Butlers

All of this brings me to the conclusion that the real reason we will buy robots by the thousands and millions is *not* so they can become our household servants. Instead we will buy them so they can become our pets, our companions, and our friends — just like a dog, a cat, a blanket, a teddy bear, or a velveteen rabbit.

According to most robotics experts, we are a couple of decades away from general-purpose household-servant robots. The sensors and computers in today's robots are too primitive for a robot maid or butler to survive in the hubbub and chaos of the average home.

Yet there are a dozen companies which are already marketing relatively low-cost consumer robots destined for the classroom or the home.

In *People* magazine and on TV talk shows, we see robot owners and their robots acting out our fantasies about what we'd do if we had our own personal robot. The robots are shown walking the dog, washing a window, or bringing the man of the house a beer while he reads the evening paper or watches a football game on TV.

This is silly!

How do you program a blind, wheeled robot who only accepts hexadecimal commands to walk a dog around the neighborhood?

How do you get a two-foot-high robot who can't pick up a dishcloth to go to the refrigerator, open the door, pick out a beer, and somehow find the TV room?

Even robot sentries and guard dogs are pure fantasy — a dangerous fantasy. I know how much trouble my parents and their friends have with the computerized security alarms they have purchased for their homes. They are constantly setting off the alarms and sending the police and the fire trucks to their homes by accident.

How would *you* like to face a guard-dog robot armed with mace, tear gas, or an electrified snout? Would you trust that robot to consistently distinguish you from a burglar or robber? Would you trust that robot alone with your children?

Why Buy a Robot?

Several months ago I wrote about a "computer friend" that parents could program for their children (see Chapter 11). I said that, ultimately, user-friendly computers would evolve into computer friends, and not just for our children.

Now I think that the biggest justification for buying a robot is that it can become a friend — to our children and to us.

And if it can't become a friend, at least it can become a pet.

Think about having a robot pet. It changes the whole way we look at robots. It changes all our expectations.

If we look at robots as servants, we expect them to be convenient, hardworking, labor-saving devices. This, robots emphatically are not.

But if we expect robots to act like pets — like the family cat, dog, guinea pig, or goldfish — then we have a whole new set of expectations. And we can define a whole new set of standards. These standards can be just as rigorous as the standards for a robot maid or butler. But they can also be a lot more realistic.

We don't expect our dog or cat to wash dishes or take out the trash. We sometimes expect them to guard and protect us, but their performance in this area (as anyone who has ever had a watchdog will attest) is notoriously spotty. Our fearless watchdog might lick a burglar's hand, then turn around and bite the newspaper boy on the seat of the pants.

What can we expect from robot pets?

First, we can expect them to be lovable. To be lovable they should be cuddly, fuzzy, and huggable. They should be small enough so we can pick them up and carry them around with us. They should be "lap robots."

Also, they shouldn't be perfect. They should be just as quirky and silly as our cat or our pet gerbil. On occasion, they should be naughty, they should pout, they should be perverse and impossible. Or they should at least give the right appearance. We can easily imagine the rest.

Second, they should be teachable. We should be able to imprint ourselves as much on them as they do on us. They should learn our names, our favorite interests, jokes, and whimsies. They should be nice to us. They should be like the big old dog who acts like he is excited to see us when we come through the front door, or like the cat who can't wait to hop in our lap the moment we sit down.

Third, they should be tough. They should wear more like Denby than like a teddy bear or a blanket. They should be survivors of a lot of rough-and-tumble affection.

Fourth, they should be portable. They should be able to go on car, train, and plane rides. They should be able to go on vacations to the beach and still work even though they have sand in their sensors.

Fifth, they should teach us. They can teach us formal things like arithmetic, the names of countries and presidents, and the spelling of polysyllabic words. But they should also teach us little intangible things, like loyalty, affection, trust, ethics, and values. They should learn our values then echo them to our children and our friends.

They Could Become Friends

We should remember that robots are, above all, creatures of our imagination. That is why we find them so fascinating. The more a robot encourages us to use our imagination when we deal with it, the more successful that robot will be. On the other hand, the more a robot tries to act like a mobile appliance, the more it will set us up for frustration and disappointment.

After all, what is a robot? I'm not sure I can answer my own question. But I do know that a robot is something more than an average machine like a dishwasher or vacuum cleaner. We project a great deal of ourselves into robots. We do not do this with vacuum cleaners and dishwashers.

What else do we project ourselves into? We project ourselves into pets, dolls, and toys. This is why we value these creatures more than our vacuum cleaners and dishwashers. It would be a shame if we were to build robots to imitate common household appliances. Then we would devalue our robots and they could never realize their potential. They could never be truly real.

If we want robots to become real, we should stop trying to get them to "grow up" and become common appliances. Instead we should direct them toward their greatest potential — to become mirrors of our minds, our feelings, and our imaginations. Today, using current technology, robots cannot become our household servants. But today's robots can become our toys and our pets. And perhaps someday they may become our friends.

Chapter 3

Children's Software

Software for Toddlers

I first started working with children and computers back in the early 1970s. I was a programmer for a large computer time-sharing company, and I took a briefcase computer terminal with me to elementary school classrooms around the District of Columbia. We dialed up the main computer on the telephone and plugged it into the terminal.

I wrote all the programs that I demonstrated to the kids. That's because there wasn't anything else out there.

Sure, there was CAI (Computer-Assisted Instruction) *courseware* available. But that was mostly for older kids, and it was very expensive. I operated my little computer-literacy project on a shoestring. CAI materials were over my students' heads and beyond the reach of my wallet.

Then came the flood of personal computers. But still no inexpensive software for children in preschool through early elementary school. Parents and teachers who wanted software had to write it themselves. Or they could find an occasional listing in a computer magazine.

Now, suddenly all this is changing. People have finally realized that even the smallest kids can use computers to have fun and to learn.

And computers are appearing in people's homes by the millions. By the *millions*.

Thousands upon thousands of the families who now have computers also have little kids. These kids represent an enormous market for software. Software companies and traditional publishing companies are leaping into this market by the dozens. All of a sudden we are being deluged by programs for little kids.

E.T. on Your Computer

Everybody is going computer. Everything that now appears in a book, in the comics, in the movies, or on TV will soon be loaded into a computer. Within the next few years, we will see all our kids' heroes and superheroes, myths, fairy tales, and favorite characters appear electronically on personal computers. Big Bird, Strawberry Shortcake, and Papa Smurf will all be computerized. So will Batman, Wonder Woman, and Cat in the Hat. So will R2-D2 and E.T.

Some of this new software will be junk: dull, of little educational value, using the big names (like E.T. or the Smurfs) only for the purpose of hooking the kids.

But there will also be a lot of good software. Its range and diversity will be breathtaking. And it will be fun *and* educational. Some of the new packages include *PLATO* software from Control Data Corporation; "Sesame Street" software from the Children's Computer Workshop (CCW is a spin-off from CTW, the Children's Television Workshop); "Dr. Seuss" software and games from Theodore Geiss and Coleco; electronic books from TI that read themselves (using TI's Magic Wand bar code reader); plus software from dozens of other major companies and institutions, including the Children's Capitol Museum in Washington, D.C., and Milton Bradley.

An Unparalleled Opportunity

The flood of programs for little kids is the cutting edge of the computer revolution. Programs for older kids and for adults will also have a powerful impact. But the impact on little kids will be the greatest.

Why? First, because they are little kids. Computers will be among the first things they see. Computer-assisted learning will be part of their earliest learning experience. It will affect *what* they learn and *how* they learn. It will shape kids' feelings about learning in general.

Second, up until now, most learning by little kids has been informal. Very few children today receive sustained, cumulative instruction before the age of five, when they are enrolled in kindergarten.

Soon all this is going to change. Four-year-olds, three-year-olds, two-year-olds, and kids even younger will sit down in front of their family computers and run exciting, fun programs that teach them things they otherwise wouldn't learn until they were twice as old. Or even older.

Third, much of this learning will be noninstitutional and extracurricular. Educational TV programs like *Sesame Street* made a stab at turning the home into a learning center. Now computers and the new "toddler" software will make this possible. Formal learning at home will skyrocket. And it will be largely self-sustained and unsupervised. Parents will encourage their kids to run the programs. But the kids will either do it or not. The amount of learning that takes place will depend mostly on the kids themselves and on the quality of the software they are exposed to.

When this class of computer-literate kids enters the public school system, watch out. Each kid will test out at a different grade level *on different subjects*. The strain on public schools will be enormous. Parents will pressure schools to continue the individualized instruction that the children began at home on their computers. The schools will have to respond. Whether they want to or not, the public schools, from kindergarten up, will be forced to computerize their curriculums extensively. Otherwise, the teachers will be overwhelmed by too many kids operating at too many levels.

What will be the outcome of all these changes in terms of children's values and the overall quality of their development? Millions of our youngest children will soon be exposed to computer software embodying all sorts of values. These values will affect the children's emotional disposition, their learning ability, and their social and spiritual development.

Little kids are especially vulnerable to new values. Their characters still have not fully formed. And yet what supervision are these kids likely to get when they sit down at their computers and run these programs? What control will parents, and even teachers, have on the shape and scope of their kids' development?

The Learning Center

What is the best way to teach little kids? Is it drill? Simulation? Invention? Discovery? Games? Or some combination?

The programs now appearing for children are based on one or more of the above learning philosophies. When you are selecting software for your kids, it's good to know which philosophy (or methodology) the software uses.

For each of the various philosophies, there are several good software packages. Drill is perhaps the oldest form of computer instruction. In recent years, drill programs have been maligned because they are said to be unimaginative, they don't take full advantage of the computer, and "they program kids, rather than the other way around."

But drill programs have a place, especially when they are fun and exciting, and when they teach new facts and concepts.

One drill-type package I recommend is *The Learning Center,* written by Bruce Mitchell. Bruce and his wife Diane run the Small

World kindergarten and preschool in Durham, North Carolina. Diane is one of Small World's teachers. Bruce and Diane also have two young sons. Bruce's programs are based on experiences with his sons, one of whom had a learning disability, and on several years experience with kids at Small World.

The programs are divided into three areas: Special Skills, Math and Number Skills, and Language Skills. The Special Skills section covers identification of colors, color names, and shape recognition and differentiation. The Math and Number Skills section covers counting, number recognition, addition and subtraction, and ones and tens. The Language Skills section includes programs for alphabet recognition, letter sequence, and symbol discrimination.

Children can interact with the programs using the computer keyboard or an inexpensive light pen sold by *The Learning Center*'s distributor, The Programmer's Institute. The programs are very friendly and easy to use. They are appropriate even for the youngest, nonreading children. My three-year-old, Eric, likes them a lot — especially the Count with Me program that lets him "count the monsters."

My only criticism is that the color program is sometimes not responsive to the light pen. I learned that this can be corrected by turning up my monitor's contrast control. The problem is present only in the Atari version and will be corrected with a new, more sensitive Atari light pen soon to be available from Programmer's Institute.

The Learning Center programs cost $74.95 for a cassette and $79.95 for a disk. I have the version that runs on the Atari 400/800. I understand they also run on the VIC, the Commodore 64, the TRS-80 Model I, Model III, and Color Computer, the Apple, and the TI-99/4A.

The Edumate light pen costs $34.95. To find out more about the light pen and *The Learning Center* package, contact:

The Programmer's Institute
P.O. Box 3470
Chapel Hill, NC 27514
(919) 967-0861

KinderComp

Two other excellent software packages are *KinderComp* and *Rhymes & Riddles*, distributed by Spinnaker Software Corporation of Cambridge, Massachusetts. Both packages employ several teaching philosophies. They are so attractive and fun to use that they have captivated my entire family, including three-year-old Eric, seven-year-old Catie, and their parents.

Each package is $29.95. They are available for the Atari computers, the Apple II+ (48K, DOS 3.3) and IIe, and the IBM PC. Contact:

Spinnaker Software Corporation
215 First Street
Cambridge, MA 02142
(617) 868-4700

KinderComp was written by Doug Davis for his daughter Amy. The name makes it sound like a collection of arithmetic programs, but it is really six programs that teach a diverse group of numerical and alphabet-oriented skills.

One of the programs is called Draw. It can be used by even the youngest children (say, kids under two). To work the program, the child twists a joystick and creates multicolored, musical pictures on the display screen.

My three-year-old had no problem using Draw to create all sorts of shapes. When I asked him to tell me what he was drawing, I was boggled. "Up here, Daddy," he said, "is an upside-down two. Over here is a house. These are steps. This is the roof. This here is the room where the doggie lives. This is a hotel. That's a big swimming pool. Over here is the fire escape. This green stuff is Hulk Grass. It's bigger than the hotel."

Draw is a super program because of its visual and auditory feedback, because it's so easy to use, and because it stimulates a child's manual dexterity, creativity, and artistic skills.

The other *KinderComp* programs are more focused and less open-ended. But they are original and exciting. Scribble amplifies and animates a child's random scribbles. Names turns a child's name into a fascinating sound and light show. (Boy, was I jealous when Eric turned his name into a hilarious musical cartoon. I never got that kind of reinforcement with my name, Fred.)

Sequence helps kids learn number sequence; Letters teaches them lowercase letters and the location of letters on the keyboard; and Match is a great pattern-matching game.

Both *The Learning Center* programs from Programmer's Institute and *KinderComp* from Spinnaker are valuable for the *specific* skills they teach young children. But they are equally valuable as "doorways" for children to enter the world of computers. Even the youngest children can use the computer for fun, purposeful activities that *they* control. They learn the computer keyboard. They learn how to manipulate and respond to material on the display screen. They learn how to operate the computer and run programs.

Computer skills still baffle and intimidate a large number of adults. People once believed that mastery of these skills required a college education. Yet *The Learning Center* and *KinderComp* teach these skills to little kids who are still running around in diapers.

An important aspect of toddler software is the way it reinforces children's responses — that is, the way it responds to kids' right and wrong answers. Both *The Learning Center (LC)* and *KinderComp (KC)* score high in this category. For right answers, *LC* gives a happy face and a happy tune; *KC* gives a happy face with a wink. For wrong answers, *LC* gives a sad face and a toot; *KC* gives a sad face crying a big tear.

I like both packages' responses to wrong answers because they are quickly over and do not intimidate a child. I like *KC*'s response very much because the computer doesn't show disapproval or anger when the child errs. Instead it becomes *sad*.

KC is good also because it gives the child hints when he is wrong, and eventually gives him the right answer. But after the child gets an answer wrong, he is not rewarded for later getting it right. This confused my son Eric. When he didn't get a happy face on the screen for an answer at which he had worked especially hard, he wilted a little bit.

On the other hand, *KC* is especially good because it lets the child follow his progress with a string of pluses (+) on the screen (a plus for each correct answer). And the child gets a special reward for answering a series of questions correctly. This feature made a big hit with Eric.

Last, I also recommend *Rhymes & Riddles (R&R)*, another package from Spinnaker. *R&R* was written by a husband and wife team. The format is "updated, nonviolent Hangman." On the screen appear a bunch of dashes. The dashes represent missing letters. The child tries to guess the letters. By guessing all the letters, a child builds either (1) a nursery rhyme, (2) the answer to a riddle (Sample riddle: Why can't bikes stand up? Answer: Because they are *two* tired.), or (3) a famous saying.

If a child doesn't guess the right letters after a certain number of tries, she doesn't see some poor little man or woman get hanged. Instead, she builds a sad face, and the program displays the correct letters.

All three games (*The Learning Center, KinderComp,* and *Rhymes & Riddles*) help kids learn the computer keyboard, the letters of the alphabet, and the spelling of different words. The kids' learning is reinforced with color graphics pictures and musical segments taken from nursery rhymes and the children's songs.

Little Kids' Computer Magazine

Software for kids isn't the only thing that's happening. There are also a growing number of kids' computer magazines. One that I recommend is:

> *Turtle News* and *TL Computing* (Kids $9/year; Adults $25/year) Young Peoples' Logo Association, 1208 Hillsdale Drive, Richardson, TX 75081. Call 214-783-7548. Focus on Logo, PILOT, and turtle graphics programming, but also features articles and programs in BASIC. Education, entertainment, and material to help kids with special needs. For kids seven and up.

Sesame Street and Interactive TV

It was like Super TV. I was sitting in a folding chair in the Grand Ballroom at the Hyatt Regency Hotel in Tampa, Florida. In front of me was a giant TV screen. Behind me was an audience numbering in the hundreds. Nearby were all sorts of mysterious high-technology devices. Writhing across the floor, like rainbow-colored pythons from a tropical rain forest, were dozens of cables.

The room darkened. The screen grew bright.

A big, blue, scruffy-looking creature appeared on the screen. It was Cookie Monster. He was wearing a chef's hat and munching a chocolate chip cookie. Crumbs flew in all directions.

It wasn't TV after all. It was a new computer game from the Children's Computer Workshop (CCW). CCW is a new division of Children's Television Workshop (CTW), the producers of *Sesame Street, Electric Company, 3-2-1 Contact,* and other children's educational programs and materials.

CCW's first four electronic learning disk packages are:

Ernie's Quiz (For children 4 to 7)*
Instant Zoo (Ages 7 to 10)*
Spotlight (Ages 9 to 13)*
Mix and Match (For the whole family)

Each package contains four programs that run on a 48K Apple. The starred packages (*) require Integer BASIC. The unstarred package (*Mix and Match*) requires Applesoft BASIC. *Ernie's Quiz* and *Spotlight* require paddles. All packages are more effective if you have a color TV. Each package costs $49.95. For more information, contact your local Apple dealer, or write:

Apple Computer Company
20525 Mariani Avenue
Cupertino, CA 95014
(408) 996-1010

Cookie Monster Munch

Barbara Stewart, a project manager from CCW, had brought Cookie Monster to the Hyatt Regency Hotel in Tampa. The occasion was the third annual Florida Instructional Computing Conference, one of the largest regional educational computing conferences in the country, held March 28–30, 1983.

Barbara was the conference's keynote speaker. In her speech, she announced that CCW was producing a new line of educational programs for the Radio Shack Color Computer (16K) and for the Atari 2600 VCS computer and game system. CCW plans to develop each cluster of programs on a particular machine and have the computer manufacturer distribute them through its standard outlets. Eventually, at least one set of CCW packages will be available for many of the best-selling computers.

Cookie Monster Munch is typical of the new Atari games. The game is a numerical maze game for kids ages three to seven. It comes with a colorful booklet explaining how the game works. The Table of Contents and other sections are all hand printed, as if by Cookie himself. I like the "Note to Parents" at the beginning of the booklet. Also, a symbol of a parent with his or her arm around the shoulders of a child is used throughout the booklet. The symbols are accompanied by suggestions to increase and enrich parent-child interactions with the computer and *with each other*.

And how do the kids and their parents interact with the computer? They use the new Atari Kid's Controller. CCW worked with Atari to develop the Controller. It's a large numerical keypad with big buttons and is very sturdy. It plugs into the left controller jack at the back of the Atari 2600 VCS and is an easy-to-use keyboard or joystick for game play. Each CCW package contains a colorful plastic overlay that fits atop the Kid's Controller.

Cookie Monster Munch is a maze game, so the child has to make characters in the game move up and down, left and right, through the maze. Accordingly, the overlay has a big picture of Cookie Monster and designates buttons (hidden underneath the overlay) as movement buttons with big arrows for all four directions. It's so easy to use that even toddlers with small hands and adults with keyboard phobia will be able to play.

Another nice feature of the games is the Read Aloud Story in the beginning of each booklet. With personal computer graphics (especially VCS graphics) still at a relatively primitive level, the images of the *Sesame Street* characters, like Cookie Monster, are nowhere near as nice looking as they are on TV. But the story helps

Cookie Monster's Munch for the Atari.

remedy this problem. It engages the child's and the parent's imagination, and it gives the simple-looking game on the TV display meaning and depth.

Cookie Monster discovers a chocolate chip cookie garden. He begins running around the garden picking up cookies. He takes the cookies and puts them in his cookie jar. Cookie's intentions are sensible, but he can't resist eating the cookies before he makes it to the jar. A little kid appears — the Cookie Kid. Cookie Kid tries to collect the cookies and put them in the jar before Cookie Monster can eat them.

The paths in the cookie garden are like a maze. There are ten different game levels and mazes. The easier games are one-person games. The harder games are one- and two-person games.

Like the *Sesame Street* TV program, the games are designed as entertaining ways to teach kids prereading skills. The kids get to move Cookie Monster or the Cookie Kid through the mazelike cookie garden. Tracing the maze pattern while remaining within its borders helps kids practice the hand-eye coordination they'll need for beginning reading and writing. Also, they learn to follow directional arrows and become familiar with the relational concepts of up, down, left, and right.

Peanut Butter Panic

Here are some other new CCW games:

- *Ernie's Magic Shapes.* This is a home game for kids ages three to six that runs on the TRS-80 (16K) Color Computer. Kids help Ernie zap geometric shapes and use them to build colorful figures. The games help kids develop classification skills including matching shapes, recognizing embedded figures, structuring parts of an object into a meaningful whole, and discriminating between similar and different shapes.

- *Grover's Number Rover.* This is a home game for kids ages three to six that runs on the (16K) Color Computer. Grover floats across the top of the screen in his Number Rover. The child helps Grover find the answer to his arithmetic problem. When the child discovers the number that solves Grover's problem, Grover picks up that number of Twiddlebugs. This is a humorous part of the game. The Twiddlebugs are upside down.

- *Taxi.* This is a home game for kids ages seven and up that runs on the 16K Color Computer. This is a junior adventure game. Kids get to operate a two-cab company in any one of six cities around the world. They pick up passengers, deliver them to their destinations, and earn fares and tips. The game encourages visual problem-solving in a cooperative environment.

- *Peanut Butter Panic.* This is another funny game. It is a home game for kids seven and up that runs on the 16K Color Computer. Two little nutniks try to catch stars that zip by above them in the sky. Kids control the nutniks and launch them from a platform that resembles a giant seesaw. The nutniks can jump up and down on their own, or two kids can launch them from the seesaw.

  When the nutniks jump up and down they lose weight and get real skinny. When they get skinny, they can't jump as high. To get fat again they have to eat peanut butter sandwiches. They build a peanut butter sandwich by catching stars. They have to watch out for mean snarfs who swoop down out of the sky and steal their sandwiches.

  The primary objective of this delightful game is teamwork and cooperation.

- *Picture Place.* This *school* game is for kids ages five and up. (I think that it is a good game for preschoolers, too.)

  Kids get to choose a picture from a library of six background scenes, including a city and a countryside. At the bottom of the screen are word boxes with words inside like *dragon, car, bicycle, family,* and *castle*. Kids choose a word by moving a joystick and positioning a big "cursor box" so that it overlaps with one of the word boxes. They pick up the word box and move it up the screen and position it on the background scene. Then, when they press the RETURN button, the word transforms into a picture. For example, the word *dragon* becomes a picture of a dragon, set in the world pictured in the background scene.

CCW's Values and Goals

Barbara Stewart thinks that personal computers will evolve into "interactive TV." She wants to create programs for TV that will accomplish the same goals as the *Sesame Street* programs on regular TV. The programs will focus primarily on developing math and reading readiness skills. But they will also stress certain fundamental *Sesame Street* values, including teamwork, cooperation, and nonsexist, nonviolent, pro-social play.

The programs are to be appropriate to their target age group and appealing to both girls and boys. They should meet educational goals for children of each age group and development level. They should be easy to understand, easy to play, and nonjudgmental. They should not frustrate children. Instead, they should encourage a child to grow and improve his or her self-image.

If these games prove to be as thoughtfully and as creatively executed as *Sesame Street* itself, children (and parents) everywhere can look forward to exceptionally rewarding educational experiences via interactive television.

Big Bird, Blue Jeans, and Blackboards

The week after I returned from Tampa, I hopped aboard another plane and flew up to New York to visit the people at the Children's Television Workshop and the Children's Computer Workshop. CCW and CTW were a treat. It was good to meet relaxed, smiling people dressed in blue jeans and T-shirts. And big fuzzy Cookie Monster, Kermit, and Big Bird dolls were perched on file cabinets and smiled down from colorful posters on the walls.

During this phase of my travels, I got to see a lot of educational software. My chief impressions were that the software is quickly improving and that its creators are beginning to deal with learning in a totally new manner.

Only a year ago, educational software on personal computers consisted almost entirely of old-fashioned "electronic textbook" programs and drill and practice programs.

Six months later we were besieged by educational game software, really *disguised drill and practice.*

Now we are beginning to see something new. We are seeing the first real microcomputer simulations, where the kid's computer "pretends" it is a world or environment and challenges the child to playact and build a face, conduct a chemistry experiment, pilot a starship, run a nuclear reactor, solve a crime, or map out a new world. Some of the forerunners in such simulation games include The Learning Company's *Gertrude's Puzzles;* Spinnaker's *Facemaker, Snooper Troops,* and *In Search of the Most Amazing Thing;* and Children's Computer Workshop's *Electronic Blackboard* game.

Electronic Blackboard suggests an even newer type of educational software for children: kids' workstations — where the computer becomes a general-purpose tool to enable children to use the computer to do *whatever they want* (just like adults).

Electronic Blackboard creates an electronic mailbox for kids. Several blackboards are pictured on the computer's display screen. At first they are empty. Kids get to "borrow" a blackboard, associate their name with it (as a mail address), and use electronic *chalk* to write messages on the board for other kids to see.

If a message isn't private, you get to see it just by calling up a particular blackboard. If, however, it *is* private, the child can hide it. You can access private messages "for your eyes only" by typing your name. It's not a foolproof security system, but it makes a great educational activity. Kids get to practice their reading and writing skills. And they are learning how to do word processing and send electronic mail.

Follow the Leader

Recently I had a conversation with Dr. John Victor, the president of Program Design, Inc., in Greenwich, Connecticut. Victor is a gifted educator who creates voice-enhanced storytelling programs for preschoolers — programs such as *Sammy the Sea Serpent, Teddy's Balloon, Robin's Halloween,* and *The Adventures of Oswald.*

When I spoke with him, Victor had recently completed a study of the way young children learn. Victor found that children have problems in their first experience with formal learning because they have trouble following directions.

According to Victor, when a teacher or parent gives oral or written directions, children quickly get bogged down in the directions' abstractions. The directions are usually given in a sequence of instructions that involve a mastery of directionality and discrimination. Children get confused when they hear strings of words like left, right, top of the page, center, lower right-hand corner, before, after, above, and below.

If the sequence of directions is too long, a child will get swamped and not remember more than the first couple of instructions.

If the instructions are written down, a child has different problems. For example, he or she has to discriminate between different problem letters (like *p* and *q* and *g, b* and *d, w* and *m,* and *z* and *n*). They have to follow the instructions across the page, and they must obey the instructions in order. This is a lot to ask of a child who is just beginning to read and who may suffer from early-learning dyslexia.

Also, when children go to obey instructions, they are hampered by their terrible motor skills. According to Victor, "Adults have no idea how hard it is for young children to make their fingers and hands do what they want. The best way to appreciate what kids go through is to use your nose to try to open a door, or try to drive a car with your elbows."

Caught in the Directions Swamp

The problem with most materials which young children use is that they lump all these problems into one exercise. A child's difficulties with long sequences of abstract directions are compounded with problems of visual discrimination, eye-tracking limitations, and the lack of fine motor skills. As a result, says Victor, material that is designed for four-year-olds is really more appropriate for eight-year-olds. In his study, Victor determined that even a select group of very bright children were understanding only about 70 percent of the instructions given to them by a trained teacher.

What is worse, the teacher was not aware the kids were having problems. And the kids didn't know either. They weren't aware when they made a wrong turn. The wrong turn came when they missed an instruction, when they got the order of instructions mixed up, or when they misinterpreted an instruction.

Forget It! I'll Do It Myself!

Victor's comments reminded me of my own four-year-old son Eric. Eric is a pretty normal kid of at least normal intelligence. Yet I can't count the number of times I have rattled off a bunch of instructions to Eric and had him look up at me and say, "What?"

This drives me crazy.

The first time Eric says "What?" I usually respond by carefully repeating the instructions.

Eric usually responds to this by repeating, "What?"

Then I usually respond by telling him, "Forget it! I'll do it myself!"

After talking with Victor, I realize that Eric is not being dense or unattentive — or worse, a smart aleck. I realize that he probably does not have his ears clogged with ear wax or lint.

The problem is he just doesn't understand my directions.

Learning How to Juggle

Young children need materials which help them learn how to juggle lots of directions at a time and then follow them. They need material which lets them take the subsidiary direction-following skills (including visual discrimination, eye-tracking, understanding abstract instructions, directionality, memory, sequencing, and motor skills) *one at a time*.

Below we'll look at three outstanding programs for the PCjr which can help your child learn how to follow directions. The programs are:

Juggles' Butterfly	For kids ages 3 to 6
Bumble Games	For kids ages 4 to 10
Bumble Plot	For kids ages 8 to 13

The programs were produced by The Learning Company of Menlo Park, California. The Learning Company, led by

Ann Piestrup, is one of the premier educational software companies in the world.

Building a Rainbow

Juggles' Butterfly is for children ages three to six, but I have seen two-year-olds and seven-year-olds using it happily and successfully.

Two jugglers ("juggles") appear as soon as you put the disk into the computer and start the program running. Then a menu (list) of activities appears. The activities appear as four large white numbers and four multicolored figures: a rainbow, a butterfly, a windmill, and a question mark.

The first activity, the rainbow, is the simplest. It helps a child follow directions for "Above" and "Below."

The *Juggles' Butterfly* package with the disk and manual also comes with a keyboard overlay and two cut-out blue strips. The first long strip goes across the top of the keys on the second row up from the bottom of the keyboard. Your child will probably need help attaching it.

The strip divides the keyboard into two halves: a half *above* the blue line and a half *below* the blue line.

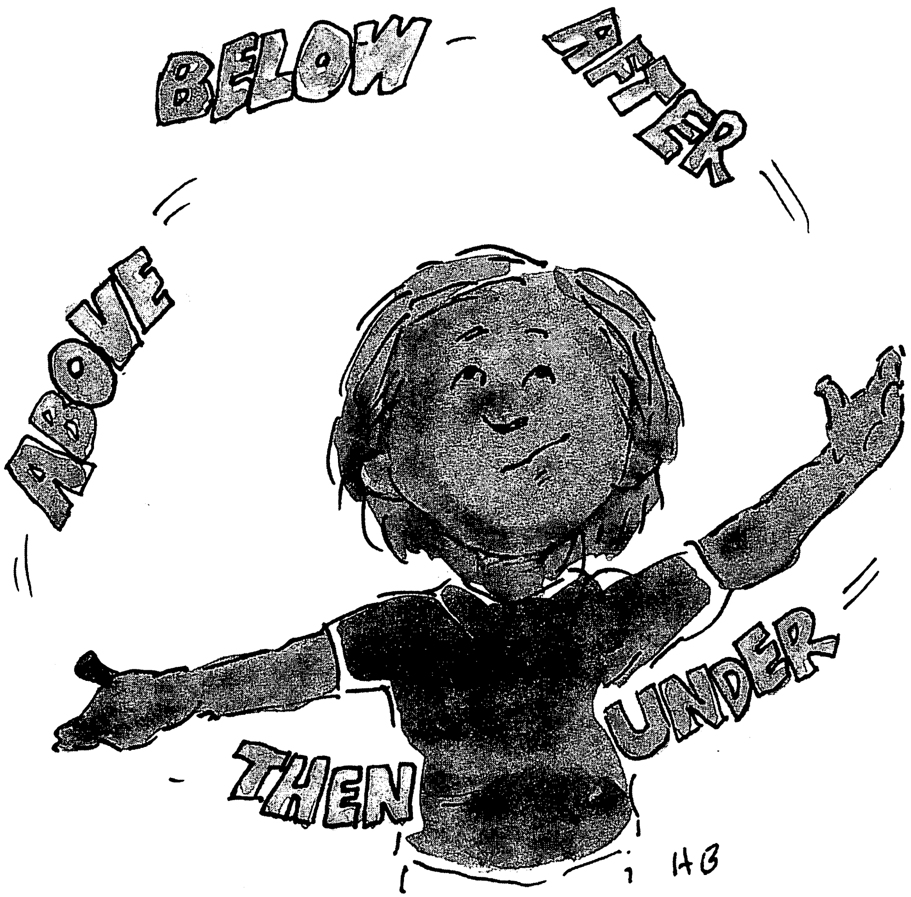
A picture of the blue strip appears on the computer's screen as a large, blue, horizontal line, stretching from left to right across the screen.

Your child begins the exercise by pressing any keys he or she wants. If the key is above the blue line, the word ABOVE appears, and a colorful stick appears above the blue line on the screen. If the key is below the blue line, the word BELOW appears, and a stick appears below the blue line.

After your child has pressed five keys and five sticks have appeared, he or she is asked to press any key ABOVE or BELOW the blue line. At the same time, a dotted stick appears on the screen, either above or below the line.

If the child is right, he or she builds solid-colored sticks below and above the line. If the child is repeatedly wrong, the computer goes back to the earlier activity and lets the child practice pressing keys all around the keyboard. Each time the child presses a button, the computer reinforces the choice by printing a stick above or below the blue line and by printing the word ABOVE or BELOW.

At the end of the activity, the child is rewarded by being able to play with an animated picture of the rainbow. The child can make the rainbow "rain" and change colors. He or she gets to match the colors of the rain with the colors in the rainbow.



Windmills and Butterflies

The other two activities in the *Juggles' Butterfly* package work on the same principles as Juggles' Rainbow.

Activity two, Juggles' Butterfly, teaches children how to distinguish between left and right by building a butterfly (with symmetrical, left, and right multicolored wings). The child prepares for this activity by dividing the keyboard up into right and left segments by placing the short, blue strip down the center of the keyboard.

Activity three, Juggles' Windmill, teaches children how to distinguish between right, left, above, and below, by building a windmill with four blades. The child prepares for this activity by placing both strips on the keyboard and dividing it into four segments that coincide with the four windmill blades on the computer's screen.

The last activity in the *Juggles' Butterfly* package is a "help" activity. If the child presses the 4 button on the keyboard (the number four on the menu is next to a picture of a big rainbow-colored question mark), then the child can get instructions, turn off the picture clues, turn off the music, return to the picture menu, or quit playing.

Letter Stumpers

One of the best exercises in *Butterfly* is where the computer teaches a child to recognize the difference between the four letters *d, b, p,* and *q*.

The exercise starts with a circle on the screen. When the child presses a key in one of the four quadrants on the keyboard, a vertical stick appears, sticking to the circle. If the child presses a key in the upper-right part of the keyboard, a stick appears on that side of the circle. The two figures, together, become a big letter *d*. Similarly, by pressing:

A key in the lower-right part of the keyboard, the child gets a *q*.
A key in the lower-left part of the keyboard, the child gets a *p*.
A key in the upper-left part of the keyboard, the child gets a *b*.

These four lowercase letters — *d, q, p,* and *b* — are the four most difficult letters for children to distinguish correctly. This game helps kids learn to recognize these letters at the same time they learn how they are different. And the entire exercise occurs in the context of the "Above, Below, Left, Right" activities and with the goal of developing the child's direction-following skills.

Hidden Learning Power

Juggles' Butterfly helps your child learn some basic direction-following skills. It lets children master the skills at their own pace. The simplicity of the games makes them enticing and accessible to even the youngest child. Yet they still have many vital exercises hidden inside which encourage a child to pick up several skills, including:

- Relating horizontal to vertical planes (keyboard to screen, like desk to blackboard).
- Eye tracking (from left to right, and from top to bottom).
- Noticing shapes and their positions; watching patterns develop.
- Recognizing the shapes *b, d, p,* and *q*, the four hardest letters in the alphabet.
- Learning spatial awareness (halves, quadrants).
- Matching colors.
- Learning opposites (above/below, left/right).

- Using words to give directions — matching directions with activities on different sections of the keyboard and screen.
- Noticing symmetry.

Children can practice all these skills in a self-paced, enjoyable game. Whenever they have trouble acquiring any of the skills, they receive immediate, constructive feedback, and then they go back for additional practice.

Follow the Bumble

The two other Learning Company packages are *Bumble Games,* for children ages four to ten, and *Bumble Plot,* for children ages eight to thirteen.

The "Bumble" packages have 11 activities on two disks (five on *Bumble Plot* and six on *Bumble Games*).

The Bumble activities pick up where *Juggles' Butterfly* leaves off. They teach a child more complex direction-following skills. After a child has mastered all the Bumble activities, following directions on a simple sheet of paper (for a homework assignment or a test) should be a snap.

Number Lines and Arrays

As you will see on the cartoon pictures on the manuals, Bumble is a mischievous, playful bee. Bumble likes to play games. Her games turn scary, technical things such as negative numbers, number lines, and arrays into playgrounds for you and your child.

For example, in the first game in both *Bumble Games* and *Bumble Plot,* Bumble is thinking of a number but won't tell you what it is. It is up to you to "trap" the number. Bumble will give you clues about the number, but you must guess it yourself.

The game is played on a number line (a horizontal line with large blue numbers written beneath it). In one game you try to trap a number by telling Bumble a range of numbers on the number line. Your guess is represented visually on the screen by a purple box around the numbers on the number line. If you are right, the box shrinks and you get to guess again. You keep shrinking the box (the range of unguessed numbers) until you've caught the number.

In the other game, Bumble tells you if the number is less than or greater than the number you guessed. Bumble's hint is represented visually with left and right arrows over the number line.

Array Playgrounds

In the other nine Bumble activities, you play on an array — a two-dimensional grid, represented either by stacks of big, colored boxes or crisscrossing lines.

When I was a child, we did lots of activities on graph paper arrays. But the work was tedious because I had to plot all my points, then draw lines between the points (dot-to-dot) with a sharp pencil and a ruler.

I plotted these points over several hours with a great deal of grunting, puffing, complaining, and erasing. I plotted the points mechanically. I never thought about what I was doing. I hated all the painstaking drawing. And I hated the teacher for demanding that the plotted pictures be both accurate and clean.

Now, with these Bumble activities, you and your child can type in array coordinates (the row number and the column number or letter — all in big, bright, colorful characters), and your dots and lines appear magically on the screen an instant later. You get to concentrate on thinking your way through the Bumble game, and the computer takes care of all the messy drawing and plotting. If you are right, you get results in a flash, and they are picture perfect every time.

Catching Bumble

In the early games (in both Bumble packages), the arrays are small, either four columns and four rows, or five columns and five rows.

When the games start out, Bumble is missing. A giant picture of Bumble appears on the screen, just to remind you what you are looking for. You learn that Bumble is a big, cute bumblebee who is hiding somewhere under a block in the array, or flitting about on the array grid, just daring you to try to trap her.

You try to find or trap Bumble by giving the row and column where you think she is — or where you think she may fly next.

If you are wrong, you will know it because she will get away, or the computer will point, with arrows, toward the section of the array where Bumble is hiding.

Hidden Treasure and Tic Tac Toc

These are masterful programs. They turn arrays into a playground — a landscape where you have your adventure. You (and your child) will be having so much fun playing the games you will forget you are dealing with negative numbers and mapping array coordinates.

As you explore deeper and deeper into Bumble's world, you are challenged with bigger and bigger arrays. The numbers and letters get smaller, and your mapping and direction-following skills have to get sharper.

If you are a good mapper and direction-follower, you will reap all sorts of delightful rewards. You will beat your partner (or the computer) at a game of four-dot Tic Tac Toc. You will find underwater treasures, and you will learn how to create beautiful clowns, tulips, mushrooms, boats, whales, and stars, by connecting dots.

By the end of your journey through Bumble's world, you will be confidently entering negative numbers, X (column) and Y (row) coordinates, and dealing with 20-by-20 (400-dot) arrays. When you are done, arrays will look like arrays, rather than cute bumblebee hiding places and playgrounds.

The biggest reward for completing the Bumble activities will come after kids leave Bumble's world. Any kid who has plotted coordinates on a 400-dot array should find following directions on tests, homework sheets, and other teacher handouts a piece of cake.

Keeping It Simple

Learning Company programs have a unique, very appealing quality: They are simple and *clean looking*.

Many computer companies, comments John Victor of PDI, "get drunk doing tricks on the computer." The computer can make all sorts of dazzling pictures and sounds, so the companies incorporate all these pictures and sounds in their programs.

The problem is that the pictures and sounds often do not make any contributions to the software's educational goals. (They do not help teach the child.) Instead, they may actually serve a negative purpose by covering up the learning activities in the program with excess music, noise, and images that distract the child from the program's learning exercises and objectives.

Learning Company programs, on the other hand, use only simple sounds and bright, simple shapes. Victor believes this is what kids prefer.

Imagination and Storytelling

Learning Company programs respect children's imagination and respect children as people. The programs act like nice people. *Juggles' Butterfly*, for example, begins by asking the child's name. The program refers to the child by name throughout the activity. The program always says good-bye to the child when the child tells it he or she is done.

Children's Software

All the programs introduce imaginary characters in a storybook format. The simple pictures encourage children to use their own imaginations to make the characters more lifelike.

The storybook format gives the child a real-world context for each learning activity. The activity doesn't take place in a vacuum or a world of pure abstraction. Instead, the child is encouraged to interact with bumblebees, butterflies, clowns, rainbows, and kites, as if they were companions in an adventure.

The storybook format makes the learning exercises more meaningful for the child. The simple pictures stimulate the imagination of children and enable them to tailor and mold the characters to the picture they have in their own mind's eye. This makes the activities and the stories even more personally relevant to the child. The stories, the characters, and the pictures aren't complete until they use their imagination to complete them.

The Learning Company
545 Middlefield Road, Suite 170
Menlo Park, CA 94025
(415) 328-5410

The Electronic Chalkboard

The PowerPad, a touch-sensitive tablet in a 20-inch-by-17-inch hard plastic case, is one of the most exciting new products on the market. The PowerPad plugs into VIC-20, Commodore 64, IBM PC and PCjr, Apple, and Atari computers.

The PowerPad can replace the keyboard as the primary means of inputting information into the computer, especially for children. It has numerous mylar overlays which easily clip on top of the 12-inch-by-12-inch touch-sensitive pad. Each overlay is a new keyboard, a keyboard with colorful shapes and figures.

The PowerPad has only a few "keys" on each overlay, so the child or the beginning user isn't overwhelmed by choices.

The "keys" are large and in bright primary colors. Printed on them are bold words (RED or PLAY) and symbols such as # or *. They are separated by plenty of space to make typing simple.

The PowerPad features a novel design in which signals from the tablet are digitally sensed and encoded. A grid of 14,400 contact points (100 points per square inch) is sandwiched inside the PowerPad. The points can record one contact at a time or a dozen contacts. Older touch pads can sense only one finger at a time. But kids (and adults) can put *all* their fingers on the PowerPad at the same time, and the PowerPad will sense all of them and transmit the proper signals to the computer.

This capability is especially nice with the PowerPad's music software and overlay, *Micro Maestro*. The overlay has a piano keyboard at the bottom and a musical score in the middle, with colorful "buttons" for each note on the score. A child can play a musical chord on the piano keyboard by pressing all three fingers on different keys at the same time. This would be impossible on any other touch pad.

Another software-and-overlay package, *Leo's 'Lectriç Paintbrush*, helps a child create colorful pictures on the computer. Again the multicontact feature of the PowerPad becomes especially useful. Before drawing each new part of the picture, the child can press the Pen Up button on the upper left-hand corner of the PowerPad. Immediately a pen tip appears. The child can change the color of the ink being used by pressing one of the colorful paint keys on the upper right-hand corner of the board. When the child picks a new color, the color of the pen tip changes to that color.

This family is using the PowerPad with the overlay for *Leo's 'Lectric Paintbrush,* which allows you to "paint" colorful pictures on the computer with your fingers. Using the "buttons" at the top, you can create shapes, move them around on the screen, create copies, and then save an entire screen onto tape or disk.

Leonardo's Library

The PowerPad is supported by a large and growing body of educational software, called Leonardo's Library. Programs in the library will cost between $25 and $50. The library will include programs focusing on visual arts, music, math, science, language arts, and social studies. Included among the first programs are:

- *Leo's 'Lectric Paintbrush.* An electronic fingerpainting kit.
- *Micro Maestro.* Turns the PowerPad into a piano keyboard.
- *Music Math.* Lets children explore the relationships between math and music.
- *Programmer's Kit.* Lets older children and adults write their own software and develop their own creätive uses for the PowerPad.

PowerPads and programs in Leonardo's Library are already available at K-Mart, Apple computer dealers, and many other computer and discount stores. If you would like to know more about the PowerPad, you can contact Chalk Board directly:

Chalk Board, Inc.
3772 Pleasantdale Road
Atlanta, GA 30340
(404) 496-0101

Chapter 4

Watch Me!

How to Get Intimate with Your Computer

If you are intimate with your computer, you are comfortable, cozy, even attached to it. You know enough to put the computer to work, but you don't have to know *how* it works. Computer intimacy is a totally new relationship between people and computers, one made possible by the new developments in computer hardware and, especially, software. Once we are intimate with our computers, many of us will also want to become computer literate. But not all of us. Nor will we need to.

Many adults envy children's relations with computers. The myth is that children are computer whizzes, that they are computer literate. But this is untrue. Most children are no more computer literate than most adults. What they are is computer *intimate*. They like computers. They have a warm, affectionate, and playful relationship with computers. They don't fear computers. They aren't overawed. To them the computer is just a snazzy appliance or toy, a cross between the TV set, the typewriter, the piano, and building blocks.

Children move rapidly toward computer literacy because they become intimate with computers first. This is the same path adults should follow. Adult computer courses make the mistake of skipping the intimacy part and moving right into computer literacy. But in most circumstances, this strips computers of all their fun.

The first impression the average adult has of a computer is just as he or she imagined: The computer is technical, dry, and complicated. Adults know that it is for their own good to become computer literate, but that doesn't mean they want to. No wonder the adults look enviously at the children. The children look like they are having fun. For them, learning about computers is exciting, hilarious, and very rewarding.

But why should children have all the fun? For many adults, computer literacy is a huge roadblock that separates them from learning more about computers. We should clear away this roadblock and start adults in the right direction, and introduce them to programs modeled after children's programs, programs that promote computer intimacy.

Establishing a Balance

In the job market of the 1990s and the twenty-first century, very few people will be computer literate, if by literacy we mean having the ability to create real, nontrivial computer programs. Yet most people will need to be computer intimate. They will need to be able to work with computers — confidently, comfortably, efficiently, and sometimes even joyously.

But this doesn't mean we should abandon computer literacy. Computer literacy is not just a technical skill for a few mechanics and specialists. It is a doorway that many should enter. Then they can begin using the computer to its fullest potential. For people who are computer intimate and literate, the computer can become a medium for self-expression, a "new age" culture for creativity and communication, and an environment for invention.

Children, especially, should be encouraged to move beyond computer intimacy to a higher level of computer literacy (appropriate to the sophisticated software tools that will be running on computers of the future). Many will not want to go, and they shouldn't be forced. *They will not need to be computer literate to live happy, productive lives in the future. Computer intimacy will suffice.*

However, as a social goal, computer intimacy is not enough, not if our culture is to keep evolving, changing, and responding to the challenges of the present and the future.

The Magician's Top Hat

How do we see computers? Today most children and more and more adults see computers as a magician's top hat. All the new software cartridges, disks, and tapes are stuffed inside the hat, like white doves, flaming scarves, brilliant-colored parrots, and soft, fuzzy bunnies. You can reach into the computer "hat" and pull out almost anything you can imagine — word processors, adventure games, file managers, video paint kits, turtles, and electronic pianos.

And the software industry is growing like a colony of healthy bacteria. In the future we will be able to pull a thousand times as much out of the magic hat.

But what fuels the software industry? What is its source of dynamic power and energy?

Computer literacy. Not among a handful of computer scientists and experts, but spread across millions of computers and millions of users. Computer literacy is the training ground for computer invention. And computer invention makes computer intimacy possible — at higher and higher levels.

Mass-produced microcomputers and increasingly sophisticated software tools have unleashed the imaginations and enlivened the ambitions of an army of youthful, would-be inventors. The inventors are firing off their software inventions like fish launched from a host of catapults mounted on canoes rushing down a swiftly moving river.

Millions of Computer Inventors

The personal computer is more than a magician's hat. It is also a miniature toolshed, workshop, or laboratory. And as personal computers become less isolated, and enable their users to communicate with each other, they will become a *round table,* a forum for people to bounce ideas off each other and then implement those ideas, jointly, as new computer software.

Computer literacy — appropriate to new, higher-level computer tools — is needed in the future, not just among a few experts, but among thousands and millions of young inventors with fresh ideas and with the energy and self-confidence to turn them into computer inventions. And computer inventions will be woven into the fabric of our economy, our society, and our lives.

So computer literacy is necessary. All children should get a crack at becoming computer literate, at the youngest possible age.

Yet computer literacy still does not come first. Computer intimacy comes first, especially for the majority of adults who are scared to death of computers, yet realize that computers are the wave of the future. The strident cries for universal computer literacy only increase these adults' fears. For these adults, computer literacy is not the answer — at least not yet.

A New Religion

Computers are powerful new machines, so powerful that they are treated by many people as a new "religion." Computer enthusiasts are the evangelists for this religion, and they are winning converts by the millions.

Most adults, however, have mixed feelings about computers. They see computers for what they are. Computers are valuable tools and servants, but they are not the most important thing in life. Computers are not an end. They are merely a means to more important, human-defined ends.

Also, computers, like any other powerful and pervasive technology, are valueless in themselves. Whether their impact is good or evil depends on how they are used.

Most adults have a very healthy skepticism and distrust of computers, especially when the "true believers" market them as a necessity and tout them as a new religion.

Most adults do not need a startling plunge into the icy waters of computer literacy. First they need to get their feet wet. They need to follow in their children's footsteps. They need to play with computers, learn with computers, and have fun.

The Myth of the Klutzy Adult

A pervasive and pernicious myth is being spread unthinkingly throughout our society. The myth is that our children are whizzes with computers, but we adults are klutzes. This myth is almost completely ungrounded in fact. Why are children so good with computers? They are good because they see only the colorful, musical, exciting side of computers. The first time they meet a computer, it is wearing a smile.

Children are spurred to master computers because they are so attractive. When we adults see this side of computers, we, too, can master computers just as fast, just as happily as our children.

Trust Your Feelings

What is computer intimacy? What is intimacy? Intimacy is a gut feeling. You know you have become intimate with your computer when you are totally comfortable and relaxed with it, when using it becomes a pleasure rather than a chore, and when you develop excuses just to spend more time with it. When you begin to think your computer is lovable, that's when you know the two of you are becoming intimate.

Getting to know a computer can be like getting to know an attractive yet intimidating member of the opposite sex. I think there is a great similarity between my first experiences with girls and dating and the average person's first experience with computers. When the average person first looks at computers, he or she feels the same sense of fear and anxiety that I felt when I gazed across the gymnasium floor at the girls clustered on the opposite side of the room. That was my first school dance. Computers evoke the same sense of shyness, yet they can also be tremendously attractive, even seductive.

A New Love Affair

For almost 75 years, Americans have had a love affair with their cars. Computers will soon be like cars. Like cars, they will remain machines, and our servants, yet they will also have an emotional, gut-level appeal that will turn people on and bind them to them.

The kind of car we drive depends on the kind of person we are or would like to be. Our car's appearance, model, and year often accurately reflect our values and the kind of image we want to project to fellow human beings. Cars project all sorts of images. They can be inconspicuous, efficient, and sedate, or they can be clunky ragamuffins. They can be flamboyant, garish, and ostentatious, or they can be sensual and adventurous.

Computers, too, will soon reflect our lifestyles, values, and self-images. They will also reflect our needs. Like cars, computers will come with model names pulled from the animal kingdom. Depending on our needs, we'll buy a Cobra (fast as lightning), or a Hippopotamus (it digests huge quantities of information), the St. Bernard (it saves your life in tight situations), the Peacock (it really struts its stuff), or the Donkey (slow and stubborn, but real dependable).

Computers, like cars, can evoke a passionate attachment, a rush of affection. But to inspire real intimacy they must throb to life at the turn of a key, and they must get us where we're going — the faster the better.

A computer can be seductive and lovable, but it is not an end in itself. Many people can get excited about a computer for its own sake. Many more, however, can get excited about using a computer to have fun, get work done, and communicate with other people.

In the future, computers will promise even more than they do today. But let's make sure they keep those promises.

More Than Tools

Computers are only machines, but they are more than tools. A hammer is a tool. So is a broom. But can you get intimate with a hammer or a broom? Not easily. Yet it's easy to get intimate with a computer, because computers are more than workhorses and tools. Computers obey our commands. They carry on conversations. They listen to us. They are infinitely patient. They can be friendly, playful, even silly.

Friendly computers? Playful computers? Silly computers? Where do you find them? Just ask a child. Children love computers because they use computers to learn and have fun. But why can't adults learn on computers, too? And why should kids have all the fun? Adults who peek over kids' shoulders at their programs find that the programs are challenging, enjoyable, and enlightening. Adults can use these programs, overcome their fears about computers, and relieve their computer anxiety. Adults can get to be just as good with computers as kids, and they can have just as much fun.

Computers That Frown and Look Mean

Most adults still think that computers are dry, cold, and unfriendly. No wonder! Most computers in the past were number crunchers, bill collectors, and tax watchdogs. Even today's computers, in their heart of hearts, do nothing more than juggle ones and zeros. But computers don't have to be technical and boring. They can be funny — if you just add people. The relationship between computers and people is often hilarious, if we keep a sense of humor. It pays to look at the lighter side of this relationship, and if we do, we find it helps to break down the barrier of fear separating us from the computer.

Coming Out of the Closet

In recent years all sorts of groups have come out of the closet and have honestly revealed who they are and what they stand for. It's time that computer lovers do the same.

As a person who is on extremely intimate terms with his computer (it follows me into my bathroom and into my bed), I'd like to confess here and now one of the most closely guarded secrets of our relationship:

My relationship with computers is not rational.

This is a shocking revelation, but it is true. My relationship with my computers is emotional, quirky, and antic. It is infuriating, enlightening, and silly. It is happy, frustrating, and ecstatic. But it is rarely rational. And I contend that this is true throughout our society among the millions of computers and computer users. A rational relationship between a human being and a computer is the exception rather than the rule.

Take today. My assistant and I were working on a personal computer. We were sailing along, turning out letters, articles, and forms at a swift, productive pace. The world looked bright, and we were happy.

Then disaster struck. The computer made a mistake. The computer's mistake was only a little one. It wouldn't save any of our text files on disk so that we could print them out on the computer printer.

Until it made its mistake, the computer had been behaving itself. I felt very close to the computer and was extremely fond of it.

After the computer made its mistake, I had a change in heart. No matter what I did, the computer wouldn't save or print my files. So I hated the computer. I called it names. I threatened to walk out on it, abandon it, put it up for adoption.

Now, I ask you, does this sound like a rational relationship?

A rational relationship must have at least two parties who are rational. First we look at the first partner — the human being. Occasionally, philosophers have proposed that humans are rational, but most of us know otherwise.

Next let's look at computers. This is more of a problem. Computers are incredibly complex machines, composed of millions of interacting circuits and thousands upon thousands of operating instructions, rules, and conditions. Computers are too complex to be simple, too complex to be totally rational.

Nevertheless, people think they are rational. For example, the popular wisdom now contains two catchy phrases that most people unquestioningly believe:

> **First**: Computers don't make mistakes. Only people make mistakes.
> **Second:** Computers do only what you tell them to.

As I mentioned, I am extremely intimate with computers. Since I am in this privileged position (along with two or three million children), you would think that I would be able to see through the popular wisdom and realize that the two catchy phrases above are pure hogwash — myths and nothing more.

Alas! I am as much a victim of these myths as the next human being, at least when I am working with my assistant. Whenever anything goes wrong while she is using the computer, whenever the computer acts strange, whom do I blame? Why, her, of course.

Take the other day. I was upstairs in my study and Darshi, my assistant, was downstairs in the office. All of a sudden, she yelled, "Fred! Come quick! The computer's going crazy!"

Sure enough, the cursor was rolling across the screen wiping out the words almost like Ms. Pac-Man rushing around devouring dots. I pushed every button on the keyboard, but nothing worked. The cursor was determined to eat the whole file.

Finally, in desperation, I turned off the computer. Then I turned toward Darshi. "What did you do?" I said, in a not very friendly voice. "The computer was fine just a few minutes ago. You must have done something to mess it up."

Sadly, this was the last thing Darshi needed to hear. She was already extremely timid around the computer and afraid that the next button she typed might blow the computer up. When I accused her of her worst fear, she looked ill and ran out of the office.

Why had I blamed Darshi? I blamed her because she was a beginning user and a human being. Those two facts alone were enough evidence to convict her.

Sometimes computers are agreeable, responsive, and meek. They do everything you tell them to. But then, a moment later, without any warning, they turn on you. They suffer amnesia. They pout and get sullen and write gobbledygook all over your lovely files. Or they act crazy and start doing awful things like eating up the words on your picture screen. And they get out of control. Then the only way to get their attention is to switch off their power. This, of course, is an undesirable and drastic solution. But, sometimes, it's the only way to make them come to their senses.

A Little Breathing Room

When you are in the middle of a squabble with your computer, it doesn't seem very funny. However, after things have quieted down, and you look back, you might be able to put things into perspective, and maybe even laugh about them.

But one thing you should not do is pretend that you and your computer have a rational relationship. It is anything but that. It may be quiet, sedate, and low key. Or it might be wild and boisterous. But it is not rational. It can't be. You're not rational. The computer's not rational. So how can your relationship be rational?

The sooner people stop looking at their relationship with computers as rational, the sooner they will become intimate with computers and learn to accept them for what they are. Computers are moody and complex creatures. But they try hard to please you. They really do.

Computer Popcorn

Last night I woke up in the dark with my head spinning. I turned to the digital clock beside the bed. It said 3:00 a.m. Musical notes and rainbow-colored rubber bands bounced around inside my mind. Over and over, a little voice inside me kept repeating two words: "Computer popcorn. Computer popcorn. Computer popcorn."

The voice told me that I was supposed to get out of my warm bed and go into my dark, cold study. And what was I supposed to do when I got there? I was supposed to write about *computer popcorn*.

When I protested that it was the middle of the night and that I didn't want to get out of bed, the voice became surly. "If you don't get out of bed," it said, "you'll forget everything by morning."

"Forget what?" I thought.

"Computer popcorn," said the voice. "Computer popcorn is a computer program that is so good you can't put it down. You can't stop thinking about it. You even dream about it."

Dreaming in French

At different periods of my life I have become so obsessed by and so immersed in a new subject that I can't stop thinking about it. I even take it to bed with me at night.

For example, I spent a couple of months backpacking around Europe one summer when I was in college. I spent the first night away from home in a hostel in Paris with a lot of my college friends. The next morning they all yelled at me. "You kept us awake all night," they complained. "We don't know what you said, but it was all in French."

I used to be an international relations major. I learned a lot of languages and visited a lot of countries. When I visited Mexico I dreamed in Spanish. When I went to Brazil I dreamed in Portuguese.

I doubt if my French, my Spanish, and my Portuguese were grammatically correct, and I'm sure my pronunciation was not perfect. But the important thing is that I was excited about exploring a country and a new culture — so excited that I continued my exploration even while I was asleep.

Watch Me!

Doorways into New Worlds

This experience is a little like a religious conversion. It is a sense of rapture that you feel when you throw open a door and see endless vistas you've never imagined. Then you step through the door.

This is the feeling I've gotten recently from some of the new programs and computer peripherals I've been reviewing. In this section I'll take a look at one of these programs (*The Music Construction Set* from Electronic Arts) and a combined program/peripheral (the KoalaPad and the *Micro Illustrator* program from Koala Technologies).

Mechanical Drawing

When I was in high school, I took two classes that I thought were particularly painful. One class was mechanical drawing. The other was geometry. I found these classes so agonizing because they both involved the painstaking, *precise* drawing of geometric figures. In geometry class we mostly stuck to two-dimensional figures like squares, triangles, circles, and polygons. In mechanical drawing we began with blocks and cubes, and ended up drawing spaceship nose cones, automobile crankshafts, and "exploded" watch gears.

Surprisingly, I got good grades in both classes. I got the grades because I was such a perfectionist. I would struggle with the assignments for hours and finally turn out a beautiful, finished drawing.

But I hated every minute of it. By the time I finished doing the drawings my arm, wrist, and finger muscles felt so cramped I thought I would go crazy.

And I never thought about what I had drawn. I was too exhausted just getting the shapes down on the paper. The engineering and mathematical concepts underlying these drawings went right over my head. I never even considered them.

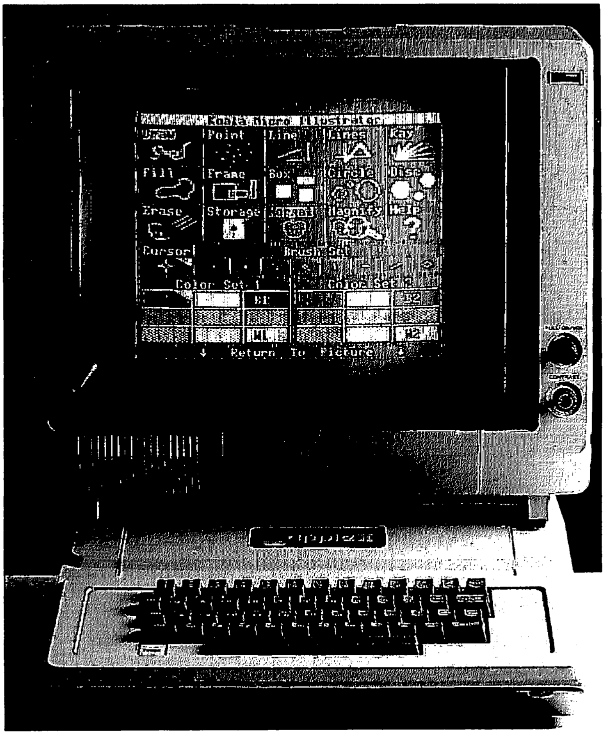
Rainbow-Colored Rubber Bands

With images of nose cones and polygons floating through my mind, I sat down with the KoalaPad from Koala Technologies (3100 Patrick Henry Drive, Santa Clara, CA 95050, 408/986-8866).

You can buy a KoalaPad for from $110 to $150 (depending on the computer) and all supporting software packages for less than $50. Different versions of the KoalaPad are made for the IBM PC; the Apple II, II+, and IIe; the Commodore 64; and the Atari 400, 800, and XL computers.

The pad is smaller than a TV dinner and weighs about as much as a paperback book. You plug the pad into the joystick port of your computer, and you hold it in your hand or lap while you draw, using either your finger or a plastic stylus that comes with the pad.

The KoalaPad comes with the *Micro Illustrator* program from Island Graphics.



With *Micro Illustrator,* drawing geometric shapes and figures is easy. The program, which is used with a touch tablet, makes picture creation effortless.

"Growing" Circles and Boxes

The KoalaPad and *Micro Illustrator* are *computer popcorn*. They're delicious! Once you and your family start using them, you won't be able to stop.

When you boot up the *Micro Illustrator* program, you see a menu of lots of little boxes with words and pictures inside. Each box is a doorway into new worlds of self-expression for you and your family.

With a KoalaPad and *Micro Illustrator,* all of a sudden making geometric figures is easy. *Micro Illustrator* encourages you to make figures of great beauty and complexity. It's seductive. It's so easy to draw elaborate, symmetrical shapes that you keep thinking: What if? What if I connected these two lines, or what if I created some circles over here? What if I rotated this figure and colored it in?

My family and I have had the KoalaPad and *Micro Illustrator* for only a week, but, already, each of us has used them for several hours apiece.

And we still haven't explored all the features. I love the circle, disc, box, and frame commands. Using these commands you can "grow" geometric shapes in seconds.

Our favorite feature is the mirror. The mirror lets you draw simultaneously in four directions. Combine the mirror command and the line command and you can create glowing rubber-band lines that stretch like a net across the screen. Tack the circle command onto the mirror command and you can draw hosts of rotating circles. In no time at all you can create beautiful patchwork quilts, ornate tiles, bug-eyed aliens, and solar systems full of planets and moons.

The KoalaPad and *Micro Illustrator* are marvelous skill and imagination amplifiers. They allow me and my family to do things we could never do on paper. And they make it so effortless that we don't have to concentrate on the mechanical aspect of creating new shapes and pictures. We are free to create and to discover, and when we're finished, we're still fresh enough to be amazed.

The proof of how amazing these products are is how proud we are of what we create. The person on the computer is always calling to the other members of the family: "Come here, everybody! Look what I've drawn! You've got to see this one!"

Mechanical Bach

When I was seven years old, my mother started me on piano lessons. Maybe I wanted to learn about the piano at the time. I really can't remember, because the original joy of making music was quickly submerged by the daily grind of practicing and the weekly pilgrimage to the music studio where I suffered under the harsh tutelage of a nonstop stream of boring and unimaginative music teachers.

The teachers weren't really so bad. It was the method I hated. Like any kid, I had aspirations to create my own music, to make beautiful, original sounds that expressed how I felt and what I thought. But all I ever did was mechanically translate the printed musical notes of mediocre songs from the scores onto the piano keyboard.

I never realized that my teachers weren't treating me like a human being. They were treating me like a machine — a music player, like a player piano. I learned how to read other people's "frozen" music and then miserably try to reproduce it on the piano keyboard. The problem was that I didn't care for the music I was playing, and the sounds I made rarely pleased me. If I liked music (as I did) it was a lot easier to go to a record store and buy a record. Then I could hear the music I liked and it sounded right.

At some level I realized that my original purpose had been perverted. And like any decent human being, I made a very bad machine. I repeatedly showed up for class late, I never practiced, and I never played a piece the way it was written.

This used to drive piano teachers crazy, and they never lost the opportunity to tell me how little musical talent I had.

I didn't care. I would rather have been playing baseball or touch football. Anything rather than have to practice the G-major scale for another half-hour.

The Music Construction Set

The Music Construction Set is Will Harvey's response to piano teachers who teach their pupils to pretend they are machines. Will Harvey is a 16-year-old from Uplands High School in Foster City, California. When asked why he wrote *Music Construction Set,* Will replied: "It was something that needed to be done. I wanted someone who didn't know anything about music to be able to learn it simply and have a lot of fun doing it. I also thought it would be great if you could save what you wrote."

According to Will, his program is "simple, hot, and deep." By this he means that it is easy to use, it appeals to a person's senses, and it can grow with a person. The program is enchanting to musical novices as well as musicologists and musicians.

Music Construction Set (MCS) retails for $40 and is published by Electronic Arts (2755 Campus Drive, San Mateo, CA 94403, 415/571-7171).

MCS currently runs on the Apple IIe, the Commodore 64, and the Atari computers.

If you plan to use *MCS* on an Apple, you should consider a special offer by Electronic Arts. You can buy a Mockingboard stereo sound card from Sweet Micro Systems, for $100 ($25 off the regular price of $125). The Mockingboard lets you create polyphonic sound on the Apple. That means you can create chords with up to six notes playing at the same time. And you get stereo, too. Sweet

Micro Systems is at 150 Chestnut Street, Providence, RI 02903 (401/273-5333).

If you have the Commodore 64, you have the SID (sound synthesis) chip, and you don't need the Mockingboard. When you run *MCS* on the Commodore 64, you will be able to compose and play music with up to three notes playing at a time. On the Atari, you will be able to create music with up to four notes playing at a time.

The other add-on to *MCS* that I enthusiastically recommend is the KoalaPad from Koala Technologies (see review above). You use the KoalaPad to move the notes and other musical symbols around on the picture screen. You can also move symbols using a joystick or the keyboard. But the *easiest* and *fastest* way to move musical symbols is using the KoalaPad.

This is how *MCS* works — on the Apple II+ with the Mockingboard and the KoalaPad:

First, you boot up the *MCS* disk, and you see empty musical staffs at the top of the screen and a pictorial menu on the lower half of the screen. If you do nothing, the computer starts playing music itself, as if you have just put a record on your stereo. You hear ten songs, including Pachelbel's "Canon," Rimsky-Korsakov's "Flight of the Bumblebee," and the "Pat the Hat" rag by Douglas Fulton. Then the music starts again. It will keep playing until you press the RETURN button.

Since you're into creating your own music, you immediately press RETURN. The next thing you do is press the plastic stylus on the KoalaPad down on the pad. Immediately a pointing-hand "icon" appears on the screen. You use the hand to "build" your song.

You move the stylus point across the pad and the hand moves across the screen. When the hand on the screen gets to an eighth note (there are also whole notes, half notes, quarter notes, sixteenth notes, and thirty-second notes), you press the top left-hand button on the KoalaPad. The note "jumps" into the hand. You move the hand onto the empty musical staffs and position the note in the E-note position on the treble clef. When you let go of the KoalaPad button, the note falls out of the hand and glues itself to the staff.

You can do all this in just a couple of seconds.

Then you move the hand back to the menu of notes, rests, sharps, flats, ties, octave raisers, and time signatures, and pick them up, one at a time, and deposit them on the staffs.

When you are finished creating some music — up to 1400 notes and up to 70 measures — you move the hand across the screen to point at the picture of the little house ("Home") and press the KoalaPad button. The musical score on the picture screen *scrolls* to the left, back to the first measure — the beginning of the music. Then you move your hand to point at the picture of the grand piano. You press the button, and the song you just created plays — in stereo.

This is just the beginning.

Cut and Paste

Now that you have created a song, you can play with it. By moving the hand around the screen and pressing the KoalaPad button, you can speed the music up, raise or lower the volume of each speaker, change the type of sound from regular to smooth, to vibrato, to drumlike. With the push of a button, you can transpose the music to other keys and replay the music in each key.

And you can use *MCS* like a word processor to *cut and paste* measures of music. On the screen are a little pair of scissors and a paste pot. Using them, you can cut up to nine measures out of the beginning part of your song and move them forward or backward in your song.

This is one of the most exciting parts of *MCS*. As I said earlier, when you play your music, the measures filled with notes scroll by on the picture screen, from right to left. As you listen to the notes you also watch them scroll by. Playing music becomes an effortless experience that is visual as well as auditory. You can concentrate on hearing and seeing the notes, not just playing them. It's a great joy (to an ex-player piano like myself) not to have to concentrate on stretching your hands and positioning your fingers to get each note right. The computer takes care of these details for you.

Since music now becomes a visual experience, you can begin perceiving patterns visually as well as by sound. And if you like certain patterns, you can repeat them in the music by using the hand and the scissors to cut the measures in which they appear and "paste" them into other places in the music.

When you are done creating your own music, you can fool around with it. Then you can save it on disk. And if you have a printer, you can print out a copy of the score.

Poppin' Hot

Now you see why I have trouble sleeping at night. You can see why I go to bed and dream about rainbow-colored rubber bands and dancing musical notes.

Perhaps you can also see why the little voice inside my head so persistently kept telling me to write about "computer popcorn." The KoalaPad, *Micro Illustrator,* and *Music Construction Set* are like popcorn. They are so much fun and taste so good, once you start with them you just can't stop.

Winnie the Pooh's Alphabet Adventures

One afternoon while Eric was riding his Big Wheel bike on the sidewalk in front of his house, a brown UPS truck pulled up, and a man hopped out and put some giant boxes on Eric's front porch. Eric went and got his dad. His dad told him that inside the boxes was a new NEC Trek home computer that had been sent, on loan, from the NEC Home Electronics Company in Elk Grove Village, Illinois.

They set the computer up in Eric's bedroom. He liked the computer. It was neat to look at, with its ivory case and its gray and orange keys. It was easy to use, too. He used its Micro Painter program to make pictures and its Electric Pencil program to do lots of gobbledygook processing.

The NEC Trek was special, too, because it had games with all of Eric's favorite Walt Disney characters. He wanted to play the games and see Mickey Mouse, Donald Duck, Cinderella, the 101 Dalmations, and Winnie the Pooh. But, so far, he still hadn't played any of the games.

His dad had lots of excuses to explain why the games weren't ready. He mumbled something about RAMs and ROMs and an Extended BASIC cartridge that hadn't arrived.

Eric already had a game disk with the word WINNIE written in big letters in blue ink. The disk had a game called Winnie the Pooh's Alphabet Adventures. But his dad told him that they still didn't have a disk drive to put the disk into.

He really wanted to see the Winnie the Pooh program, so he put pressure on his dad to get his act together and find the equipment they needed to make the program work.

Pretty soon, more big brown boxes started arriving in the mail. Eric loved opening boxes. He had never run into a box he couldn't open. When he was only six months old, his parents put a box around him, with holes for his head, legs, and arms.

On the side of the box his dad drew, in big letters, the words PAPER SHREDDER. And he drew lots of pretend dials and switches. It was Eric's first Halloween costume. He went to three Halloween parties, crawled around on the floor, and shredded any paper that he found in his path.

But Eric wasn't a baby any longer. He was four years old, and he could shred boxes the way he used to shred paper. When the computer boxes arrived, he opened all of them with his bare hands. Inside the boxes were the computer parts his dad had told him about. He helped his dad attach all the parts to the main computer that was sitting on a little table in Eric's bedroom.

Run, Winnie, Run!

Finally a box came with the last part. Eric huffed and puffed and "Hulked" open the box. Then he and his dad raced to his bedroom to put the missing part into the computer.

His dad turned on the power. The computer worked! Eric hopped around the room. He almost fell on the computer, he was so excited.

His dad let him put the Winnie the Pooh disk into the disk drive. He let Eric do everything on the computer all by himself. While he was working on the computer, sometimes he saw his dad put his hands over his eyes. Sometimes he saw him grit his teeth and look like he was going to cry. Sometimes he even heard him growl. But he always let Eric do everything. Because of this, Eric was getting pretty good at computers, even though he was only four years old.

His dad read from the NEC manual for the *Alphabet Soup* package. Eventually there would be two programs in the package: the Winnie the Pooh alphabet game and another game called Mickey's Lucky Stars. Mickey's Lucky Stars would teach Eric how to match small letters in the alphabet with big letters, and help him learn which letters come before other letters and which ones come after.

Eric's dad read the commands from the manual. He repeated the letters, one by one, and Eric typed them into the computer. When he was done, the command RUN "winnie." was on the screen. He pressed the RETURN button to send the command to the computer.

Out of the computer's speaker came the song "Winnie the Pooh," and the Pooh bear himself appeared on the screen. Beside him was a big, yellow, blinking question mark.

Just then the telephone rang, and Eric's dad took off. "I'll be right back!" he called.

"Sure," Eric thought. "In about a million years."

Eric didn't feel like waiting a million years. Besides, he knew what to do next, even without a manual. When he saw a question mark on the screen, that meant the computer wanted him to type something in. "But what should I type?" he wondered. He picked his favorite word: ERIC.

He typed an E, then began searching for the R. But before he got there, the disk drive light came on, the drive began clacking like his Big Wheel bike, and Winnie the Pooh vanished from the screen.

A moment later, a new screen appeared. It was divided into several rectangles, each a different color. The Winnie the Pooh character, Tigger, appeared in the upper left-hand corner of the screen. In the upper center portion of the screen, two E's appeared — one uppercase and one lowercase. On the right-hand side of the screen was an elephant. In the lower left-hand corner of the screen was a yellow box. The box was empty.

All these things appeared on the screen, but Eric didn't notice. He was still busy typing his name. He typed an I and a C, then he looked up.

His dad sailed back into the room. He looked at the screen. "Hey, that's great, Eric," he said. "How'd you do it?"

"By typing my name," Eric answered, not sure whether to be proud or puzzled. "It made an E, but it didn't make an R. Or an I. Or a C."

I Know What to Do!

"I wonder what we do, now," his dad said, peering closely at the screen. The NEC company had sent Eric and his dad about ten pounds of computer manuals to assist them on the computer. But the two of them rarely used manuals, especially when they were just getting started. The fun part of running new programs was to see if they could make them work without reading the instructions.

Eric's dad was naturally cautious around computers. He tried to figure out which button might make the program do something.

Eric had a better approach. When he didn't know what to do next, he pressed *all* the buttons.

His strategy worked. After only a few seconds and a couple of dozen buttons, he found one that did something. He pressed the DEL (Delete) key, and the empty yellow rectangle in the lower left-hand corner of the screen turned blue. He pressed the button again and it turned red. Then it turned green. Each time he pressed the button, it turned a new color.

When Eric pressed the E key, the computer played a little more of the Winnie the Pooh song, then went back to the picture of Pooh and the big question mark.

"Hey!" Eric said. "E makes a picture. Then E makes the picture go away."

He pressed some more keys. He eventually made it up to the orange function keys on the top of the keyboard. When he pushed the F4 key, Winnie the Pooh, Tigger, and Rabbit appeared on the screen and, with musical accompaniment, waved good-bye.

"Oh, terrific!" said Eric's dad, more than a little distracted and disturbed by Eric's shotgun approach to using the computer. "Now you've terminated the program, and we've only gotten to see one letter."

Eric was momentarily stymied. But at the moment he felt like he could do anything — the way he felt when he was rustling up a jellybean, Cheerio, and dry-noodle stew in the kitchen, or tying his shoes, or stirring up Mowie's breakfast of gooky cat food and kibbles. He surveyed the keyboard. Then he was ready. "I know what to do," he said confidently, and began pressing all the keys at the same time.

He got to the F5 key and pressed it. Winnie and his friends disappeared. The title frame came back on. He had restarted the game. He looked up at his dad. "See?" he said.

All It Takes Is Teamwork

Eric and his dad worked well as a team. With their combined brainpower and Eric's penchant for button pushing, they soon figured out how to use the rest of the program.

For example, when Eric pressed the F1 button, the NEC thermal printer started making noises like a tire spinning on ice, and paper started creeping out with a copy of the picture on the computer display screen.

Eric loved this part. Printing pictures was so easy! Very quickly, his bedroom floor filled up with 4-inch by 4-inch scraps of paper featuring all the Pooh characters and creatures whose names began with every letter from A to Z.

Solving the Mystery of the Blank Box

The blank box in the lower left-hand portion of the screen was the greatest challenge. Even when Eric printed out the display screen, the box was empty. Why was it empty? Either the program was broken and the box was supposed to have something in it, or Eric and his dad were supposed to put something in the box themselves.

They tried using the joystick. That didn't work.

They pressed all the keys on the keyboard again. No luck there, either.

They were about ready to give up and peek at the Winnie the Pooh program's instructions. Then they figured it out. They could fill up the box by drawing things on the NEC Trek touch panel, a flat drawing tablet that reproduced a copy of a picture on the computer's display screen.

The touch panel freed Eric from the computer keyboard. And that's when the real fun started!

His dad went into his study and cut up lots of pieces of paper to fit on the touch panel. Two flexible magnetic strips held each piece of paper on the panel so it wouldn't move about.

Eric climbed on the metal truck and, on top of his dresser, found the black felt-tip marker that NEC had supplied with the touch panel.

He began drawing on the panel. He drew circles, triangles, straight lines, and random squiggles. Then, satisfied with his artwork, he pressed the F1 button and printed his picture — complete with a letter of the alphabet (in upper- and lowercase), a picture of an animal whose name began with that letter, and a character from Winnie the Pooh.

Eric then took the pictures he had just drawn and put them onto the touch panel. He created new pictures by tracing the animals and letters on the old pictures. He created drawings that looked reasonably like Winnie the Pooh, skeleton hands, elephants, alligators, and birthday cakes.

For Eric this was a thrill — such a thrill that he drew pictures on the touch pad, picture screen, and thermal paper for another two hours. And the next morning, when he woke up, it was the first thing he wanted to do, even before his all-important bowl of Cheerios.

Drowned in Alphabet Pictures

The night before, after the first hour, little scraps of paper were all over Eric's bedroom. Eric wanted to create a picture for each of his pets (his robot Denby, his puppy, and his kitty), for each member of his family and all his friends. Each picture had the first letter in the name of the person or creature it was going to.

Paper scraps flooded the bedroom, and his dad grew alarmed. He had visions of being drowned by Pooh pictures. He suggested that Eric try to group the papers into piles.

To his dad's relief, Eric came up with the idea to make books out of several of the pictures. The letters could be grouped together to make alphabet books, or to form the complete names of his mother, father, sister, grandparents, cousins, and his pets, creatures, and friends.

He and his dad got busy and turned Eric's bedroom into a miniature printing company. They stapled the pictures together into books, and they taped lots of pictures together on pieces of notebook paper to spell words and names, and make signs that spelled things like MOWIE, WINNIE, PIGLET, EEYORE, ERIC, CATIE, and BACK OFF!

The flow of loose paper scraps slowed somewhat, but not enough. Out of desperation, Eric's dad peeked at the instructions to the Winnie the Pooh program and discovered lots of additional activities using the alphabet-pictures that Eric was churning out of the computer.

They began to use the pictures as alphabet flash cards and played lots of games, including Concentration (guess the missing letter), Scrambled Letters (trying to reorganize letters to make up a word), Letter Match (matching up lowercase and uppercase letters), Tasty Letters (matching up flash cards with alphabet cereal letters), Alphabet Clothesline (taping the letter pictures to a string hanging in the room), Mystery Letters (letting Eric run his fingers along the clothesline, and trying to guess which letter he is pointing to).

The Winnie the Pooh user's guide even had a short BASIC program to type in to create a new game. Eric and his dad typed in the game. It was a Mystery Letter game. It typed a sequence of letters on the computer's display screen, but one letter was missing. Eric had to guess the missing letter. If he got the letter right, his dad let him print the letter out on the computer printer.

Typing with His Toes

The more Eric used the Winnie the Pooh program, the more relaxed and creative he became. In the beginning, he sat stiffly in front of the computer keyboard and picture screen, held the touch panel in his lap, and drew on sheets of paper. But by the end of his first session things had changed drastically. His dad lay on his side, sprawling behind Eric, watching him draw his pictures. Eric decided he wanted to get more comfortable, too, so he climbed up on his dad, using him as a reclining lawn chair. He stopped using the paper and marker to make pictures and, instead, began drawing pictures with his finger on the white, glossy plastic surface of the touch panel. It was like electronic finger painting, and he loved it!

When Eric climbed on his dad the first time, he accidentally kicked the Expansion Panel on the side of the computer. Loaded in the Expansion Panel were a RAM cartridge and the Extended BASIC cartridge needed to run the program. When the Expansion Panel became dislodged, the screen went blank and the program disappeared.

Eric pushed the Expansion Panel back against the computer, but he didn't want to reboot the disk (he'd already done that before), so his dad had to do it. While the program was reloading, Eric did backward somersaults across the bedroom floor.

His dad lay back down. Eric stopped doing his somersaults and climbed onto his dad again. As he was making himself comfortable, he pulled the cord out of the touch panel. His dad saw the cord fall off, but he didn't say anything. Eric spent about a minute making a drawing with his finger before he looked up at the computer's picture screen. The little picture box was still empty.

Eric pushed all sorts of buttons on the computer before he realized that nothing was happening because the touch panel was no longer connected to the computer. This prompted his dad to deliver a little lesson on computer cables as "highways" for the computer's information to zoom back and forth from the computer to peripherals like the touch panel and the printer.

Eric and his dad also discussed the pins on the ends of the cables, so that Eric would know the proper way to plug the cables into the computer and the other equipment.

Eric got the touch panel hooked up. He climbed back up on his dad, dug his elbow into his dad's rib cage, and began drawing. But now the touch panel was upside down. This appealed to him. Everything he did on the touch panel showed up backwards and upside down on the picture screen.

He tried typing the letters in his name. He tried making numbers. He made faces, houses, and robots. Everything appeared on the screen backwards and upside down.

Eric turned the touch panel on its right side and drew pictures. Then he turned the panel on the left side. Then he turned the touch panel over and tried to draw pictures on its bottom. When he found that this didn't work, he improvised by drawing a picture with his knee.

When he was done drawing, he said, "Daddy, please press the print button."

"Phooey!" his dad said. "You're lying on me. How am I supposed to press the button?"

"Please, Daddy?"

When his dad heard that "Please, Daddy?" he couldn't resist. "I'll see what I can do," he said. He looked down at the computer. His bare, sockless foot was only a couple of inches to the left of the keyboard. He lifted his leg carefully (so as not to dislodge Eric and his touch panel) and stretched his big toe toward the F1 button on the keyboard. He missed. The computer made haunted house music to show that he had pressed the wrong key.

He tried again. This time his toe hit the right button. The printer started chugging away and printed Eric's picture.

"Wow!" Eric said, impressed by his dad's display of pedal dexterity. Unfortunately, this gave Eric ideas. It opened his eyes to new ways to interact with computers. He knew that using his fingers was OK, and his sister had once operated her computer using her tongue. But he had never considered using his toes. Until now.

The rest of the evening Eric practiced pressing all the buttons on the NEC Trek with his toes.

He did pretty well, too. And his dad let him do it. But his dad created one rule that Eric had to obey. Before he could continue using the computer, he had to submit to a thorough sponge bath of both feet.

New Family Learning Games

When I was a kid I used to belong to all sorts of mail-order book clubs. When I knew a book was coming I would rush home from school or spend an entire Saturday prowling around my front yard waiting for the mailman to pull up our driveway. It was a great feeling when he brought a big box addressed to me. I knew that a new book was inside the box. It didn't matter that I could never remember what book I had ordered. That was part of the fun.

Ambushing the Mailman

Now my two children (Catie, 8, and Eric, 4) have followed in my footsteps. Catie and Eric get as excited as I did about receiving packages in the mail, and they are as good as I was at ambushing the mailman. On Saturday mornings, they lie in wait behind two big pine trees just outside the porch door. When the mailman arrives they spring out and grab all his packages and run into the house. They reach the living room, and they start ripping the packages apart.

But do you think they are looking for books?

Nope. They are looking for new computer learning games arriving in the mail. And they act like wild things until they unwrap the games, load them into the computer, and begin playing them.

Champions and Cheerleaders

Here, below, is a group of seven games that captivated me and my family. They are remarkably diverse and quite varied in the thinking and skill they demand from the human player. But they are similar in four key traits. First, they are just as much fun for adults as they are for kids. Second, they can be played at many different levels, so, with help from an adult, even a toddler can benefit from them. Third, the games are constructive and non-violent. They let families build things rather than train them in creative destruction. Fourth, the games are much more fun when people play them together.

All these games encourage interaction among family members, schoolmates, and friends. At our house we almost always play the games together. The approach we use is to have one person step forward as the stalwart champion and have the other family members be coaches, cheerleaders, and the peanut gallery. At the end of each game we rotate all the roles.

Playground or Swamp?

It was interesting to see Eric and Catie approach these new games. They never want to read any directions before starting. They equate direction-reading with "adult," "slow," "dense," and "boring." On their own, they never read directions — *unless they appear on the screen*. They just boot up a disk or plug in a cartridge. Then they start madly pressing buttons or swiveling a joystick or game paddle. Pretty quickly something begins happening. Then it's play it by ear all the way.

This sort of approach makes me very nervous. Nevertheless, I usually climb aboard for the ride, just to see where we'll all end up.

Most of the time, the kids wander through a program — *at a gallop* — and usually figure out what's going on. Then they begin playing with a passion.

But sometimes their approach is akin to turning down a blind alley and running, full speed, into a brick wall. Then, with their noses out of joint, the kids turn around to me and announce, "Daddy, this is a dumb game!"

And that's that. When the computer doesn't respond, when my kids feel powerless and out of control, they abandon the game. It's like watching their playground turn into a yucky swamp. While it's still a playground they love to race around and use the equipment. But when suddenly the ground turns into sticky glue, the kids feel paralyzed.

I used to think they had reached a real dead end. Now I look at it as an opportunity to start doling out some game rules and special "power" buttons that get the game moving again and put the kids back in control.

Pipes

Pipes is available on cartridge for the VIC-20 ($29.95), and the Commodore 64 ($34.95), and on disk for the IBM PC and PCjr ($29.95).

It won the 1983 CES (Consumer Electronics Show) Software Showcase Award for Home Education. It is made by:

Creative Software
230 East Caribbean Dr.
Sunnyvale, CA 94083
(408)745-1655

Pipes is a game that never turns into a swamp. When the program begins there is a plumber, a house, and a water supply tank on the display screen. On the far right is a pipe factory with bins full of pipes of all shapes.

My kids were puzzled by *Pipes* at first. But that didn't stop them from leaning on the joystick and racing the little plumber around the screen. They learned how to use the "radar" display — a little window in the corner of the screen that lets you see the plumber, the house, the water tank, and the pipe factory, all at the same time. And, by randomly pressing the joystick buttons, they discovered they could buy pipes from the factory and hook them up to the house and the water tank.

The first couple of times we played the game the kids created some pretty weird plumbing. Pipes squirreled out of the house, then corkscrewed, pirouetted, and pretzeled themselves into oblivion. We found out how to turn on the water supply (by pressing the V key on the computer) and squirted water all over the ground with ecological abandon.

Eventually we ended up with some pretty decent plumbing. The pipes were in efficient right angles out of the water tank and into the house. When we turned on the water, it flowed in a direct route from the tank to the house.

After hooking up the plumbing to one house had become a snap, we graduated to a whole neighborhood with up to five houses. We even figured out how to do the plumbing with the cheapest pipe and save the most money.

Now the kids mostly play *Pipes* alone. The other day I went into the dining room and found Eric busy building a circular pipe network out of the water tank. I frowned and screwed up my face. "Why would you want to do that?" I asked him.

"Because," he said, not looking up, "this way the water never goes away."

Delta Drawing

In Chapter 9 my daughter Catie and I review the Apple version of *Delta Drawing*. Spinnaker Software has released *Delta Drawing* on cartridge for the Commodore 64 ($39.95). You can reach Spinnaker at:

Spinnaker Software Corporation
215 First Street
Cambridge, MA 02142
(617) 868-4700

The Commodore 64 version of *Delta Drawing* is significantly more powerful than the earlier Apple version. And the Apple version was a knockout.

Catie and I found *Delta Drawing* to be a lot like Logo — only upside down! To make the Logo turtle do something, you have to define a procedure (or program) and type in lots of one- or two-letter commands. Then, when you're all done, you have to type the procedure name to make the turtle do its tricks.

This kind of programming is called delayed gratification. It requires a lot of patience — especially when you are only four years old.

Delta Drawing is just the opposite. The payoff comes at the beginning *and* at the end. Here's a typical session with Eric:

Eric plugs the *Delta Drawing* cartridge into the Commodore 64, and, a moment later, a triangle and a blinking dot appear in the center of an empty screen. The triangle is "DeeDee" the turtle. The dot is DeeDee's tail. DeeDee uses her tail to draw.

Eric starts DeeDee on a trip across the screen by pushing the D key (for Draw). DeeDee moves about a quarter of an inch up the screen, then stops. Behind her is a white line.

Eric pushes the D key again, then the R key (for turn right 30 degrees) three times. Then he pushes the S key.

Eric has made DeeDee do something significant by pushing just five buttons. First, he has made DeeDee move and draw a line — as soon as he presses the button. (This is called immediate gratification.)

Second, he has just created a *program*. The program is extremely simple, but it will act as a building block for the shapes that Eric is planning to make DeeDee draw next.

Eric saves his program by pressing the S key. (At this point Eric's daddy likes to press the T — Text — key to see the actual commands Eric has given DeeDee. This reassures Eric's daddy that Eric is, in fact, creating a real program. Eric, however, is confident that he is programming even without seeing the list of commands.

He can see that his programs are working by watching DeeDee whiz around the screen drawing the shapes he has dreamed up.)

When Eric presses the S key, the screen goes blank and DeeDee reappears in the home position. With only a moment's hesitation, Eric presses the X and the 1 buttons to run Program 1. DeeDee spurts forward two paces and turns right. Behind her is the straight line.

Eric presses the X and 1 buttons three more times. When he is done, DeeDee is back in her home position. She has just drawn a square. Eric types the R button to turn DeeDee 30 degrees to the right. Then he types an S to save his second program.

Next Eric presses the X and the 2 keys seven times to run Program 2 seven times. When he is finished, he smiles. DeeDee has just created a flower made up of little boxes rotated around a central axis.

Is Eric done? Not yet. He likes flowers so much he wants them all over the screen, and he wants them in different colors. He presses a couple more keys and colors the flower petals orange and blue and green. Then he presses the M button and holds it down. DeeDee scoots up the screen. Eric presses the S button to save his third program.

Now he's finally ready to do his picture. To make the picture, he uses the building block Programs 1-3 that he has just created. To fill the screen with colorful flowers, he has to press only two keys: the X and the 3. Each time he runs his third program, DeeDee draws a flower, colors it in, then zips to a new part of the screen.

Pretty soon Eric and DeeDee have filled the entire screen with flowers. Eric is done. He gets up from the computer and goes looking for his family to show off his latest creation.

Delta Drawing is a spectacular learning game. I have described only a tiny bit of what kids can do with it. But the neatest thing about *Delta Drawing* is that children can explore all its powerful features, or they can spend hours on a single part of *Delta Drawing* and still not exhaust it. The program is made for children to explore. And if my children are any guide, they love doing it.

Kids on Keys

Kids on Keys is available from Spinnaker Software. The Commodore 64 disk costs $29.95; the Commodore 64 cartridge costs $34.95.

Kids on Keys is one of those programs that my family ought to like. It teaches all sorts of good things like the alphabet, shapes and colors, problem-solving, and the computer (or typewriter) keyboard.

Watch Me!

It turns out that my family really does love *Kids on Keys*, but not because it teaches all that sound educational stuff. We love it for lots of little intangible reasons, like the neat music it plays. Or like the little person who whimsically floats up and down in a balloon. Or the way the letters we correctly identify make a loud *BURP!* and crumble like cookies. Or the funny way the cats, rabbits, boots, and faces fly off the screen after we correctly identify them.

Somehow, subtly, and disarmingly, *Kids on Keys* is charming. So we all love to play it. (Even though some of us are 34 years old, and we're supposed to know our alphabet already.)

And for those adults out there who are snickering in their sleeves, I dare you to try *Kids on Keys*, Game 3, Level 4. Just try to guess all those fragmented shapes, especially after they have changed color and scrambled their positions. Let me tell you, it is no laughing matter. Especially since the key word is quickly fading away.

How well-developed is *your* skill of pattern recognition? Play *Kids on Keys*, and you'll find out.

Alphabet Zoo

Alphabet Zoo is also available from Spinnaker Software. The Commodore 64 disk costs $29.95; the Commodore 64 cartridge costs $34.95.

Alphabet Zoo is a trip — a trip into a dark maze filled with colorful glowing letters. At the heart of the maze is a fox or a vase or a bottle of ink or a pair of socks (or dozens of other objects). Your goal is to guess the first letter in the object's name (like "f" for fox or "s" for socks). Then you run through the maze and chase down that letter. The letter skulks around the maze trying to elude you, but you can enter special doorways and take shortcuts through the maze. When you capture the letter, the computer plays a musical tune, you win points, and you get plopped down in a new maze with new letters and a new object.

Alphabet Zoo is very flexible. You and your child can play alone or together. You can choose to chase down capital letters, lowercase letters, or a mixture of both. Also, you can graduate to game 2 where you have to chase down entire words that match the picture in the maze's center.

There are six levels in each game. This lets your child work on different types of letters and words: easy and difficult consonants, vowels, etc., and words of anywhere from two to nine letters.

Alphabet Zoo is a valuable game for you and a child to play. It teaches all sorts of reading readiness skills, including letter recognition, letter sequence, and letter sounds. And having to chase the letters around the maze helps children develop fine motor skills that they will need when they begin writing.

All these things are terrific, but I've saved the best part for last. When you start each new trip into the alphabet maze, you get to choose your own player-creature. And the creatures are hilarious. One is a plump, pumpkin-like happy face. Another is a little, excited monster who keeps jumping up and down.

All the player-creatures are very lovable. Making them hop and bump their way around the maze hunting letters is a big part of the game's charm. And that's the secret of a good game. The game has worthwhile goals and desirable rewards. But it's also fun just playing. You and the child will still enjoy yourselves even if you never do track down one of those tricky letters.

Cosmic Life

Cosmic Life is available from Spinnaker Software. The cartridge for the Commodore 64 costs $34.95.

Cosmic Life originated long, long ago, in the mists of time, before the Apple, before the PET, and before the TRS-80.

In that long-ago time there was a math wizard named John Conway. Conway created a game called *Life*. In Conway's little world, creatures lived according to three very simple rules:

- Survival

  Every creature with two or three neighbors was happy and survived until the next generation.
- Death

  When a creature was surrounded by four or more neighbors, the creature felt overcrowded, became sad, and died. If the creature had only one neighbor or no neighbors at all, the creature became lonely and died.
- Birth

  Whenever three creatures got together and shared an empty space, they produced a new creature for the next generation.

Conway published his game of *Life* in *Scientific American* over ten years ago. But it wasn't until recently that Ken Madell, the author of *Cosmic Life*, showed Spinnaker that he could convert Conway's intellectual parlor game into a fun computer learning game for kids and adults.

The creatures in *Cosmic Life* are known as Digi-Bugs, cute little

Pac-Man-like creatures. They are born, they live, and they die according to Conway's original rules.

When you play *Cosmic Life* you begin with a barren, uninhabited planet. You pilot a joystick-controlled spaceship down to the planet and begin seeding it with Digi-Bugs.

Then prepare to be entranced. Digi-Bug colonies start popping up all over the screen. The little creatures grow, multiply, dwindle, and disappear, right before your eyes.

You can set everything in motion, then retreat to a cloud to watch the action, or you can dive your spaceship back down and continue to seed the planet's surface with new Digi-Bugs.

Pretty soon you will develop a real affection for the little creatures. You will learn which patterns help them grow and which patterns make them sad and vanish.

Something happens each Digi-Bug day. Each day lasts about four seconds. You can create a game of anywhere from 10 to 250 Digi-Bug days. At the end of each day the computer scores points based on how many Digi-Bugs are currently living on the planet. Your goal is to create settlement patterns for the Digi-Bugs that make them happy, fruitful, and fertile. But you have to keep a balance. If your Digi-Bug planet gets too crowded, the Digi-Bugs will start disappearing again.

Up for Grabs

Up for Grabs is also available from Spinnaker Software. The program costs $39.95 and comes on a cartridge for the Commodore 64.

Up for Grabs is supposedly for kids eight and up. But it is an instant swamp for kids, and maybe for adults, too. This is not to say that the program is not fun, because it is fun. But *Up for Grabs* is not an intuitively charming game like the other games above. It takes lots of practice, and you'd better read the instruction book if you want to know what's going on.

Up for Grabs is an electronic Scrabble game. A cube spins around in the center of the screen. On each of the cube's faces is a letter. The letter rotates around, in view, then disappears. When the cube face comes around the next time, a new letter has replaced the old letter.

You pick a letter by pushing the button on your joystick. An arrow appears and points, in turn, at each of the letters on the cube that are visible. When the arrow points at the letter you want, you press the joystick button again.

There are four letter boards for up to four *Up for Grabs* players.

Once you have chosen a letter, you can place it on one of the squares on your board by manipulating a row pointer and a column pointer.

When Catie and I first tried playing *Up for Grabs* without reading the directions, we got nowhere.

Later, my wife Janet and I played. Janet spent most of the first couple of games fuming and fussing at the computer. She claimed it was stealing her letters, putting them on the wrong squares on the board, and substituting other letters for the ones she'd chosen.

I had the same problem.

But then things started improving. We got better at manipulating the letters and the game boards. All of a sudden, we were hooked. We played game after game.

We kept playing. I looked at my watch. It was ten o'clock, it was a school night, and the kids were upstairs noisily dismantling their bedrooms. But Janet and I played on.

If you like Scrabble and you are a patient learner, you'll like *Up for Grabs*.

Tonight I'm going to talk to Catie. I'm going to try to persuade her to give the game a second chance. I think it's worth it.

Fraction Fever

Fraction Fever is available from Spinnaker Software. It costs $34.95 and comes on a cartridge for the Commodore 64.

This is one of the most frustrating yet most addictive games I have ever played. (Spinnaker recommends *Fraction Fever* for *people* eight and up. *Fraction Fever, Up for Grabs,* and *Cosmic Life* are the first three games in Spinnaker's Family Learning Game series.)

The game is not a swamp, it's just so darned tough!

When you enter the world of this game, you become a little person on a pogo stick. You start bouncing the pogo stick around on the bottom floor of a crazy, 20-floor building.

As you bounce the pogo stick, using your joystick, you discover boxes beneath the floor at intervals. The boxes, some filled and some empty, represent fractional quantities. Your goal is to find a group of boxes that matches the fraction hovering in the top-center part of your screen. For example, let's say the fraction is ½. You would bounce your pogo stick until you found, say, four boxes together where two boxes were full and two were empty. When you bounce onto the square with these boxes, you press the joystick button.

A neat thing happens. A *fraction elevator* springs out of the floor, picks you and your pogo stick up, and carries you to the next floor. You bounce off the elevator and begin hunting boxes to match with a new fraction that is displayed at the top of the screen.

One of the best features of this game is the pogo radar. The little radar screen shows the floor you are on and the floor above and the floor below, each in a different color.

The radar is important because you can use it to estimate where you will find the boxes to match the fraction. The boxes are like distance markers. The fraction they represent is equal to the portion of the whole floor you have traveled, measured from left to right. For example, if you are trying to find boxes representing ¾, you can locate your little pogo-stick person on the radar, then bounce him three-quarters of the way along the floor to the right.

When you find the boxes — four of them, three full; or, perhaps, eight of them, six of them full — you have three visual matches for a particular fraction. First, you have the fraction itself (¾) in the upper part of the screen. Second, you have the four boxes (three full out of a total of four). And third, you can see the little pogo stick on the radar, and it is exactly three-quarters of the way along the floor (measured from left to right).

The radar is also important because it warns you that holes in the floor are close by. If your pogo-stick person drops through the hole, he falls to the next floor below. This doesn't hurt him, and he can summon the fraction elevator to go back up by matching a new fraction to new boxes. But he can fall only ten times. After that he runs out of pogo sticks.

Where did the holes come from? The only way to get points in this game is by punching holes in the floor with your pogo stick. You get points each time you punch a hole in the floor over a set of boxes that do *not* match the fraction that is appearing on the screen.

But watch out. You have to punch and run, or you will drop through the hole you just created and fall down to the floor below.

And there's the rub. Those holes are a darned nuisance. The first few times I played *Fraction Fever* I deliberately punched lots of holes to score lots of points. But then my floors had holes everywhere, and I ended up falling down a hole before I could find the correct boxes and catch a ride upward on the fraction elevator.

So I changed my tactics. I tried to get to the topmost (twentieth) floor first. Then I planned to work my way backwards, punching holes and falling through the floor.

This tactic worked fine until the sixteenth floor. Then the boxes changed to partly filled beakers. I had to see if the current fraction matched the amount of liquid in the beakers, and then check to see if the partly filled beakers matched the portion of the floor I had traversed. By the time I went through all this estimating and guessing, my time would run out and I would have to hop off the current floor (or fall through a hole) and drop to the floor below. Then the timer would start again and I would try to match the fraction, the beaker, and the floor, and catch another ride upward on the fraction elevator.

Unfortunately, I kept timing out and falling through holes faster than I could estimate fractions. Pretty soon I was back near the bottom of the building with no more pogo sticks to bounce on.

Now I'm a veteran of *Fraction Fever*. Even so, I've never made it past the sixteenth floor, and I've never scored over 16 points.

But I'm going to keep trying. And because I'm persisting, I'm becoming a better fraction-guesser and a better pogo-stick bouncer.

I just wish that Tom Snyder, the designer of this game (along with other Spinnaker best sellers, such as *In Search of the Most Amazing Thing* and *Snooper Troops*), would have been more generous with his point allotment. After scoring thousands of points with videogames, I found it quite hard to be content with scores like 6, 11, or 3.

Also, I would have loved it if Snyder had awarded me points for guessing the correct fraction rather than for punching holes in the floor whenever I spotted an incorrect fraction (or group of boxes representing a fraction).

Last, I wish that Snyder had designed the game with several levels, including three or four below the level the game operates at now. I can live with the knowledge that I've only made it to the sixteenth floor (that's 16/20 of all the floors, or 8/10, or 4/5, or four full boxes out of a total of five). But it would have made it easier for me to get Catie and Eric past the first floor.

I've caught a terminal case of fraction fever. Now I'm anxious to pass it on to my kids.

Chapter 5

Children, Computers, and Values

Superbaby Meets the Computer

If you haven't seen it already, you should go to a library and find the March 28, 1983, issue of *Newsweek* magazine. Turn to page 62 and read the cover story, "Bringing Up Superbaby." The story is about how parents are pushing their kids to learn earlier and earlier. Kids who are only a few months old are studying art books, gazing at flash cards, doing toddler gymnastics, going to dance class, putting together puzzles, taking swimming lessons, and *learning how to compute*. In the article there's a picture of a little kid who is pounding away on the keyboard of an IBM Personal Computer.

Just a few years ago, Elizabeth Wall (a media specialist in Sarasota, Florida, and author of *The Computer Alphabet,* Avon, 1983) sat down next to one of the pioneers of personal computing. He asked her what she was up to. "Teaching elementary school kids how to use computers," she told him. He was shocked. "There's no future in teaching little kids computers," he said. "They will never get the hang of it."

Since that expert made his remark, use of computers has dribbled downward, from college to high school kids; from high school kids to middle schoolers; from middle schoolers to kids in elementary school — and beyond.

In Bruce and Diane Mitchell's Small World preschool and kindergarten, in Durham, North Carolina, four-year-olds and five-year-olds are playing educational games on Atari computers and Timex/Sinclairs. They are programming a turtle robot by tapping on the keyboard of an Atari 800.

But preschoolers and kindergartners are old. They're almost over the hill! The *Newsweek* article mentioned a school called Tiny Bytes where kids can begin computing before they've celebrated their first birthday.

Computer Literacy or Else

Some toddlers are going to be victimized by pushy parents trying to fill their offsprings' "little sponges" with computer facts even before they've learned to walk or talk. I can imagine an "enlightened" household where the parents are trying to give their three-month-old an early start on her way to a high-tech future. The baby, blithely unaware of her parents' designs, is reaching for a rubber duck. The mother pushes the duck away. "Too easy," she says. She whips out a stack of big white flash cards. "Let's practice these first, then you can see the duck on your lunch break." As the baby gazes sweetly at her mother, the mother runs through the flash cards. "RAM!" she calls out. "RAM . . R . . A . . M . . RAM! BIT! . . B . . I . . T . . BIT! CHIP! . . C . . H . . . ."

One wonders what a kid who gets computer flash cards at three months is going to be like when she gets to the ripe old age of five years, or ten, or fifteen. She may have a lot of computer facts under her belt, but how well-adjusted will she be? What will be the result of all this parental prodding?

This is not to say that computers shouldn't be introduced to kids who are still wandering around the house in dirty diapers. Because they should be!

The question is *how*.

Parents who are pushing their babies and toddlers into computer literacy are missing the point — at least as far as computer literacy is defined. We are presently in the Age of Computer Literacy. But we are quickly moving beyond it. Pretty soon it will not be productive for us to study such arcane terms as *bit, byte,* and *CPU*. We won't have to know how a computer works, just how to work a computer. We will be leaving the age of computer literacy and entering the Age of Computer Intimacy.

Take the TV or the car. These are high-tech machines that are part of almost every little kid's environment, right from birth. Do parents go around with flash cards with words like CHANNEL SELECTOR or PHOSPHOR SCREEN? Or with words like CARBURETOR or PISTON? Of course not. Nevertheless, the smallest children learn how to operate TVs, almost before they can walk. And little kids play with model cars, toy cars and trucks, all through their childhood. And when their magic birthday arrives and they can get their driving license, they quickly learn to drive and operate an automobile.

How many kids suffer from automobile anxiety or TV phobia? Very few.

Even more important, how many kids can expect to find a job when they grow up as an automobile mechanic or an expert in TV repair? Again, very few.

Yet TVs and cars are far more common than personal computers.

The point is that we have moved beyond "TV literacy" and "automobile literacy" to a new age of intimacy with both these machines. The technologies have matured. They are black boxes, idiot boxes that almost anyone can learn how to use. They're everywhere. We're comfortable with them in our garages, our living rooms and bedrooms.

This is where computers are headed, too. They've just started, but at the speed they're going, it won't take long. By the end of the 1980s, computers will be black boxes, just like cars and TVs. They will be in most people's homes. They will become so common that they will cease being an eye-catching phenomenon. In fact, they will almost be invisible. Like electric motors, they will slip into other appliances and disappear from view.

Kids who are less than one year old in 1984 will be less than six in 1990. So why are parents teaching them computer literacy terms and concepts, preparing them for a job market that exists in 1984, but will change radically even before the kids have made it through elementary school?

Parents are pushing because they are panicking. The swift pace of computer technology has them running scared.

And they are pushing their kids because of the status of having them say "floppy disk" as their first word.

What they don't realize is that they are training their kids in what will soon be an obsolete technology and, worse, an obsolete approach to technology. They are being trained to become the automobile mechanics and TV repairpersons of the twenty-first century. These are honorable professions. But is this what the parents intend?

Computer Osmosis vs. Computer Bullying

Millions of personal computers are going into people's homes. Millions and millions of little children are waking up each morning and walking or being carried past computers on their way to their bottle, their Boo Berries, or baby cereal. For them, computers are no more wondrous or rare than the floor lamp, vacuum cleaner, or telephone. They're just one of the many things that belong in their lives. They have a place, along with everything else.

This is exactly as it should be. Computers *are* a big deal to us. And our kids will see that. When we spend all night in front of a keyboard trying to debug a program or escape from the wizard's castle in an adventure game, they'll notice. If we shout and point at the new computer and say "Gee whiz!" and "Oh, gosh!" enough times, they'll notice. And if we get frustrated with the computer and begin saying unkind things to it or give it a good bop, they'll notice that, too. Whether positive or negative, our kids will pick up on the attention we give to computers and the amount of emotional involvement we have with them. Kids are very sensitive about this sort of thing.

Growing Up Together

You and I are already grown. We're big people. But computers and kids haven't stopped growing. In fact, they've just begun. Both are going to change rapidly over the next 20 years.

At the end of that 20 years, what will they be like?

We imagine that our kids will end up pretty much like us. But how about computers? When kids enter the job market in the late 1990s or early twenty-first century, what will computers be like?

According to experts, we are quickly entering a new era of personal computers. I call this era the Age of Computer Intimacy. Others call it: The Age of User Friendliness. The Age of Forgiving Systems. The Age of Easy Computing. The Age of Humanlike Machines.

As anyone who has struggled with a cranky program recorder, or with a cryptic BASIC error message, or with computer cables, plugs, and connections knows, we have not reached computer heaven yet. Far from it!

But we are moving closer. While at the West Coast Computer Faire in San Francisco, I attended a seminar on "Second Generation PC Software." It was mind-boggling.

I learned that if you have enough money, you can buy computer programs and computers that are really, truly friendly. They hold your hand. They speak English (most of the time). They help you out of tight spots. They remind you of what you are supposed to be doing when you get lost.

And, boy, are they powerful! With just one package, one electronic mouse, and 45 windows, you can figure out your income tax, send electronic mail, draw pie charts and bar charts, do word processing, and file, sort, and retrieve records. All with the same set of commands.

At present, these systems are extremely expensive. Only the folks who carry around Pierre Cardin calculators can afford them. But computers, in general, used to be this way, too. Only wealthy, technically sophisticated organizations (universities, large corporations, the government, and the military) could afford them. But computers have come a long way. Now you can buy a programmable computer for under $60. Pretty soon the price will be even lower, and the computer will be more powerful and easier to use.

The new generation of "easy" computers and "friendly" computer software is coming. And it will include machines and programs that we can all afford.

What Do We Tell Our Kids?

If we're not supposed to tell our children (and babies) about bits and bytes, then what do we tell them?

Nothing is okay. Unless they ask. Or unless you're so excited about something neat that you just feel like babbling.

Just have a computer around the house. That's enough. Treat it like you'd treat a typewriter, a telephone, or a calculator. *But let your kids touch it*. That's the best way for them to learn. For example, my four-year-old son Eric drives me crazy when he uses a computer. He has grimy, dirty fingers. He presses buttons in such a way as to make a computer act like an amnesiac. But he loves to play on computers because he is allowed to play freely. And (with quiet wincing and cringing) I let him. One of his favorite games is filling up the picture screen with graphics symbols, multicolored bars (using color keys and the reverse-video button), and random letters, numbers, and punctuation symbols.

Another of his games is to use the computer as a Gobbledygook Processor (that's "GP"). He types all sorts of strange-looking words like:

```
IXCCY##559 ISK ERIC    !!!!!  AAAAAAAAAAAAA
```

then sends them off to the computer's printer. He rips off the printer paper (in the same lavish, boisterous way he handles toilet paper) and tapes it up around the house as a sign of who knows what. Or he stuffs a wad of it in an envelope, and it becomes a letter. Or he gives it, as a gift, to me, to his mother, his sister, his kitty, or his toy robot, Denby.

A Tool or a Crutch?

Actually, there's more to computer education than this. Our responsibility as parents (and teachers) extends beyond just making computers available to our children. Much further, in fact.

When our youngest children start entering the job market, in another 15 to 20 years, *all* computers will be "easy" computers; *all* programs will be "friendly." Computers and programs will also be a lot more intelligent than they are now. There will be a tremendous temptation to let computers take over many of the thinking chores that we humans find bothersome, tiresome, boring, or too difficult. At some point, for many people, the computer will cease to be a support and start to be a crutch.

Our responsibility, as parents and teachers, is to teach our children the value of using computers in the proper way: to help them do their *own* thinking.

New Resources

A book has just been published for parents of older children (ages nine and up) who are interested in computers. I recommend the book because it is a practical guide to the technology *as it exists today*. If you want to launch yourself and your family into computing today (and you should), then you need a survival manual. The best survival manual of all is *COMPUTE!* magazine, with all its tutorials, articles for beginners, practical programming tips, and actual programs for you to copy into your machine. But if you're a parent, you should also take a look at:

Eugene Galanter, *Kids and Computers: The Parent's Microcomputer Handbook* (Perigee Books, The Putnam Publishing Group, 200 Madison Avenue, New York, NY 10016; $7.95; paperback; 7-page index; 190 pages)

Sample chapters: Microcomputers and Your Child; What Is a Microcomputer?; The Microcomputer's Parts; Programming by, for, and with Children; Running the Machine; Kids Can Write Programs; Evaluating Computer Education.

The author, Eugene Galanter, has been teaching kids about computers for several years. You can write him for additional information or to ask specific questions about kids and computers:

Eugene Galanter
The Children's Computer School
21 West 86th Street
New York, NY 10024

Looking for Good Software?

Of course you are. So you should get in touch with an organization that evaluates the newest educational software:

Educational Products Information Exchange (EPIE)
Columbia University Teachers College
P.O. Box 27
New York, NY 10027

EPIE has recently entered into an agreement with the Consumer's Union to test and evaluate hundreds of consumer-oriented and educational computer products. The results of their research, reviews, and laboratory tests are just becoming available. You can read about these results in their *MICROgram,* published monthly as part of:

The Computing Teacher
The International Council for Computers in Education (ICCE)
135 Education
University of Oregon
Eugene, OR 97403
Subscription: $16.50/year

Teachers (and interested parents) can write EPIE & Consumer's Union to obtain special publications, including the Micro Courseware PRO/FILES for K–8 Educators ($84), and the Courseware Resource List (as a bonus for ordering the PRO/FILES):

EPIE & Consumer's Union
P.O. Box 839
Watermill, NY 11976

Software, Sexism, and Other Topics

I have received lots of mail from people who read my columns in *COMPUTE!*. Jan Murphy wrote that she had been enjoying my column each month. Then she read the column on the computer friend. (See the chapters in this book on the Computer Friend.) On page 82 of the column *(COMPUTE!* August 1982), she read the following words:

Is this child a boy or a girl? The computer friend should know.

This line profoundly angered and upset Jan. Why? I'll let her tell you in her own words:

Why am I mad? I said to myself, "How refreshing it would be to have a friend who didn't care *if you were a boy or a girl. And this computer friend idea would be a great chance to do that, but if everyone makes the computer friend* know *then that chance would be wasted."*

Why *"should" the computer friend know the gender of a child? So the friend can treat the child in an "appropriate" way? How about letting the computer friend treat the child like a child instead?*

I don't go around saying all men are horrible, or get active in political things, or do many other things that people imagine "feminists" do. I reject the term feminist; I want human *liberation, freedom for both men and women to be who they are. So if my brother can't fix a car, who cares? That makes him bad at fixing cars, not "less masculine." And if I like computers, why shouldn't I? This is the kind of trap I see us all falling into when I read that the computer friend "should know" the child is a boy or a girl.*

It can't be coincidence that girls often do better in math in elementary school, yet by the time they are in the 12th grade, they can't cope with numbers. There were 3 out of 30 in my physics class, 3 girls and 27 boys. Why do you think I never learned which way to turn a screwdriver until one of my (male) friends taught me, in high school? Because girls don't get building toys for Christmas, that's why. Or telescopes, or chemistry sets, or tools. Why? There's no reason — it's just the way things have always been done, that's all, and we don't bother to cast off all the old baggage from the past when a better way of doing things turns up. (Perhaps I should say it this way: Nobody likes to give up old software.) That's *natural. But (as usual) Ursula K. LeGuin has said it before, and said it better than I ever could:*

> *To oppose something is to maintain it.*
>
> *They say here that "all roads lead to Mishnory." To be sure, if you turn your back on Mishnory and walk away from it, you are still on the Mishnory road. To oppose vulgarity is inevitably to be vulgar. You must go somewhere else; you must have another goal; then you walk another road.*
>
> —The Left Hand of Darkness
> (Estraven, in Chapter 11)

I want our children to be people *first, and doers of great deeds, and makers of great works, creators and conservers and heroes; all these things first, and then, later, when they know who they are and that they can do great things, and that the world is full of wonderful things for them to learn, when they are firmly settled with a sense of their worth, then, they can also learn to be* boys *and* girls *(if it is still necessary) or* men *and* women. *But I want us to break the circle, as LeGuin would say, and go free. That's my goal. You want kids to be able to learn and grow, too, otherwise you wouldn't write such a neat column. Right?*

So when you make your computer friend with your child (and this goes for both your children) please *be careful and think about what you might be doing (unconsciously, I know) when you tell the computer friend things about your child. You've been pretty good so far, using "she" and "her" some of the time (I've noticed, yes, and I was impressed), so keep up the good work.*

Well, you wanted input on your idea for a computer friend/pet. I don't know if this is quite what you had in mind. But a computer named after the Archmage of Roke deserves the best, and it would break my heart to see him acting like the computer in a bad Heinlein novel.

Thanks for your columns (I loved the story about Eric and your floppy disks; I'm glad you didn't lose anything) — I'm looking forward to Catie and Eric's further adventures.

A Response to Jan's Letter

Jan's letter was thoughtful and thought-provoking. It angered me and upset me. It also convinced me that I had made a serious blunder.

I spent several days thinking about what Jan said and talking it over with my wife. I've come up with a tentative response.

First, I'm glad Jan wrote. The issue she raised is vital. The more "friendly" that software becomes — the more it acts like a person — the more it will carry hidden values. The question is: What are those values? Are they fair to all people? Or are they prejudiced and unfair?

Also, this issue assumes even greater importance now that young children are beginning to use personal computers. These children's values are largely unformed or, at least, extremely malleable. Values hidden inside computer-friend and computer-tutor software might be easily transmitted to young children. Again, it's important for us, as parents and teachers, to uncover these values and make sure they are similar to our own values.

In this specific case, however, I'm afraid that I disagree with Jan over whether the computer friend should ask the child's sex. I still think it should. It should for two reasons.

First, the child's sex is a biological fact. Second, the child's sex is an important, perhaps decisive, factor in determining how other people will treat the child. For good or bad, it is too big a factor for the child, or the friend, to ignore. Only by dealing with it can the child (and the friend) overcome it.

Back to reason number one: biology. For biological reasons alone, the child's sex is a central fact of the child's life. The child identifies herself or himself, in part, based on that fact.

I'm not saying what that fact means. I'm not saying that girls and boys shouldn't be free to express their personalities. I'm only saying it is an important fact and should not be covered up, rationalized, or denied. If I were a girl or a boy and I had a computer friend, I would want *my* friend to know my sex.

Second, unlike the computer friend, the child does not live in a vacuum. It lives in the real world. And the real world is filled with people who discriminate against women and men (in different ways, of course). Prejudice is built into the laws, into people's values and opinions, into institutions, and into almost every activity of our lives.

If the friend is to become a real friend, it must learn which side of the sexual fence the child is on. Only then will it be able to relate to the challenges the child will face in trying to overcome the injurious sexual stereotypes that pressure the child into a certain kind of behavior, career, style of life, or whatever.

This issue seems a little over-dramatized, given the extremely simple computer friend we have discussed so far. But a major trend in computer software is to anthropomorphize computers and make them more lifelike, human, and friendly. Computer friends in the near future won't be toy programs to amuse preschoolers. They will be built into silicon chips and be an intimate part of our daily lives — in the office, in the school, and in the home. Therefore, the type of values our "friends" should have is a good issue to be thinking about right now.

Computer Friends for Adults

A couple of weeks after I'd received Jan's letter, I got a letter from Irwin J. Davis of Bridgewater, New Jersey. He proposed a computer friend for adults. Here is an excerpt from his interesting letter:

> *I read with interest your article about building a computer friend for a child. It did occur to me that the same concept could apply to adults. Why not build into the computer an adult personality like The Sage or Chief Mentor. The programmer could put in all his favorite sayings or aphorisms from secular or religious sources as Thoreau, Montaigne, the Bible, etc. The computer could suggest meditation exercises or relaxation techniques depending on how the person felt. In the past people would keep a journal and write sayings or thoughts of importance in them. Why not put them into a computer under certain categories and recall them for certain moods. Suggested types;*
>
> *The Sage or Philosopher*
> *The Psychologist*
> *The Swami*
> *The Man of Action*
> *The Rabbi, Minister, or Priest*
>
> *The programmer would have to know quite a bit about his character, which would be a good exercise for him or her.*

Catie's Christmas Card

Sometime between Chanukah and Christmas, my seven-year-old daughter Catie got a letter. I took it with me when I went to pick her up from school and gave it to her as we were driving to a doctor's appointment.

Catie had been talking to me about her upcoming holiday play at school when I threw the letter to her in the back seat.

I hear some ripping-paper noises, then silence.

I didn't notice the silence for the first couple of moments. But after it wore on for about a minute, I grew alarmed. Had Catie fallen out of the car? Had the letter knocked her unconscious?

I turned around to look for her in the back seat and whacked my nose against Catie's hand. She had come quietly forward between the bucket seats of our Toyota and had been perched only inches from my right ear.

In Catie's hand was the letter. And on Catie's face was an expression that, until then, I had only seen in Doris Day movies and documentaries on religious pilgrims.

Catie was beaming. Even more than beaming. Her expression was so extreme and the emotion inside her that produced it was so contagious that I felt like crying, or grinning, or both.

"What have you got there?" I asked, trying to appear casual.

"A Christmas card," Catie said.

"Why don't you read it?" I said.

Catie read the card. It was from her "secret" boyfriend at school (the boy whose name I am not permitted to mention in this column). At the bottom of the card he had signed his name. Above his name was the magic word: "Love."

Sharing What Is Special

These family events relate to the computer friend project I discuss from time to time. More specifically, they relate to the kinds of information we give the friend and the kinds of experiences and feelings we share with the friend.

I won't pretend that when Catie got home, she immediately sat down at the computer and told her computer friend about her card. She *did* tell her mother and her brother and her cat and her unicorn. But she didn't tell her computer.

But what if she had?

And what if her friend didn't know Catie was a girl? And that the person who sent Catie the card was a boy?

What if the friend didn't know about boys and girls, at all?

How much of Catie's experience could she have shared with the friend then?

Of course, she could have said that a "person" or a "child" sent her the card. Or that another "friend" sent it to her.

But that would have been leaving out the special part: that Catie is a member of one sex, and the person who sent the card is a member of the opposite sex. And that, because of the difference in their sexes, Catie likes this person in a special way. And now she knows that this person maybe likes her that way, too.

Computer Friends of the Future

I believe that one of the most powerful, important, and swiftly evolving trends in computers is to make them more like human beings. I also believe that computers, as teachers, nannies, pets, and playmates, will have a growing role in our youngest children's lives.

What information and values will these computers carry and communicate to our children?

Facts vs. Values

The issue at stake here is much larger than sexism. Computers will soon be used to transmit values of all sorts: about sex or gender, race, religion, nationality, intelligence, lifestyle, employment, education, physical and mental well-being (or handicaps), and so on. As groups with different values begin to use computers extensively, their computers will increasingly reflect their values and points of view.

How do these values relate to our children's computer friends?

No matter how hard we try, we won't be able to create a completely value-free computer friend. Nor would we want to. All of us (parents, teachers, policymakers, etc.) will want our children's computers to echo our own values, or at least our nobler values (what we preach as opposed to what we practice).

We will all (each in our own way) attempt to screen out values with which we disagree. We don't want our children's computer friend to be sexist, racist, xenophobic, or bigoted in any way.

Many of my readers felt the best way to make the friend open-minded was to leave it ignorant of such matters as race, sex, etc. *I feel the exact opposite.* Just as with human beings, we can't expect prejudice to disappear in an information-poor environment. In fact, prejudice flourishes when facts are not known or are distorted.

I believe the same is true for the friend. I believe we should carefully monitor the *values* that our children's friend acquires, stores, and transmits. But we should not censor the *facts* that it learns and communicates to our children. Instead, the friend's (and, by extension, our children's) diet of facts should be as rich and diverse as possible. Only in this way can the friend begin to understand the world as it really is and the unique niche in that world that is occupied by our children.

Chapter 6

Learning on the Computer

Revolution in the Nursery

There is a computer revolution afoot. It is not a revolution in computer hardware or software. It is a revolution in computer *learning*. Like a classic revolution, it is starting at the grassroots, with the smallest kids. But it is percolating upward through our society like an erupting volcano. All of us will feel its impact: parents and teachers, of course, but also office workers, managers, professionals, and eventually, presidents of corporations and countries.

We are just beginning to recognize the true power of the computer as a catalyst for learning in the home, classroom, and workplace. Current learning methods such as CAI (Computer-Aided Instruction), computer drills, simulations, and learning games are just the tip of the iceberg.

The computer will soon alter the way family members spend their time together. It will have a bigger impact on daily family life than the TV.

Wave of Computer Kids

Kids who use computers almost from birth were nonexistent just a few years ago. Now there are thousands of them, scattered through our society. These kids are acquiring models of personal behavior, learning, and views of the world that are profoundly different from those of kids without an early, rich experience with computers.

These kids are now the priviledged few, but in just a few years they will be the norm. What happens when this wave of kids hits the schools? Will kindergarten be ready for kids who have been working with numbers and words for years? Will first grade be ready for kids who have been conducting science experiments, programming robot "pets," creating advanced geometric designs, and role-playing in advanced simulations? Will schools be able to offer the rich computer-learning environment the kids are accustomed to at home? Or will there be only one or two computers for 30 or 60 kids? And will the computers and the school curriculum be locked into precomputer teaching philosophies and methods?

Learning on the Computer

We are at the threshold of a riot of new learning. That's what it will be, too: a riot — rampant, disjointed, and uncontrolled. Each kid will be different. No two kids will be at the same level when they enter the kindergarten. Many kids will have had a rich computer-learning environment since birth. Some will have had no computer experience.

Among those kids who have had exposure to computer learning, there will be a kaleidoscope of backgrounds. Some kids will have used only computer drills, others advanced simulations. Some kids will have had an array of computer tools — light pens, microphones and headsets, robots, musical keyboards, videodiscs, etc. Other kids will have used only a computer, a TV, and a cassette recorder.

The "home computer curriculum" will be completely hodgepodge, dictated by each family's budget, interests, and informal educational philosophy. Computer kids will know a lot, but what they know may be thin and spotty. They may have skills of sixth- and seventh-graders, they may be accomplished readers and whizzes at figures, but there may be no structure, discipline, or order to what they know. They may know advanced concepts, but they may lack basic skills.

This revolution will come, no matter what we do. And it is coming fast. It will soon spread past the homes, past the schools, and into the workplace.

Kids Teaching Adults

"User-friendly" is a buzzword connected with office-automation software. But the real "user-friendly" programs now in use are not available in the office. They are in the classroom and, especially, in the home. They abandon traditional teaching methods, but they make learning rewarding, exciting, and fun. *And they have universal appeal.* The other night, my seven-year-old daughter was playing with a *Sesame Street* program from Children's Computer Workshop. I came in to watch her. Then her three-year-old brother arrived, then her mother. Pretty soon we were all working with the program, and cheering, squabbling, laughing, and crying. Our involvement with the program — and with each other — was intense. I felt as if we were in front-row seats at an exciting basketball game. Only we weren't spectators, we were all *participants*.

The *Sesame Street* program is unusual, but it will soon be typical of a new generation of computer-learning programs that involve the whole family. And the learning will be happening at all levels.

The program will be teaching everyone. The adults will be teaching the kids. The kids will be teaching the adults. The kids will be teaching the kids, the adults teaching the adults, and everyone teaching the computer. That night was a rare, exhilarating moment of family interaction, but I believe it will become more and more commonplace as new and better software is introduced.

Taming the Revolution

This is a wonderful time to be alive. It is a wonderful time to work with children — as teachers in the classroom, and as parents in the home. It is a time that holds great dangers but also great promise.

Two important issues stand out. First, what are the new methods of learning that the computer is opening up to us? How do they compare with the old methods?

Second, how can we coordinate computer learning in both the classroom and the home? How can we tame the revolution and exploit its full potential rather than let it become disruptive and go wild, as revolutions are prone to do?

I've done a lot of exploring on my own, and now I'd like to do some exploring with you. So, climb aboard. It's good to have your company!

Albert Zap, Won't You Please Come Home?

Albert Zap woke up from a long nap and looked out the window of his house. It was a sunny day. The sky was blue. The grass was so green it glowed. Albert Zap decided it was a perfect day for a walk.

Albert walked out the front door of his house. "Tra-la-la!" he sang. He skipped through the gate of the white picket fence. "Such a perfect day for a walk," he said to himself.

Then something terrible happened. Albert had wandered only twenty yards from the front door of his house when the sky grew dark. A shadow fell on him and on the grass at his feet.

Albert looked up, expecting to see a gloomy storm cloud passing overhead. He gasped. Something was falling out of the sky. It was a giant purple letter, and it was falling right toward him. If he didn't move quickly, he, Albert Zap, was sure to get *zapped*!

Albert tried to run, but he couldn't lift his feet. He was frozen in one spot. He looked up again. The elephant-sized letter was coming closer and closer, but he couldn't move. What was he to do?

Coming to Albert's Rescue

Computing families to the rescue!

Albert needs a hero or heroine to save him. But not just any hero or heroine will do. It must be one with steady hands, nerves of crystalline graphite, and a heart that's true.

And one more thing: Albert's hero or heroine must be able to hunt across the computer keyboard and peck the letter that is about to zap him.

Do you qualify? Does your mom or dad? Or your kid sister? Then pitch in, find the letter on the keyboard that is threatening Albert Zap, and give it a quick tap.

If you find the letter before it finds Albert, the letter will crumble like a peanut butter cookie you've just sat on.

You're Not Home Yet, Albert Zap

When the letter vanishes, Albert breathes a sigh of relief. He has decided that he has had enough walking for today. All he wants to do is go home. He takes a giant step in the direction of his house.

But what is this? Another letter, this time a giant green P, is heading right for him. And, as Albert stares at the sky, other letters follow the P: a plump yellow W, a pair of brown Q's, and a hefty green Z.

Albert feels like Chicken Little. The sky is falling. All he wants is to go home. But he can't move on his own.

Albert looks your way. You are his only hope. Can you find the letters on the keyboard and tap them, one at a time, before they reach Albert?

Finding a Champion

Albert is not picky about his hero or heroine. He might be a four-year-old with a quick little finger. Or a 68-year-old who won't desert Albert in his hour of need. He'll stay by the keyboard hunting and pecking until he gets Albert home.

But one thing is certain: Albert needs a true champion, someone who will stick with him even when the going gets tough.

And it does get tough.

Getting Albert home looks deceptively easy. After all, the letters are floating down so slowly, lazily dropping out of the sky. This game is a snap. You merrily tap the keys, and Albert jogs along toward the safety of his house.

Then disaster strikes. A giant Y falls toward Albert. You see the Y and try to type it, but you miss and strike a T instead.

Albert notices your keyboarding blooper and despairs. He loses all hope. And he loses his mind. He does the opposite of what he is supposed to do. He turns around and takes a giant step away from his home.

Albert's foolish behavior rattles you. Hastily you type another letter. You aim for a B but instead type a V. Albert takes another giant step away from home. You see more letters and make more mistakes. Now Albert is jogging in the wrong direction. The distance between him and his home is steadily widening.

Safe at Last!

Don't give up hope. Albert isn't lost yet. A true champion will keep typing. She may hit some wrong letters, but most of the letters will be right. Each time she types the wrong letter, Albert scurries away from his home. But each time she taps the right letter, Albert turns around and runs in the right direction.

And eventually Albert makes it home.

But if he doesn't, that's all right, too. Albert isn't hurt — except for a mild headache from being beaned with a blimp-sized letter. And with just the touch of a button, you can start the game over again and give him another chance. Once again, Albert is back outside his house going for a walk, and a new group of overweight letters is falling out of the sky. You can keep coming to Albert's rescue until you finally bring him home.

Unfortunately, Albert Zap does not have a lot of sense. As soon as you bring him home, instead of thanking you he heads back out his front door for another walk. Seconds later, more letters are raining down out of the sky.

It seems Albert Zap likes to live dangerously.

QuickFinger

Albert Zap is the hero of a typing game called AlphaZap, put out by Quick Brown Fox, Inc. — the people who make the *Quick Brown Fox* word processor for the VIC-20 and the Commodore 64.

AlphaZap runs on the Commodore 64. It is part of a trio of typing games, collectively known as *QuickFinger*. The *QuickFinger* package comes on disk and costs under $40. It should already be in the stores, but if you can't find it, you should contact Quick Brown Fox directly:

Quick Brown Fox
536 Broadway
New York, NY 10012
(212) 925-8290

Tackling the Terrible Keyboard

Which part of a computer terrifies people the most? Is it the alien-looking floppy disk drive? The blank, empty-looking display screen? The snakelike cables?

Nope, it's the keyboard. The keyboard is like a wall that most adults and kids have to climb over. And most of them don't.

For little kids the keyboard is a jungle, stuffed with too many keys in too small a space. Kids' eye-scanning and finger-tapping motor skills often aren't up to tracking down and pressing that unique key that will make a program run or a game reward them. Computer keys often leave small children with tears, anger, and frustration.

The situation isn't much better for adults. Most adults don't know how to touch-type. Most adults have never heard of QWERTY. They have learned their alphabet from A through Z, then they look at a computer keyboard and find the letters all mixed up.

And if the jumbled letters weren't enough to stop them, all those extra keys on a computer keyboard will. How might a fearful adult feel about the keys with names like BREAK, STOP, CONTROL, END, RETURN, and ESCAPE? For an anxious adult, even a "friendly" computer key like HELP can take on an ominous, sinister meaning.

Taking the First Step

Yet what would computers be without their keyboards? There are more and more alternatives to the keyboard. New devices for interacting with the computer are now appearing on the market, including mice, touch pads, and light pens. But the computer keyboard is still the primary way most people tell the computer what they want it to do.

We have to find ways to help people cope with the computer keyboard. One way is the typing program.

Typing programs, like *Typing Tutor* by IBM, *Type Attack* by Sirius, and *MasterType* by Scarborough, are a godsend to the average family member who is interested in learning how to use a computer but who is put off by the computer keyboard.

QuickFinger, from Quick Brown Fox, is, in my opinion, one of the best of these typing programs. It is for a very popular machine (the Commodore 64). It makes learning the computer keys easy and fun rather than boring and tedious. It has a trio of games, so you can switch back and forth. When you get tired of playing one game, you can switch to another game and still be practicing and improving your keyboarding skills. And the games are *deep*.

Deep Typing

Tripp Hawkins of Electronic Arts has called for a new generation of microcomputer programs for the average person. To be for everyone, a program must be hot (appeal to the senses), simple, and deep.

What Hawkins means by deep is that a program must be suitable for users at a variety of skill levels *or* for a single user who is acquiring new skills and becoming more sophisticated. The program must be able to teach a person at one level, then it must be prepared to continue teaching the same person at even higher levels.

The *QuickFinger* programs are deep. They are suitable for little children with tiny fingers or high-level executives with meaty paws or long painted fingernails. They are for hunt-and-peck or swift touch typists. They can be your first introduction to the computer keyboard or your refresher course, even if you already type 50 words a minute or more.

All the *QuickFinger* programs operate at multiple skill levels. You can enter a program at level one or level sixteen. Or you can let the program lead you automatically through the levels as your skill increases. Your progress is slow but steady. If you are having trouble mastering a certain level, you can hang around until you feel comfortable and ready to try something more challenging.

As you progress, the computer keeps score. You can save your scorecard on disk and recall it easily for later sessions on any of the three programs.

An Extra Bonus

I have focused only on the AlphaZap program, but the other two programs are lots of fun, too. Keyboard Pacer shows a picture of the computer keyboard on the picture screen. One at a time, the letters on the buttons turn black. When you find that letter and press it, a new letter turns black.

This is a good program for even the tiniest child or the most fearful adult. You might be able to find only one key in two minutes, but if you get it right, you are rewarded.

The last program is called Link the Letters. This time you see a grid of fenced-in giant letters, like a screen-sized crossword puzzle. You have to link the letters into words. When you make up a real word, you get points and are rewarded with a charming sound like two crystal champagne glasses clinking together in a toast.

As a parent, I especially like the Link the Letters program. It is a good spelling game for my children. It helps develop their eye-tracking skills (up, down, to the left, and to the right). And it teaches them some of the structure of the English language, including common word beginnings, two-consonant blends, and word endings. It makes our language playful. And it helps a child learn how to organize words and recognize patterns.

And Now Back to Albert

The best thing about *QuickFinger* is that when you get tired of linking letters and pacing yourself on the keyboard, you can go back and rescue Albert Zap again. He's a nice guy, really. But he doesn't have enough sense to come in out of the rain.

Computing to Read

When Catie was four years old, lots of children used to come to our house to play with our computers. Many of the children were older than Catie, and their favorite programs were the graphics-and-text adventure games.

Older children (8 to 14) used to spend hours at our house playing Epyx's *Crush, Crumble and Chomp! The Movie Monster Game,* Sirius Software's *Copts and Robbers,* and On-Line Systems' *Cranston Manor.*

The games are a lot like the "choose-your-own-adventure" books from Bantam, TSR Hobbies, Pocket Books/Archway, and other publishers. They are electronic "interactive novels," with the child playing the lead role — as hero, heroine, villain, or monster.

In the computer's story world, the child is important; what she does matters. When she makes a decision, it changes the whole course of the story. She enters a world where she alone is at the center of the stage.

In addition, the child gets to choose the pace of the story. When she boots up the disk (most of these story games appear on disk or cartridge, since they are based on very long programs), a picture appears on the screen. This is the first frame in the electronic picture book. Underneath the picture are a couple of brief sentences and a question. The statements might say something like "You are in a purple maze. All the doors are locked." The question might ask, "What do you do now?"

The child must decide what to do. But she isn't rushed. She can proceed through the story world inside the computer at her own pace. Or she can leave that world and walk into the kitchen for a glass of apple juice or a cookie. She can talk to a friend on the telephone. When she is ready to reenter the story world, she can come back to the computer and slide back into the story.

Power Words

Catie spent hours watching the older children play the story games. At first she was happy to stand nearby and watch them play. Then she got bored just watching. She began climbing up on the kids' laps and begging to push some of the buttons.

She wanted, at first, to push the buttons the other children had chosen. Later on, this was not enough; she wanted to choose the buttons herself. Eventually, she made herself such a nuisance that the others stopped playing with her. For a brief time my wife Janet and I had to ban her from the computer room because she was so disruptive.

That's when Catie began to play the games on her own.

The older kids were so relieved that they helped Catie by writing down lots of the key words she would need to know to play the games. They wrote these *power words* down on little scraps of pink and blue paper, and Catie taped them to the side of the monitor and across the top of the computer.

As Catie grew more adept at playing the games, the number of scraps of paper taped to the computer grew, until the front of the display screen began to look like a bulletin board. It got so hard to see the computer screen that the older kids had to squeeze the words onto fewer paper scraps.

The power words Catie learned enabled her to answer the computer's questions, avoid the monsters lurking in the computer's mazes, and survive longer inside the computer's game world. They included direction words like up, down, left, and right; and the points on the compass — North, South, East, and West. She learned simple verbs like go, stop, run, jump, hop, and climb. And she learned lots and lots of nouns, like candle, flashlight, food, box, treasure, window, gate, and house.

The words were important to Catie because using them gave her power. They let Catie make choices, go where she wanted to go, and do what she wanted to do.

And they had meaning within the context of her adventure inside the computer. But they were not words that appeared on a page, or even on the display screen. They were words on paper scraps outside the computer, in Catie's world. Catie had to type the words herself and enter them into the computer's world. They were keys on a key ring that Catie herself carried. At first, Catie wasn't sure which key opened which door, but she quickly learned to match the doors with the keys. When a key worked, its effects were immediate and dramatic.

Catie could control the microworld she entered by using the right power words. But the world was not made up entirely of words. It was really an electronic "picture book" world, half words and half pictures. The words reinforced the pictures, and vice versa.

Since Catie saw the older children playing the adventure games, she didn't think of the pictures as being part of a picture book — something for younger children. Catie saw the adventure games as something only big kids could master, and she wanted to be like the big kids.

Catie had the opportunity to act like the big kids because the challenge was not too great. She had to simultaneously develop her logical, puzzle-solving, and memory skills, along with her reading and writing skills, as she played the games. But Catie could follow the pathway to these skills by taking small, child-sized steps.

Catie could play the games at her own pace — with no one looking over her shoulder or telling her what to do. This meant that she had to live with the consequences of her actions. She got to experience a certain independence inside the computer without a watchful adult hovering over her.

When Catie turned on the computer, the world inside became Catie's world. She was the only human there.

A Closet Reader

One afternoon, after Catie had been playing the adventure games by herself for several weeks, I entered the computer room and noticed that all the paper scraps had disappeared from the picture screen of Catie's game computer. Catie was playing one of the games (*Mystery House* by On-Line Systems), but she was flying solo, without any help from the power words or older children.

I was going to say something, but Catie distracted me by telling me to watch her whiz through the multicolored *Copts and Robbers* mazes, capture four jeweled rings and a vase, and take them to the Vault room so she could win the game. I remember being amazed at her manual dexterity as she pushed the four arrow buttons on the computer keyboard, and I totally forgot to ask her about the paper scraps.

Catie entered kindergarten that fall and reports from her teacher about her reading began filtering home. One day my wife was talking to Catie's teacher and learned that Catie was, perhaps, the best reader in the class. She was reading books intended for children in second, third, and fourth grade.

The teacher asked us how we had taught Catie to read. My wife said that we had never taught Catie to read. If Catie had learned, she must have taught herself.

That night my wife asked Catie how and when she had learned to read. Catie said that she had known how to read all during the previous summer but that she had been too shy to tell us.

My wife and I were shocked. We had been reading picture books to Catie all summer, while secretly Catie had been reading books written for much older children. We had a daughter who was a "closet" reader.

Once Catie came out of the closet about her reading, we couldn't hold her back. That fall and winter she had her nose in a book all the time. She began bringing home six or seven Nancy Drew books a week. She read her way through Beverly Cleary, Judy Blume, Paula Danziger, Mary Norton, C.S. Lewis, and Katherine Paterson. She read all of Donald Sobol's *Encyclopedia Brown* books and Bantam's *Choose Your Own Adventure* books. Then she returned to fairy tales and began reading her way through the Brothers Grimm.

Books and Computers

My wife and I are not certain how Catie learned to read, but we are sure that the computer played a big part.

Of course, the computer was not the only factor. Janet and I both love books, and we have read books to Catie from the time she was four months old. Before she learned to read, Catie already had a library of dozens of picture books. I write books, and both Janet and I collect children's books. We are compulsive readers, and now Catie is, too.

But as far as we know, Catie never did any reading on her own until after she had begun playing the computer adventure games. So I feel that those games were the key factor in turning her from an almost-reader into a reader. She was probably on the verge of reading, and the games gave her the boost she needed to get started.

Reading: Pain Before Pleasure

We book lovers sometimes think that reading has always been an effortless and pleasurable experience. This just isn't true. It wasn't true for us, and it's not true for our children. For beginning readers, the process of decoding thousands of strange-looking letters, punctuation symbols, and words on a printed page approaches pure agony. The decoding process is slow, painful, and often extremely frustrating.

This is where computers can play a valuable role. They can break reading skills into small, manageable pieces. They can turn each small step a child takes into a mastery, an accomplishment, or a reward. And they can bring the whole process of learning to read more under the child's control and make it more personally meaningful to the child.

In an adventure game the child is in supreme control. Each time he types in the right power word and advances to a new screen, he feels a sense of accomplishment and mastery. And the stories have meaning because he himself is the central actor in the computer's story world.

Electronic Fairy Tales

When Catie learned to read, the only adventure games were for older children or adults. Now there are adventure games for younger children. These include *Gertrude's Puzzles* and *Gertrude's Secrets* from The Learning Company; *In Search of the Most Amazing Thing, Trains,* and *Snooper Troops* from Spinnaker; *Dragon's Keep* from Sierra On-Line; and the many voice-enhanced adventure games and stories from PDI (Program Design, Inc., in Greenwich, Connecticut).

Islands of Learning

I am afraid that homes and schools are swiftly becoming islands of computer learning. I am afraid that unless something is done to coordinate this learning, the tremendous educational potential of personal computers might never be realized.

I've been speaking on this subject at several educational conferences, including the National Educational Computing Conference in Baltimore, the Florida Instructional Computing Conference in Tampa, and the Hollins College conference on "Computers in the Schools."

In Tampa, my speech was entitled "Linking Computers in the Classroom and the Home." After the speech, one of the people in the audience came up and introduced himself as Kenneth Komoski, educational director of the Educational Products Information Exchange (EPIE).

Ken and I sat down and chatted. We discovered that each of us independently had decided that the critical area in educational computing was neither the classroom nor the home. Instead it was the *connection* or the *gap* between the two.

To show how similar our thinking was, here's a quote from one of Ken's recent articles:

> It's important for parents to work cooperatively with educators and community leaders to make decisions about computers and educational software.... At stake is not only the quality of computer instruction in the schools, but also the full educational potential of microcomputers for your child, your family and your community. If parents and schools do not work together, much of this potential may be lost.
>
> Kenneth Komoski, *Network* (3/83)

Ken told me about the work he had been doing. He mentioned that his organization, EPIE, had recently joined forces with Consumer's Union to evaluate educational computing products. Evaluations are published in a monthly *MICROgram*. (Subscriptions to *MICROgram* are available from EPIE-Consumer's Union, P.O. Box 839, Watermill, NY 11976.)

Millions of Programs

Hundreds of new educational software packages are appearing each month. Families and schools are buying these packages by the *millions*. In 1982, 1.4 million educational software packages were sold.

Families are buying the software and schools are buying the software, but there is little or no coordination between the two. This is the source of Ken's and my concern.

The Miracle Machine

This lack of coordination creates a real problem. The biggest impact of computers will probably be on kids using computers at home. Yet families have the least experience in setting up a formal curriculum of instruction.

Many parents buying computers are hoping that the educational programs that their kids use will teach the kids without any parental guidance. Or that the programs at least won't do any harm.

But are these parents placing too much faith in the computer?

Ever since they were first invented, computers have been seen as a miracle solution to all sorts of problems. But people have learned the hard way that computers do nothing on their own. If computers are given the right instructions, they can help. But people still do most of the work.

The same is true for computer learning. The computer cannot teach a child on its own. Parents and teachers still need to do most of the work. They need to create an environment and a curriculum fitted to the needs and age level of each child. Then the computer can be a valuable assistant.

The Impact of Computer Learning

We have another problem. Computer learning at home is likely to be unguided and uncontrolled. Yet unless computer learning at school and at home is better coordinated, the impact of computer learning will be much greater at home.

Look at the ratio of kids to computers at home and in the schools. At home the ratio is great; at school it is miserable. This is not a nice thing to admit, but it's true. One reader recently wrote me that at her school the ratio was 400 kids to only one computer. "But," she admitted, "I suppose that's better than no computers at all."

At home the ratio is much smaller, of course. Either it's nonexistent because the family doesn't want or can't afford a computer, or it's something like two or three kids per computer.

And when you think about how much exposure each child will get to educational software at home, the situation looks even worse.

Learning on the Computer

Software prices are beginning to drop rapidly. Quality educational software will soon be available at affordable prices (from $5 to $30 a package). Yet at school the software must run on the same computer that is in demand by (perhaps) hundreds of schoolchildren. This makes it unlikely that a child using a computer at school will be able to sample and fully experience the range of software that his or her school can afford.

At home, on the other hand, a child's family might buy two or three dozen packages, and the child will have the opportunity to fully experience *all of them.*

The lesson here is that the *impact* of computers will be much greater on kids learning at home.

A Riot of Computer Learning

I have discussed educational computing in the classroom and in the home. I've written about the "riot" of learning that will soon be taking place in homes that have computers. I called it a *riot* because computer learning at home will be spontaneous, uncontrolled, and perhaps, destructive.

Parents have always depended on teachers to provide formal instruction for their children, especially in basics such as arithmetic, writing, and reading. Now there are dozens of math drill programs, typing tutors, spelling instructors, and reading games.

Parents are buying these programs and turning the computer into a vehicle to teach these skills at home. But in most cases the computer tutor will not have a parent looking over its shoulder. The tutor will be teaching the child, but its lessons won't be integrated into a balanced curriculum especially suited for the child.

And this is just the beginning. The computer's effect on home learning will soon increase dramatically. At the end of the next twelve months there will be software on the market to teach every conceivable subject or skill to kids of almost every age. There will be geography programs, programs to teach chemistry, astronomy, art, dancing, songwriting, juggling, current events, and world religions.

Parents will buy this software, but kids will be expected to use it on their own.

The Cereal-Box Strategy

Schools and families will buy millions of educational programs. Schools will buy programs only after they have evaluated the programs' pedagogical soundness and relevance to teachers' existing curricula.

How will parents choose among the hundreds of educational programs on the market? They will buy programs the way they buy everything else. They will choose a particular package because:

- Their child responds to an ad on the back of a box of cereal by crying, "I want that program, Mommy!" And how can the mommy refuse? After all, the program is educational. And it's a lot better than getting the kid the laser blaster that was advertised on the box of cereal the family finished last week.
- Software will soon be on sale at special software stores and at almost every sales outlet imaginable, including local convenience marts, department stores, bookstores, drugstores, beauty salons, hardware stores, and supermarkets. It will be sold alongside how-to manuals, mass-market books, and other inexpensive consumer items. It will become as cheap and as convenient to buy as a detective novel, a romance novel, or a carton of milk.
- Software will soon be packaged as glossily as cosmetics and rock records. Ads on TV, on the radio, and in stores will make buying software as appealing as buying a new kind of lipstick, deodorant, sports car, or diet soda. It will look like an offer that can't be refused.
- Buying software will help people keep up with their neighbors who are also buying software for their kids. After all, a responsible parent can't sit still and watch her kids slip behind. And even the President of the United States admits they're not getting the education they need at school. The only way to get the kids educated is to give them an early start on learning at home. And the only way that can be managed is with a computer — and lots and lots of software.

Help from Software Publishers

Can't parents depend on educational publishers and computer companies to advise them in making their purchases?

Not likely. Educational publishers and computer companies will likely have little or no interest in coordinating the software that children use at home and the software they use in school. Nor will they be concerned with how their software fits into a child's development and into his or her learning experiences.

That's why the average home's computer learning environment will be riotous. Most parents won't have the training or the time to properly select educational software. Nor will they have the skill to build that software into a comprehensive "Home Learning Program" for their children.

Schools have this training and skill, but they are not currently part of the home software selection and evaluation process.

Islands of Learning

I see two islands of learning that will soon be created by computers. One island will be the school, the other the home.

In the school, computer learning will be structured, well organized, well thought out, properly sequenced, comprehensive, and pedagogically sound. Unfortunately, it will also be minimal, since each kid will have an absurdly small amount of time using any particular package. The effect of computer learning at school will be diluted by the huge number of kids trying to share a few computers.

In the home, computer learning will have a great impact because kids will have the opportunity to spend a great amount of time on the computer and richly experience each program. Also, there is great potential for home learning to be exceptionally creative, open-ended, informal, self-motivated, and self-guided.

Unfortunately, this potential probably won't be realized. Children's home computer learning will be blunted by a lack of any plan; by lack of coordination with a child's physical, emotional, social, and intellectual development; by a lack of noncomputer learning activities that support the computer instruction; by improper sequencing of materials and concepts; and by the gaps in learning caused by a family's arbitrary purchase of software packages.

Looking Ahead

If present trends continue, and schools and homes become islands of computer learning, what will be the result? What will be the effect of computer learning on our children? What kinds of kids will we be turning loose in the public schools? How frustrated will computer-literate kids become when they don't find the same computer resources at school that are available at home? Will non-mainstream kids be left behind? And if so, how will they ever catch up? (They won't catch up after they leave school. In the future, the average workplace — factory or office — will be even more computerized than the average home.)

If present trends continue, there is the possibility that the effect of computer learning at school will be minimal. It is likely that the effect of computer learning at home will be profound, yet it may also be profoundly destructive. It is also possible that the enormous potential of computer learning at home and at school might never be realized.

Can we permit this? Are there any alternatives?

Building Bridges

There are alternatives. We can link computing at home and computing at school. We can build bridges between these islands of learning. The bridges will permit a two-way flow of expertise and resources. The sharing of computers, software, and learning strategies can make it possible to realize the computer's great potential as a learning tool for children and their entire family.

Ken Komoski's plan for building these bridges is similar to my own. Ken (in his *MICROgram* and elsewhere) stresses "concrete programs and policies." I emphasize communication. Both elements are important. What we need, at minimum, is:

1. *Community-wide training*. Teachers, parents, and children need to attend common training sessions to learn how to use computers and how to evaluate, select, and get the most out of educational software.

2. *Community-wide access*. Parent-teacher organizations should set up *computer cooperatives* to evaluate computer software and hardware sold by local vendors. Discounts (Ken's idea) should be offered to parents who buy the computers and software recommended by the cooperative. Low-income families in the community should get special deals. Computers and software should be available for families to lease, lease-purchase, buy at a discount, or check out from a co-op library.

3. *Communication*. The co-op should hold regular public meetings and publish a monthly newsletter to make all the parents in the community aware of the diverse aspects of educational computing, including:

- Past, present, and planned applications of computers in the school (by teacher, subject, and grade).
- Educational applications of computers in local homes (by manufacturer, name of software, subject, age of children).
- Recommendation of the best new educational software and hardware; results of evaluations and tests performed at local schools and by local parents.
- Opportunities for action: community computer fairs, computer flea markets and yard sales, parents' brag nights, involvement in software and hardware evaluation committees, donation of used hardware and software to the co-op library, etc.

4. *Sharing*. The co-op needs to set up libraries of hardware and software.

Families will be buying much more hardware and software than schools. *And they will be going through it much faster*. The co-op should get families to donate, loan, or sell (at a discount) the old software and hardware that their kids have outgrown. This strategy will create an enormous reservoir of materials that can be used in the classroom and by families who otherwise would have no access to them.

In addition, this strategy will create a supply depot of computer parts, including disk drives, cables, monitors, computer circuit cards, wires, plugs, and connectors. Since computer models change so rapidly, this depot could become an invaluable resource for the schools and the community at large. Having backup computer supplies will make it possible to keep computers up and running for longer periods of time. It will make it possible to put more students on the computers and allow the students to use a wider variety of software packages.

Computer Show and Tell

In the last article, I wrote about educational computing at home and at school as isolated "islands" of computer learning. I expressed the fear that unless bridges were built between these islands, much of the computer's educational potential would never be realized.

In the article, I suggested some home-school bridges that Kenneth Komoski (educational director of the Educational Products Information Exchange — EPIE) and I had come up with, including:

- Community-wide training of parents, teachers, and children.
- Community-wide computer cooperatives.
- Communication — A parents and teachers computer association could be formed.
- Opportunities for action — The community could organize computer fairs, computer flea markets, and "brag nights" to show what the kids are doing with computers at home and at school.
- Sharing — The community could begin collecting old computers and software and set up a "computer library" (perhaps as a section of the public or school library).

Starting Simple

The program to link home and school computing is extremely ambitious. It is not something that can be implemented overnight. It is a good idea to start simple with one or two bridge-building activities, then add new activities gradually. I have found this out from personal experience.

In my hometown, Roanoke, Virginia, I am trying to put some of these ideas into practice. In the last few weeks I have learned that building computer bridges between home and school is a major undertaking. All we have set up, so far, is a swaying, rickety footbridge made of popsicle sticks. But it's a start.

Learning on the Computer

A Warm Reception

I have a five-year-old son (Eric) in a local kindergarten and an eight-year-old daughter (Catie) in third grade.

I began my bridge-building project by calling Catie's teacher, Mrs. Albertson, and volunteering to loan the school an extra computer we had sitting around the house.

I was nervous about calling Mrs. Albertson and offering her the computer. I was afraid that she might not want a computer in her class. I was worried that she would think I was an uppity parent bent on interfering with her teaching.

I was wrong.

"What a terrific idea!" Mrs. Albertson said when she heard my proposal. "We'd love to have a computer in the room. When can the computer come for a visit?"

I told Mrs. Albertson that we didn't have a TV set or a monitor for the computer. She would have to scavenge one somewhere. Also, I told her that the class would need a table for the computer and a six-foot-square space in the room next to an electrical outlet. Mrs. Albertson said she'd talk to the lower-school principal and see what she could do.

The Project Grows

Two weeks later, Mrs. Albertson called and told me that she had talked with the lower-school principal, the headmaster of the whole school, and the head of student government. Everyone had gone looking for funds and had put together enough money to enable Mrs. Albertson to buy a new 20-inch color TV for the computer.

After hearing about the TV, I didn't even ask about the table, the space, and the electric outlet. I was sure that they, too, had been been taken care of. When I visited the classroom a week later, I found they had.

Mrs. Albertson said that everyone at the school was excited about the project because they hoped that the computer could become a resource for the entire third grade, and secondarily, for the whole lower school. It was to be the first computer for kindergarten through grade three.

Enlisting the Local Computer Store

I was so encouraged by the school's response that I drove over to the local computer store and proposed that they get involved, too. I showed them my "Islands Of Learning" article in *COMPUTE!*, and I asked them what they would like to contribute to our bridge-building project.

The computer store owners' response was amazing. They said they would be happy to donate two disk-based computers to the school for a trial, two-month period. They also offered a discount on all computers purchased by parents if the school handled the purchases.

I volunteered to act as educational software consultant to the store and to tell the store owners about the most popular programs that we used over at the school. We would test the programs in school, then let the store know which ones were best.

Where Should the Computers Go?

I spent the next few nights on the phone with Mrs. Albertson and with Eric's two teachers, Mrs. Paitsell and Mrs. Carling.

I proposed that the second computer go into Eric's kindergarten class. That would make computers available in the kindergarten and third grades. The first and second graders could try out the programs that would be running on the third-grade computer. Their teachers could also take them to the kindergarten and let them try the programs for younger children that would be running on the kindergarten computer.

I liked this approach because we could experiment with using the computer at two distinct development levels. It would be interesting to see what programs would work best with the different age groups.

A Sneak Preview

Catie and Eric's teachers felt that I should bring a computer to the school for a visit before we permanently installed the computers in the classrooms. Mrs. Albertson had a table, a space, an electrical outlet, and a big color TV, so we used her classroom.

I selected Catie and Eric's best educational programs to show off at school. I drove to the computer store and picked up a computer like the two that would be donated to the school. Then I drove to the school.

Foiled by Murphy's Law

I got to the school half an hour early so that I would have plenty of time to set up the computer. I lugged the computer into the classroom and started plugging in cables and cords. When I was done I turned on the computer and the TV.

Nothing happened. The TV screen was filled with static.

I fiddled with the channel selector. I checked all the connections. I took everything apart and plugged it back in.

Still nothing.

I turned around to face the class. I was going to tell the kids about the finicky computers and Murphy's law. At the rear of the room I spotted about eight adults. While my back had been turned, the school principal and several teachers had slipped into the room for the demonstration. Instead of a demonstration all they got to see was me fussing and fuming at the dumb computer.

I was so embarassed. There I was, a computer expert, and I couldn't even get a picture on the display screen.

I was afraid to look at my two kids' faces. I knew what they must have been thinking: If daddy's going to humiliate us this way in front of our teachers and friends, it looks like it's time to put him up for adoption.

Learning on the Computer

Culture Shock

After a desperate phone call and a whirlwind trip back to the computer store for extra parts, I finally got the computer to work. Once it decided to work, the computer didn't embarass me any further. It behaved itself the rest of the afternoon.

I finally relaxed. I popped disks into the disk drive and started showing off some of Catie and Eric's favorite programs — *Delta Drawing* (from Spinnaker), *KoalaPainter* on the KoalaPad (from Koala Technologies), *Rocky's Boots* (from The Learning Company), *Early Music* (from Counterpoint Software), *Math Maze* (from DesignWare), and *Bank Street Writer* (from Scholastic and Brøderbund).

I put the disks into the computer, and Catie and Eric demonstrated the programs.

We whisked through the programs at high speed. I was hoping to show about 20 programs in two hours. The kids in the class were "oohing" and "aaahing" appreciatively. Everything was going smoothly.

Then I ran into a brick wall.

"Those programs are all very impressive," a teacher called from the back of the room (the *very* back of the room). "But they are obviously intended for older children. Don't you have anything for the children in this room and for our kindergartners?"

Whoops ....

I didn't know how to answer her. I wanted to be flip and say, "Dear lady, both my children use these programs without any supervision. Eric has been using some of these programs since he was only two years old."

I wanted to say that, but I didn't. It had begun to dawn on me that the computer programs I was demonstrating may have looked like fun to me and the kids, but to the teachers they looked like a cram course in calculus or electrical engineering.

The teacher who had spoken walked up to the front of the room. She pointed to the display screen. The menu to *KoalaPainter* was on the screen. "There must be dozens of different options on this screen," she said. "How can we teach our kindergartners to operate a program that is this complex?

I explained to the teacher that little kids didn't think the program was complex. Eric, for example, pretended that the menu boxes were "doors." He opened a door just by pointing to it on the KoalaPad. Then he went through the doors into different "drawing" worlds where he made multicolored rubber bands, grew circles and squares, and drew shapes and pictures.

"Show her, Eric," I said. I picked him up and plopped him down in front of the computer. Eric showed her.

The teacher was impressed. "He can do all those things because you taught him," she said. "You're a computer expert. But you won't be in my classroom with me and my kids. Who's going to teach me? And how am I going to teach the kids?"

First Things First

At that moment everything became clear to me. I realized that, in my idealistic fervor, I was rushing in the wrong direction. I was trying to create new educational structures, but I was forgetting the basics. The first item on my agenda wasn't bridge building, it was *teacher training*. It would be pointless to stick computers in Catie and Eric's classrooms unless their teachers knew how to operate them and were comfortable with them.

What the teacher had said was true. The kids couldn't learn on the computers unless she taught them. And before she could teach them, somebody had to teach her.

That somebody was me.

The Prime Mover

Before I took the computer to my kids' school, I had thought that I was going to act as liaison between two ongoing computer learning centers. I saw the home as one learning center and the school as the other. The way I saw it, my job was to get the two centers communicating, sharing, and trading information and resources.

After my experience in the classroom with the kids and the teachers, I realized that, for a while, my job would be much more limited. Before I could coordinate the activities of the two learning centers, I would have to *create* them.

I realize now that I'll have to spend a considerable amount of time with the teachers to get them started using computers in the classroom. And I'll probably have to work with the parents to get them started using computers to help their children learn at home.

Before I begin building the bridge between the two islands of learning, I'm going to have to build the foundations.

Show and Tell at Home and at School

I've started inviting teachers from my children's school over to our house on evenings and weekends. We are conducting an informal teacher-training workshop, and we are screening the software that we plan to use in the classroom.

I'm learning a lot.

My next goal is to create a newsletter that the kids can take home to their parents. I hope that there are a lot of parents out there who know something about computers and who'll read the newsletter and get enthusiastic about my bridge-building plans.

I can use their help. They can work with the teachers and help train them on the computers. They can bring their computers to school for show and tell. They can share their software with the school.

Once the teachers are trained and we have a nucleus of committed parents, we can think about organizing a parents and teachers computer association.

For the moment, though, I've got my hands full training Eric's teacher on the *KoalaPainter* program. Like the other teacher, she is boggled by the screen menu with all its boxes.

Eric is helping me train his teacher. He is very understanding and very patient. Two nights ago, during a session, he pointed at the screen with the *KoalaPainter* menu. "These are doors into the computer," he told his teacher. "Which door do you want to open first?"

New Standards for Home Learning

Recently I read an article by a leading educator in which he called for standards for educational software for the home. "Who must design these standards?" he asked rhetorically. "*We* must design the standards," he answered emphatically, "because *we* are the experts."

But this is not necessarily true. Expertise in using computers in the schools may not carry over into the home.

Why? First, learning at home is not the same thing as learning at school.

Second, we are a long way from realizing the potential of learning using a computer. I have a strong conviction that there are whole realms of computer learning that we have yet to explore. If we were to establish a single set of standards right now, we would stifle software companies' ability to lead us into these new realms.

The Wild, Wild West

Computer learning, especially in an unstructured environment like the home, is a vast, unexplored terrain. It would be foolish to try to define and map this terrain even before we have explored it. We would end up roping off a small part of the territory to confine ourselves in. The rest of the territory, beautiful and vast, would remain beyond our reach and the reach of our children.

The present stage in home computer learning is like the days of the Wild West. We have all sorts of people in the home-learning software industry, including cutthroats, gunslingers, and the like. But we also have pioneers, scouts, traders, settlers, and explorers. And we have gypsylike Indians roaming freely through the whole terrain.

We certainly need some sheriffs and marshals in all this hooting rowdiness, in these gun duels and this disorder. But we do not need an outside expert or government official to impose mock order by garrisoning us off and forcing us onto tiny reservations. We still have too much exploring to do.

Structured vs. Unstructured Learning

According to one expert, over 10,000 companies have already created over 40,000 software packages, any of which, potentially, could be used for learning. And within another year, this number will double.

Much of the software is excellent. But there is also a lot of junk out there. And there is no way for the average consumer, a parent or a teacher, to separate the junk from programs that will help them or their children — especially since most software can't be previewed before taking it home.

People are concerned. It is natural for them to turn to government policymakers and educational experts for some help and respite. And when the policymakers and experts get involved, it will be natural for them to create a model for home learning based on learning at school.

Unfortunately, this would be a mistake.

Why? First, because learning takes place in school primarily in a structured environment, while learning at home is largely unstructured.

Second, learning at school takes place under the pressure and prod of a teacher's leadership, the school's disciplinary and academic atmosphere, and the competition, opinions, and watchful eyes of one's peers.

In comparison, learning at home is normally done in a psychological vacuum. Parents cannot hope to duplicate the school environment. Most parents do not have the time to play the roles, night after night, of cheerleader, coach, taskmaster, teacher, and friend that a learner, especially a young learner, often finds vital.

Learning at school is curriculum-based, cumulative, and progressive. It follows a well-marked path laid out by the teacher's learning plan, by the school board, the boards of education and testing, and the committees of accreditation.

On the other hand, learning at home is ad hoc and sporadic; it happens in bursts. There is no curriculum or lesson plan to follow. There are no formal standards to meet, to fail, or to surpass. Learning at home is usually marked by the joy, the pain, or the insight of the moment, rather than the result of a coordinated plan followed over days, months, and years.

A Sense for What Is Right

At school, a teacher can be a leader because he or she has some sense of what is right. This sense comes from training, years of experience, fellow teachers and colleagues, from the insights gleaned from professional books and magazines, and from attendance at conferences and meetings.

In contrast, at home a parent, as teacher, flies by the seat of the pants. Parents have to trust their gut feelings and their dim memories of being students themselves. Parents can certainly nag and demand that their children sit down and do their work, but they can't get them to learn — unless they can somehow lead them into learning. But how can a parent be a leader unless he or she knows where or how to lead?

A New Curriculum for Home Learning

Parents need guidelines for a home-learning curriculum. But it must be something totally new, unlike any curriculum found in school. And parents need help in making decisions as to which home-learning software they should purchase, and how to derive the greatest benefit from that software for their children.

They can't follow the school model. The school "carrots and sticks" will probably not work at home. Children who learn at home, over the long term, will need more incentive than their mom or dad threatening and nagging them, day after day, week after week. Children who learn at home will need software that inspires them, challenges them, and gives them free rein to learn independently and at their own pace.

Above all else, learning software for the home must be *entertaining*. The incentive for learning must come from learning itself. It cannot be imposed from outside. Otherwise, the long-term effect on the child is likely to be more negative than positive. Children will come to resent enforced learning on the computer just as much as I resented being strapped into the piano seat for a half-hour of practice every day when I was growing up. After eight years of this kind of education I came to hate the piano. Today, many years later, I still have a mental block about sitting down at the piano and playing anything.

If parents are to succeed over the long run, computer learning at home must be fun, even joyful. And it must be meaningful to the child. The purpose and meaning of what the child is doing must be clear, not just to the parent, but also to the child.

Equally important, the child must have control over the direction and extent of his or her learning. Otherwise, the child is an automaton or puppet, and will derive very little satisfaction, pleasure, or real learning from all those accumulated hours in front of the computer.

Learning on the Computer

Opportunities for Home Learning

If school models for education are artificially grafted onto the home, computer learning could become very dreary indeed.

Yet something must be done.

Home learning using computers may soon be the complement to and the extension of learning in the school. Preschool children will learn at home on computers. School-age children will do their homework on computers and get remedial instruction. Handicapped children and those with learning disabilities will get valuable learning assistance from the computer to help them keep up with or even move ahead of their classmates. Talented and gifted children will be able to use the computer as a space shuttle of learning. They will be able to blast off into new areas, on their own, areas that challenge and stretch them to the utmost. They will be able to free themselves from the fetters and the crippling fear of failure they may feel in front of parents, teachers, and peers.

Computer learning at home will also be valuable as an "eleventh period." Children will be able to learn subjects and skills not offered by their school.

Adult computer learning, too, will be important. Schools will be able to provide continuing education courses for adults at home, using computers. Adults will be able to acquire valuable job skills and gain academic degrees by using computers to learn at home.

The Free Enterprise Model

Learning at home shouldn't be constricted by a school-like institutional curriculum or standards. Instead, new kinds of curricula and standards should be created, based on realistic conditions that exist in the average home.

Learning at home on a computer should be as diverse as possible to reflect different families' and individuals' interests, personalities, goals, and abilities. Diversity in computer learning should reflect (and *enhance*) the diversity in people.

Educators need to work along with educational policymakers, parents, children, and computer users to come up with a diversity of new standards, materials, and curricula for home learning.

The best model for home learning might be a *free enterprise model*. Major government bodies, computer users groups, educators, private companies, and consumer groups should each come up with their own packages. There might, for example, be a McGraw-Hill Comprehensive Package of Computer Home Learning Materials, and other packages from Scott Foresman, Addison-Wesley, D.C.

Heath, etc. There might also be packages from MECC (the Minnesota Educational Computing Consortium), Apple, IBM, Atari, Tandy, and, of course, Commodore. Consumer's Union might have its own package. Children's Television Workshop, CBS Software, Scholastic, Reader's Digest, Sunburst Software, and HesWare might have their own packages.

Each of these packages would compete for the biggest share of computer users. Parents could read evaluations and descriptions of the packages, talk to dealers, and preview the software before choosing the package that was right for them and their family.

Extracurricular Learning

Not all home learning should be curriculum-based. Not even if we redefine curriculum to be something appropriate for homes and families.

Many kinds of software companies should continue producing what they do best — one-shot, maverick programs that are unlike anything anyone has even seen. These are works of art that delight, charm, entertain, and educate, all at the same time. They might not fit easily into a package or a curriculum, but they deserve to be seen and experienced by every family.

Also, there should be lots of room for content-free, "learning how to learn" software. I would welcome lots of new programs that don't teach us when the Pilgrims landed on Plymouth Rock, or how to conjugate a verb in Spanish, or how to solve an algebra problem. Instead they would teach us to be better learners. These programs would help us in all our learning, at home and at school.

Furthermore, learning at home and at school are not always different. I think that many of the unstructured learning exercises targeted at the home could be used in special, unstructural learning times at school. And many types of courseware aimed primarily for the school could be used, with proper materials and parental guidance, in the home. Above all, there should not be a wall separating learning at home from learning at school. Instead, it should be a broad, circular continuum that melts together and meets at either end.

Horse Breeders, Plumbers, and Brain Surgeons

Completely new modes of learning may be discovered yet.

A home-learning curriculum might be devised based entirely on real-world career domains. For example, all computer-based home-learning courses shouldn't be on just *knowledge domains,* such as Algebra I or language arts. Software companies should also offer children full-scale courses on how to be a space shuttle pilot, how to manage a nuclear reactor, how to be a software designer, a fashion photographer, a horse breeder, an archaeologist, a diplomat stationed in Latin America, an executive in a multinational corporation, a plumber, brain surgeon, or a police detective.

In these courses, knowledge domains would be subsidiary to *career domains*. Kids would pick up the physics, math, language arts, and social studies they needed to get their credentials in the various fields. The youngest kids would naturally become junior horse breeders and archaeologists. The models they would have to master would be simpler, yet for them, no less exciting and challenging.

Older kids would have to work with more complex, lifelike models of the real world and of the careers they were studying. They would work for milestones like Apprentice and Assistant on the way to achieving mastery of the career.

Perhaps work-study internships could even be set up to coincide with advanced home-study programs for teenagers and young adults. Companies and goverment agencies could open their doors to student interns who had mastered their career courses. In this way, young people could complement their home learning with on-the-job training and real-world experiences. Employers would benefit by getting to see a crop of enthusiastic, well-trained young people.

The programs of study should be diverse, entertaining, and short enough to encourage children to try as many careers as possible. The programs should be rewarding, playful, and encourage social and emotional skills as well as intellectual skills.

Preparing for Computers

Now that computers are going into the schools in record numbers, we are finally realizing that we have gone about things all wrong.

Before computers arrived in the schools, we should have laid some groundwork.

Parents need to be briefed. Teachers need to be trained. Schools need some way to purchase software, course materials, books, and magazines. Teachers need some guidelines for purchasing new computers, new software, new computer equipment and materials. Standards and procedures for product review and evaluation need to be agreed on.

Children need to be consulted.

Parent Training

The situation at home is even worse.

Computers are popping up by the millions at home. Parents who don't know anything about computers are running out and buying software, materials, and equipment based on dealer recommendations, the pictures on the software packages, and occasional reviews and recommendations they see in magazines and on television.

I think most parents are anxious and bewildered about computers. But they are also incredibly curious about what can be done. "How can my kids use this computer?" they are asking. "How can my kids use this computer to be happy and successful?" "What can my kids learn on this computer?" "*How* can my kids learn?" "How can this computer help my kids at home?" "How can it help them in their schoolwork, and prepare them for growing up?"

In school, people are finally realizing that teachers need in-service training in order to integrate computers into the schools.

And if teachers are getting training, why not parents?

Courses should be set up for parents to attend. The courses should be jointly created by teachers, computer vendors, parents, and children. The courses should concentrate on training parents on how to use computers as home-learning tools.

Learning at *home* should be emphasized. Parents should not be taught to copy what teachers are doing at school. The situation is different at home. Parents should be given the information and skills they need to cope with the problems they face at home.

A parent-training curriculum might include the following areas:

- Which computers best lend themselves to home learning?
- What are the best home-learning programs?
- Which new kinds of computer equipment can help home learning — like touch pads, light pens, speech synthesizers, and keyboard overlays?
- What materials should come with software to help guide parents and to supplement computer learning?
- What are some basic guidelines to help parents evaluate home-learning software and materials?
- Which publications, catalogs, etc., bring the best new equipment, software, and materials to parents' attention?
- What techniques should parents use to actually *shop* for and *purchase* home-learning materials?
- What are the best local stores — in terms of service, hand holding, training, and dealer support? What are the stores

with the nicest, gentlest salespeople — people who like and understand children and who know how to talk in English?
- How can parents set up user groups for parents who are interested in teaching preschoolers, elementary-age children, or secondary-age children, or handicapped or learning-disabled children, or children who need help in science, social studies, language arts, math, reading, or writing, or kids who are talented or gifted?

Learning by Surprise

Thanks to the computer, learning at home will soon be as important as learning at school. But it can't replace school. Nor should it try.

Computer learning at home should reinforce, complement, and supplement classroom-based education. Parents and teachers should work closely together to make sure that the mix of home and school learning is the most efficient mix possible and in the best interest of their children. Coordination is crucial.

The debate on home learning using computers has just begun. Most of the important subjects haven't even been covered — or discovered.

Education at school can receive important, surprising boosts from home computers. For example, let's say a child is having trouble with social studies at school. The child has the aptitude for the subject but is utterly bored by the material. Boredom and lack of interest are reflected in the child's grades and behavior in the classroom.

The parents could strap the child to a desk every night and require doing extra lessons and more time studying the boring textbook. Or they could purchase some of the new "social studies simulations" software. The software might turn the child into a cartographer to map out a newly discovered continent, or a population planner, or an advisor to President Lincoln during the Civil War.

The most important thing the software could do for the child is *bring the subject to life*. It could awaken the child's interest and bring an excitement to the subject which could transform the child's whole attitude and performance in the classroom.

Learning As Entertainment

I think that computer learning at home may soon cease to be called "learning." Instead, we might end up putting it under the category of *entertainment*.

Learning on the Computer

Think about it. The words *education* and *learning*, for many adults and children, have a negative connotation. For them, the experience of learning is associated with pain, embarrassment, tedium, and boredom.

Computer learning often isn't any different. But it can be.

Computer learning can be made so pleasurable, so joyous, and so rewarding that it may slip unnoticed into the category of entertainment. It.may become a new form of entertainment that feels good to people at the same time it benefits and changes them in a substantial way.

If home-learning programs are designed with enough imagination, subtlety, and respect for people, they may soon become more popular than videogames were in their heyday. Families will gather around the computer, like an "electronic hearth." Instead of watching TV, they will spend many active, enjoyable hours together every evening learning new things on their computer.

And they won't think of it as work, learning, or education. They will think of it as entertainment.

Learning As Expression

Computers can open new doors into areas of self-expression we never knew existed.

New programs like *Music Construction Set, Micro Illustrator, MusiCalc, Delta Music, Fun Writer,* and *Word Vision* allow us to express ourselves in colors, shapes, and designs, in words, and in music.

And we can start creating the moment we sit down at the computer.

Gone are the hours of manual-reading just to learn how to use the computer. Gone are the additional hours needed to master the technical complexities of drawing, sight-reading musical scores, composing, or typing on a standard typewriter.

With the right software, the computer can act as a booster and an amplifier for our skills and can enable us to directly tap our imagination and produce new creations in various media.

Most of us, by the time we become adults, walk around with a little editor inside our heads. Whenever we do anything, the editor reminds us that we are either good or bad at that thing. It tells us whether we are knowledgeable or ignorant, skilled or unskilled. It assesses our past efforts and predicts whether our performance will be graceful or awkward, and whether the results will be elegant or ugly.

We pay close attention to our editor. Listening to the editor keeps us from making fools of ourselves in new situations. For example, by the time we are adults, if we haven't become accomplished musicians, writers, or artists, we shy away from these areas. We do not casually sit down with guitars, typewriters, or paintbrushes. We know how bad we'd look, and the kind of trash we'd produce. We know because our editor tells us.

But now we have a way of disabling that editor and, simultaneously, of being able to produce works of art that are beautiful and *personally satisfying*. We can do all this by using new kinds of computer tools.

Just the Beginning

But expressing ourselves is just the beginning. These programs should also act as a stimulus to get people learning more about the discipline of music, the discipline of art, and the discipline of writing.

New programs should pick up where these programs leave off. They should be more than "builder kits" and "construction sets." They should challenge us to a higher level of achievement in each of these disciplines, while rewarding us with beautiful creations along the way.

And why do we have to have construction sets for the arts?

Why don't we see mathematics construction sets, physics construction sets, chemistry construction sets, and biology construction sets?

How would you like to build a budding rose, design a working star, or construct an erupting volcano? How would you like to create a *working* model of the human heart? Or construct a gene, a DNA molecule, a bacterium, or a one-celled amoeba then bring it to life?

The delight and thrill you'd feel wouldn't come just from the intellectual experience of building a computer model of a heart, an amoeba, or a volcano. It would come from creating a beautiful, emotionally satisfying work of *art* — like creating a pretty picture or a moving piece of music.

The computer's greatest value as a learning tool comes when it mixes process and product, when it blends the technical and scientific with the artistic, and when it frees us of inhibitions and taps our imagination, yet still challenges us to acquire more skill and do better.

Learning by surprise, learning through entertainment, and learning by expression are some of the ways computers can help people learn. They are ingredients for profound and permanent learning. These ingredients and many more, as yet undiscovered, should be present when we use computers to learn at home. They should be included in new standards for computer-based home learning.

Chapter 7

Computer Literacy and Beyond

From Dinosaurs to Freckles

Have you ever seen a dinosaur spit out numbers? Forty years ago, dinosaurs inhabited Earth, just like in prehistoric times. The dinosaurs were enormous. Some were the size of your living room. Others were even bigger. They filled warehouses, laboratories, and entire city blocks. And when they were well fed, they spit out numbers.

The dinosaurs had metal skin. Inside their bodies were millions of wires, some the size of jungle pythons. The dinosaurs were controlled by thousands of hot, glowing vacuum tubes the size of big dill pickles. The vacuum tubes acted like traffic cops and routed the flow of electricity through the dinosaurs' wires.

The dinosaurs spit out numbers. They also ate them. The dinosaurs liked only two kinds of numbers: ones and zeros. Dozens of human beings fed them ones and zeros in long, caterpillar-like strings.

The humans hoped that after the dinosaurs finished eating, they would say something wise. They hoped the dinosaurs would solve their problems. But the dinosaurs were slow. After weeks of eating bucket loads of numbers, the dinosaurs finally answered. Unfortunately, they were often wrong.

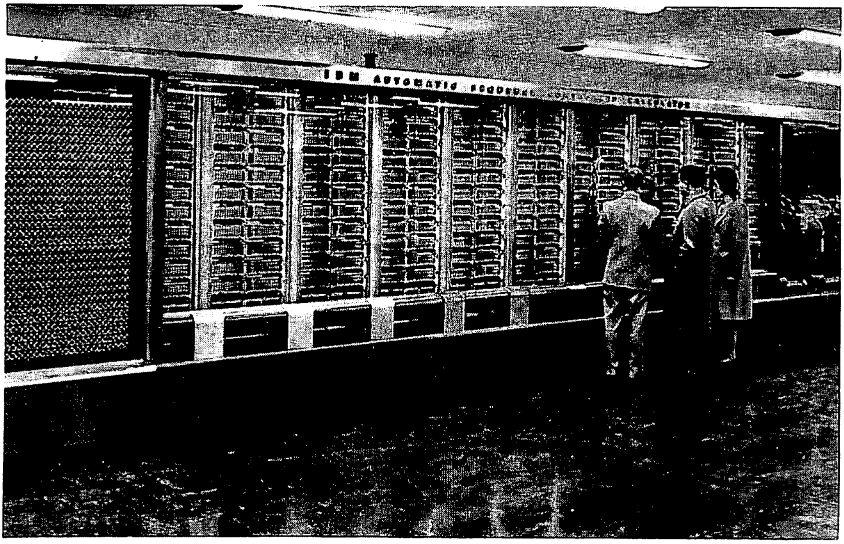
In some ways, these recent dinosaurs were unlike their ancestors. For example, the old prehistoric dinosaurs were mostly brawn. They had tiny brains, the size of a pea or a walnut. Like the old dinosaurs, the new dinosaurs were big. But their bigness was all brain.

The new dinosaurs were different in another way, too. The first dinosaurs were living creatures. They were *reptiles*. Their descendants include alligators, crocodiles, snakes, and lizards. These creatures are alive today.

But the new dinosaurs were not alive. They weren't even creatures at all. They were machines. They were the world's first *computers*!

The computers were almost as rare as dinosaurs. For several years after they were invented, there were only half a dozen computers scattered across the whole world. And though they were big, they were sensitive and fussy creatures. Every time you turned them on, one or two of their vacuum tubes would explode from the surge of electrical power. Then it sometimes took people several days to find the bad tubes among the thousands of good ones.

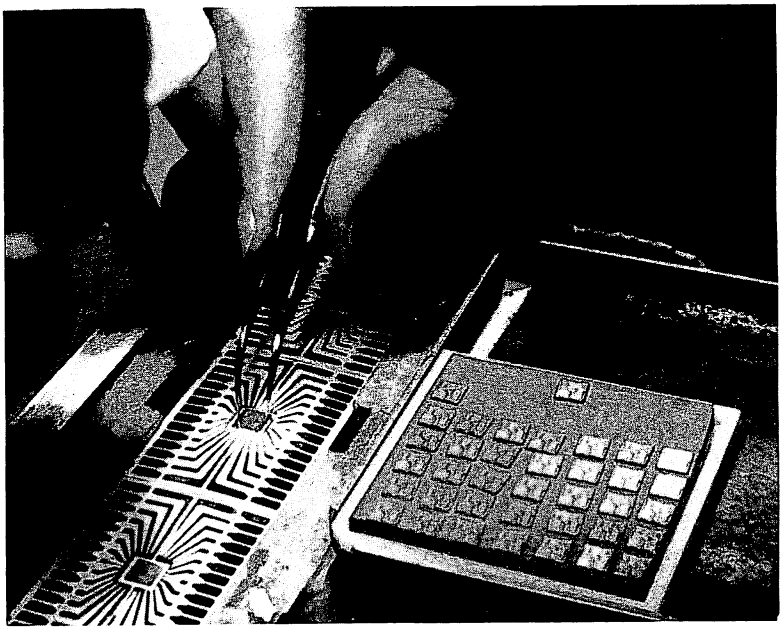
Almost as soon as computers were invented, scientists, business people, and military generals realized that computers, one day, could become important machines. Computers could help people conduct experiments, solve mathematical problems, process a company's records, and guide missiles, satellites, and spaceships. But somehow, they had to become smaller, faster, cheaper, and more reliable.



A "dinosaur" computer of the 1940's. When the IBM Mark I began working in 1944, it sounded like a roomful of knitters using metal needles. It went "clickety-clickety-clickety."
Courtesty of IBM.

No More Tubes

Then a breakthrough occurred. Two days before Christmas in 1947, scientists at Bell Laboratories in New Jersey invented the *transistor*. Transistors could function just like a computer's vacuum tubes. They could act like magnifying glasses and create a big electrical charge from a small charge. They could act like traffic cops and route charges through a computer's thousands of wires. And they could act like light switches and turn charges on and off.



"Brain" chips are lined up like soldiers on parade. The packages resemble spiders with 28 gold legs, so they are known as "bugs."
Courtesy of Texas Instruments Inc.

Transistors could do everything vacuum tubes could do. They were also cheaper, smaller, faster, and more reliable.

The first transistors, used in computers in the late 1950s, were about the size of paper clips or small Tootsie Rolls. This was small, but it wasn't small enough. Scientists kept searching for new ways to make the transistors even smaller.

In the early 1960s, scientists invented the *integrated circuit* (or IC). The integrated circuit could squeeze up to a hundred transistors onto a round surface the size of a small sugar cookie.

To make an IC, scientists grew a large *silicon crystal* in their laboratory. Silicon is an element — one of the basic building blocks of the universe, like oxygen, mercury, and iron. When you go to the beach, you see silicon everywhere, mixed in with the sand. It is the silicon that sparkles up at you when you run across the beach on a sunny day.

The silicon crystals in the scientists' labs resembled long, fat Italian sausages. The scientists sliced the crystals into thin wafers using an extremely sharp buzz saw. They took a photograph of lots of transistors' wires, then reduced (or shrank) the photograph until it was the size of one of the wafers. They placed the photograph on top of the wafer and dropped the wafer in a strong chemical bath. The chemical dug tiny trenches across the surface of the wafer. The trenches followed the wires in the photograph. When the chemical evaporated, the trenches were filled with metal. They had become transistors and incredibly tiny pathways for electricity.

The scientists cut the round silicon wafer into little squares the size of a bread crumb. Each little square was called a *chip*. On its surface were dozens of transistors. The transistors crisscrossed the chip's surface, like a maze of tiny roads.

Computers on a Chip

During the 1960s and 1970s, scientists found ways to pack more and more transistors onto a single chip. Today, in the mid-1980s, scientists are able to build a chip with more than a *million* transistors.

The first chips were primitive. With only a couple of transistors, all they could do was turn lights off and on, or remember a couple of numbers, like 5 and 14.

Today's chips are completely different. They can do almost anything. A single chip can act as a computer brain and add a million numbers in only one second. It can act as a computer memory and remember a hundred thousand kids' birthdays. It can tell the time, control a car, guide a robot, or act as your opponent in an electronic game.

An entire computer can fit on a single chip. But chip-sized computers are too small and delicate for us to carry around in our pockets. To use them, we must connect them to something larger. We can wear them on our wrists inside digital watches. Or we can hide them inside dishwashers, arcade games, and microwave ovens. Or, if we connect them to typewriters, TV screens, and tape recorders, they become *personal computers*.

Kids use personal computers to draw pictures, make music, do their schoolwork, and play games. Personal computers fit on top of a kitchen table.

Personal computers are small and easy to use. They use less power than a simple light bulb. Yet they are a million times more powerful than the dinosaur-sized computers of forty years ago.

The early computers weighed more than a basketball team of elephants. They were powered by up to 20,000 vacuum tubes and cost millions of dollars.

A personal computer might weigh less than five pounds. It might cost less than $100. And its brain is a chip the size of a freckle.

Electronic Art

Let's imagine we have a time machine. We climb on board. We shut the door. We set the controls for Cambridge, Massachusetts, in early 1961.

The time machine whirs. Our stomachs feel queasy, as if we were on a rapidly falling elevator.

The whirring stops. The door opens.

We are in a darkened laboratory. The hulking forms of giant computers tower overhead and surround us. In the laboratory is a young man, unshaven, gazing at the screen of a computer terminal.

The young man is Ivan Sutherland. Ivan is a graduate student at the Massachusetts Institute of Technology in Cambridge. He sits, frozen, in front of the computer. It is 3:30 in the morning. His wife and kids are at home sleeping. But here he is, dead tired, half-asleep, and eyes glazed. Yet he has to be here. It is the only time he is allowed on the computer.

Sutherland stares at the computer screen. A shape appears. Sutherland grins. "Yaaaaaa-hoooo!" he cheers. He dances around the deserted lab.

Why is Sutherland so happy? Because he has just become the first human being to teach a computer to draw.

On the screen was a straight line. That's all: just a straight line. It was special because it was the first line ever drawn by a computer — and because it opened a whole new age of electronic art.

Bending a Line

Sutherland was excited about what he did. But he didn't stop there; he went on to teach the computer how to draw new things. He taught the computer to change the straight line into an elastic rubber band. He had the computer bend the line, shrink it, and stretch it.

He taught the computer to draw other shapes, too: circles, triangles, squares, and polygons. He taught the computer to spin the circles, rotate the triangles, and fold the squares like pieces of construction paper.

Sutherland published his findings in a book called *The Electronic Sketchpad*. His enthusiasm and knowledge about computer art inspired a whole generation of young people to learn how to teach computers to draw pictures.

Sutherland and his friend David Evans moved to Salt Lake City, Utah, and founded the world's best *computer graphics* company, Evans & Sutherland.

Evans & Sutherland now builds million-dollar flight simulators for the U.S. Air Force. The simulators look like the inside of an airplane's cockpit — except that the windows are all computer screens. When a pilot trainee operates the controls of the plane, he or she is really controlling the computer. The computer images on the windows look like what pilots would see if they were flying a real plane.

Sutherland and his graphics computers are world-famous. Yet it all started one cold, dark morning when he taught a computer how to draw a straight line.

Today's Graphics

Amazingly, your average personal computer is more powerful than Ivan Sutherland's warehouse-sized computer of 1961. Also, many of today's personal computers have graphics commands built right into their BASIC language. Sutherland had to program his computer to draw by feeding it commands written in long, snakelike strings of binary 1's and 0's. Modern computers can draw pictures with English-like commands such as PEN DOWN, DRAWTO, PLOT, and FILL.

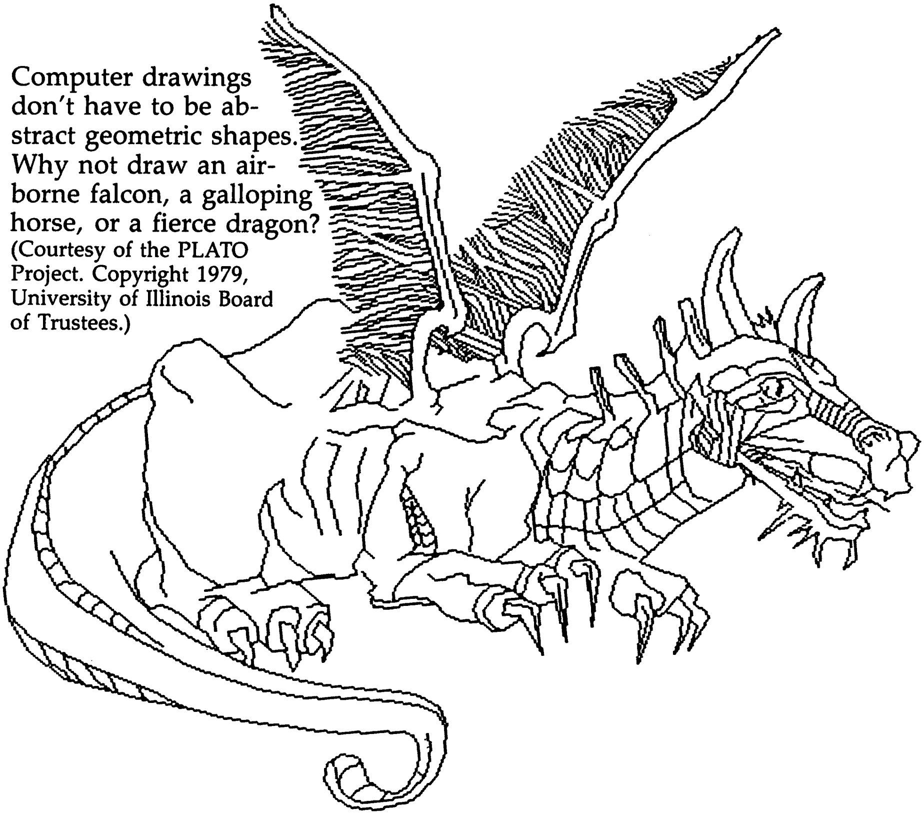
Also, Sutherland's graphics were all in black and white. But today's computer graphics can be in color — up to 256 different colors on some machines.

But no matter how complicated computer graphics get, they must be built in one of three ways.

First, you can use letters or special graphics characters and combine them into some kind of shape. The shape might be a giant letter A, a monster's face, or a skyscraper. These kinds of graphics are holdovers from the days when most computer terminals used paper printers instead of video display screens.

Another way to draw pictures on a computer's TV screen is to divide the screen up into tiny blocks called *pixels* (for *picture elements*). The picture is built by filling in the blocks. It's like drawing rough pictures with graph paper and magic markers. Using this technique, you can make pictures of spaceships, human stick figures, or running horses.

Computers that can display large numbers of very small pixels are capable of creating images with finer resolution. The more pixels, the better. If your TV screen is divided into thousands of tiny



Computer drawings don't have to be abstract geometric shapes. Why not draw an airborne falcon, a galloping horse, or a fierce dragon? (Courtesy of the PLATO Project. Copyright 1979, University of Illinois Board of Trustees.)

building blocks, the pictures look smooth and realistic. On the other hand, if the building blocks are big, the pictures appear blocky and rough.

Most personal computers let you draw pictures using at least a thousand blocks per screen. Some microcomputers let you use up to a quarter of a million blocks. With this many blocks on the screen, they look less like blocks and more like tiny dots.

Yet another way to create computer art is to use *turtle graphics*. Many personal computers can run the PILOT or Logo languages. In turtle graphics, a small, imaginary turtle (often the shape of a triangle) walks across the screen, leaving behind a trail. By issuing commands such as TURN 90 and FORWARD 10, shapes are drawn.

The turtle draws pictures by connecting two points on the screen with a straight line. The turtle can draw simple shapes such as squares or triangles. Or it can draw lots of little straight lines to make a circle. The turtle makes the circle by going forward one tiny space, then turning to the right one degree. Since it takes 360 degrees to make a circle, the turtle has to go forward and turn right 360 times.

The turtle doesn't have to stick to circles. It can make stars, snowflakes, even fierce dragons. It can make any number of complicated, beautiful shapes — all from tiny straight lines (see example printout).

The turtle usually draws with lines made up of dozens of blocks strung together. But its lines could be made up of special figures or letters of the alphabet.

Graphics Using Pencil and Paper

Until recently, the only way to make computer graphics was to type commands on the computer keyboard. But that's not the way traditional artists draw. They use pencils, pens, and paintbrushes. They draw the picture directly on a piece of paper. They don't have to type a PAINT command on their paintbrush, or a SKETCH command on their pencil.

Now computers are being made to work with equipment and programs that will make it as easy to draw pictures on a video screen as on paper. Computer art will be more like drawing with a pen, pencil, or paintbrush.

One relatively new device is the *light pen*. The pen is wired to the computer. As you touch the tip of the pen to the screen and move it across the glass, it leaves an electronic line. Some light pens draw in different colors. Others let you touch the screen and fill in a whole picture with a particular color. With a single touch, you could paint the sky blue, clouds white, or a robot metallic orange (see photo).

Another device is the graphics tablet. The tablet is a flat plastic rectangle or square. You mount a picture you want to trace on top of the tablet. Above your picture is a plastic arm, often with a magnifying glass on the end.

You trace the picture by moving the tip of the plastic arm above each line in the picture. The arm and the graphics tablet are wired to the computer. As you move the arm, it sends signals to the computer. These signals are an electronic copy of your picture, stored in the computer's memory. With just a couple of commands, you can get the computer to display the picture on the TV screen. It is fuzzier than the original, and the colors are different, but it is still very much like the picture you traced (see photo).

You can use a graphics tablet to make computer images of maps, photographs, shapes, drawings — almost anything. You can also draw original pictures of your own with graphics tablets. An electronic copy is stored inside the computer and appears on the TV. On some computers, you can manipulate the picture once it is inside the computer: shrink it, expand it, change its colors, spin it around, or stretch it apart like Silly Putty.

Million-dollar mainframe computers are being used more and more to create graphics for things like flight simulators and Hollywood movies (such as *TRON*). These computers are becoming so powerful that, sometimes, their pictures look like photographs of real people, real places, and real things.

Drawing pictures directly on the TV screen with a light pen.
Courtesy of Matsushita Electric.

Moviemakers and human artists are programming computers to create movie and TV scenes that would be too expensive or impossible to stage with live human actors. In days gone by, a team of human artists would paint these scenes (called *mattes*). Now humans program computers to paint the scenes. And the scenes aren't frozen, like a still photo. The computer brings the scenes to life. They become realistic computer cartoons.

Computers that draw scenes for movies are far too expensive to become personal computers today. But scientists are inventing a new generation of special-purpose computer chips that will soon drastically reduce the cost of these computers. Million-dollar computers can now fit on a $5000 graphics chip. One day soon, these chips will be inside personal computers. The results will be the birth of a new era in computer graphics and electronic art.

Further Reading

For an overview of computer graphics and computer art, I recommend the following books:

D'Ignazio, Fred. *The Creative Kid's Guide to Home Computers*. New York: Doubleday and Company, Inc., 1981.

D'Ignazio, Fred. *Electronic Games*. New York: Franklin Watts, Inc., 1982.

D'Ignazio, Fred. *Messner's Introduction to the Computer*. New York: Simon & Schuster/Julian Messner, 1983. (This book contains my interviews with many of the pioneers in computer graphics and computer art.)

Greenberg, Donald (and others). *The Computer Image: Applications of Computer Graphics*. Reading, MA: Addison-Wesley, 1982. (This is an expensive book. But it has beautiful color pictures and is a good, clear introduction to computer graphics. I recommend it for your computer class at school or for your library.)

Leavitt, Ruth. *The Artist and the Computer*. New York: Harmony Books, 1976.

If you are ready to start creating computer art of your own, you should look at:

Myers, Roy. *Microcomputer Graphics*. Reading, MA: Addison-Wesley, 1982. (Lots of Apple II BASIC examples.)

Thornburg, David. *Picture This!* Reading, MA: Addison-Wesley, 1982. (Lots of examples in Logo and Atari PILOT.)

Architects of the Microworld

Our journey into the world inside the computer might be like some exotic travel story. It might be like Jack London telling tales from the far north, or like Gulliver describing his voyage to Lilliput. It might be far removed from our everyday experience and the concerns of us and our children.

Except it's not.

All of the changes, the fascinating developments in chip design and technology, might not be expected for many years, and these new designers might all be faceless adult experts, hidden away inside corporate and university labs, performing mysterious feats of electronic alchemy.

Except they're not.

A few years ago, Lynn Conway of Xerox and Carver Mead of Cal Tech wrote a book called *An Introduction to VLSI Systems* (Addison-Wesley, 1980). In it, and in college courses they taught, Conway and Mead called for a new generation of chip designers — architects of the microworld. Conway and Mead proved that, using automated drafting tools (CAD/CAM — Computer-Aided Design/Computer-Aided Manufacturing), even graduate students at universities could design custom-made computer chips. And they didn't have to be engineers or experts in computer logic.

Young people by the thousands are following the lead of Conway and Mead's students, and are designing custom-made chips and revolutionary new kinds of computers. They treat the basic components — the transistors, gates, and memory cells — like "bristle blocks." They sit in front of a computer keyboard, press buttons, and the computer fits the bristle blocks together and displays the mazelike circuit on a color picture screen. When the chip is finished, the student and his teacher send the design to a "silicon foundry" — a regional center that prints the chip on a sliver of silicon.

Within days, the student has the chip back and can plug it into a circuit board and turn on the power. The chip might be a new kind of computer, a "graphics engine" specializing in high-speed, animated color pictures, or a music-synthesis chip capable of making the computer sound like a bass guitar or a pipe organ.

For two decades, chip design was done exclusively by experts at major corporations, such as Intel, Texas Instruments, and Motorola. Now, all this has changed, due to the new CAD/CAM tools, to the new microminiaturization (VLSI — Very Large-Scale Integration) of circuits, and to Conway and Mead's paint-by-numbers approach to chip design.

Today, computer *software* is being written and sold by thousands of people all over the world. It is a cottage industry performed in the home by a lot of low-budget suppliers. Big companies are becoming software publishers and distributors for the software authors.

Similarly, in just a few years we will probably see a cottage industry of chip authors. CAD/CAM equipment will be rented or cheap enough to install in the home. "Intelligent" chip design programs, like Xerox's *Palladio*, will work with young chip designers. The programs will check the young people's designs for mistakes, suggest new designs of their own, and explain the tradeoffs between different designs.

Chip architects are now appearing in colleges and universities. They will soon appear in high school. They will design thousands, even millions of new computers. The best computers will find their way to the marketplace. Large "chip foundries" will print the chips. Large "chip publishers" will market them.

The Playdough Computer

What do we mean by "personal computers," or by "personal computing?"

Until now, we meant *one person, one machine* (Portia Isaacson's definition).

Personal computing has made the computer accessible to the average person, the nonexpert. Even little kids can sit down in front of the family's home computer and use it to learn or play a game.

But personal computing will soon have an added meaning. It will mean *computing on a machine we designed ourselves*. Even young people will be able to fashion their own computers, like they build an airplane or castle from playdough or clay. Everyone will have the opportunity to become a computer designer — a computer artist, an architect of the microworld.

Naturally, airplanes made of clay do not fly. And computers made by young children may not compute. Still, they will be an important exercise in creativity. And unlike models made from clay or plastic, simple computer models can gradually become more sophisticated and realistic. They can be a first step toward building real chips and computers that work.

Many of today's most innovative, most successful software entrepreneurs are young people, of college or high-school age. In the near future, young people will be some of the most creative chip architects, artists, and inventors.

Your Kids: the Computer's Creators

Although the surface of the computer will continue to remain the same, the computer's insides will soon change dramatically. New computers will soon be created with capabilities exceeding our wildest dreams. The revolutionary new computers will be molded from youthful imaginations. Young people will be the computer's creators.

Fantasy, Turtles, and Sprites

The present trend toward black-box computers is positive, in that it is encouraging millions of people to try personal computing. But don't let this trend "distance" you or your children from the world inside the computer. The real action is taking place under the computer's "hood."

How do you and your children learn more about the world inside the computer? How can you learn *today*?

Computer microworlds are so enchanting, so exciting that they can be a pleasure to learn about and explore. Pretend that you and your children are pioneers.

But what about a pioneering vehicle — a land rover, jeep, or canoe? After all, sophisticated computer-design tools, such as CAD/CAM machines, still aren't off-the-shelf items, sold at your local Radio Shack for $19.95. How can you explore the world inside the computer without the proper tools?

Fortunately, there are what Bob Albrecht calls the "Rainbow Computers" — the sound-and-color machines that are relatively inexpensive. They provide an environment for simulating the world inside the computer. Using the rainbow computers, you can build models of that world with computer programs.

What kind of programs? You can use a language like BASIC. But new languages are now appearing that make superior microworld building tools. These languages are Atari PILOT and the various versions of Logo, including Apple Logo and TI Logo.

Computer Literacy and Beyond

When you write a program in BASIC, you are like a chef inventing a recipe for a new food dish. The recipe itself is the list of steps you must follow (the *algorithm*) to get the program to perform some function. To make the recipe work, you need to add ingredients in precise amounts and in the proper order. In a program, these ingredients are your *data*.

But programming in other languages — especially Logo — is quite different. You are no longer a chef working with a recipe and ingredients. Now you are a band leader conducting a band of jazz musicians, or a film director overseeing the efforts of a diverse bunch of live actors.

When you program in BASIC, you are dealing with inert objects and structures, and the computer marches in lockstep, doing one thing at a time.

On the other hand, when you program in Logo, an *actor's* language, you treat the computer like a tiny world. You are the world's creator. You define the world's laws. You populate the world with tiny creatures. TI's Logo world comes with "canned" sprites. Various Logo, PILOT, and Pascal worlds come with prepackaged, artistic turtles.

And using these languages, you can also create other beings drawn purely from your own imagination.

When you type RUN, you turn the world *on*. You breathe life into your creatures and set them in motion. All together. All at the same time.

A Rabbi in a Box

Have you ever been to a novelty store and seen the Thing-in-a-Box? The box is really a small, black plastic cube. You flip an ON switch, and the cube begins whirring strangely. Then the top of the cube opens, and a little green hand pokes out, knocks the switch to OFF, and, in the blink of an eye, disappears back inside the cube.

After watching the box in action, you have the strong impression that someone — or something — is living inside.

We may soon have the same impression about personal computers.

Personal computers may soon have Artificial Intelligence (AI) programs and AI chips. Your computer may become an intelligent assistant that can think, reason, even learn. It will be a fluent computer that can carry on a casual conversation with you in normal English. If your family or friends won't listen to you, you will be able to turn on the computer and have a good heart-to-heart discussion with a machine.

AI research, tools, and technology are hot items. After almost four decades of research, AI professors have tied their bedsheets and pillowcases together into a rope, and have descended the ivory tower. Suddenly, at the same time their feet touch the ground, AI is becoming a big business.

What kind of big business? The hottest item in AI is known as *knowledge engineering*. AI scientists have created "expert systems" — chip-sized electronic clones of human experts. Already, these intelligent programs have been put to work diagnosing lung diseases, locating mineral deposits, and designing new clothing fashions. For a relatively small price, you will soon be able to have a world-class expert working alongside you, even if you are a lonely country doctor, or an oceanologist on an oil platform in the turbulent North Atlantic.

Putting a humanlike, intelligent program on a chip is like having a wise slave who lives in a package of gum, or like a genie who lives inside a music box. We will very shortly see experts-in-a-box, advisors-in-a-box, and teachers-in-a-box. The knowledge, expertise, and personality of a doctor, a minister, a psychoanalyst, or a lawyer can be mined and captured on a home computer in the form of a small, inexpensive chip. AI companies like Cognitive Systems, Inc., Computer Thought Corporation, and Machine Intelligence Corporation are already developing the first AI software for personal computers.

"Know thyself" is an ancient prescription and a unique ability of human beings. Only humans seem to have a strongly developed sense of self-consciousness and self-awareness.

But not for long.

New computer chips are being developed with "expert" programs that know how to design other computer chips. These chips are being fed huge quantities of knowledge. (Some use optical scanners and hungrily read all sorts of technical papers and reports.) What do they learn? They are learning about themselves, and about others of their kind. They are being endowed with a primitive sense of self-awareness.

Imagine someday when entire microuniverses inside the computer — the chip neighborhoods, cities, and worlds — all have a sense of self-knowledge and self-consciousness. This awareness will be only partly human. Much of it will be non-human, perhaps even alien. It will be suited to the chips' electronic, inorganic bodies and to the complicated knowledge pattern stored in their local and global memories.

Computer Literacy and Beyond

One day you and your children may design your own chips and create your own real computer worlds. Yet, even today, you can imagine them, then implement them in Pascal, PILOT, BASIC, or Logo. Right now the worlds are fanciful, the blue-sky stuff of fantasy and magic. But by inventing them and building them into programs, you and your kids will gain insight into the real computer worlds that await you. These worlds and the tools to fashion them already exist as blueprints and prototypes in scientists' labs. Before you know it, they will enter your office, your classroom, and your home.

Additional Reading

Anacker, W. (IBM) Computing at 4 Degrees Kelvin. *IEEE Spectrum*. May 1979. Page 26.

Business Week (Cover Story). Artificial Intelligence. March 8, 1982. Page 66.

Business Week (Special Report). The '80s Look in Chips: Custom, Not Standard. January 18, 1982. Page 36D.

Business Week (Cover Story). Japan's Strategy for the '80s. December 14, 1981. Page 39.

Carroll, Lewis. *The Annotated Alice* (Notes and Introduction by Martin Gardner). New York: Clarkson N. Potter, Inc., 1960.

Computer Design. Liquid Immersion Technology Cools Supercomputer's Circuits. February 1982. Page 58.

Conway, Lynn and Carver Mead. *An Introduction to VLSI Systems*. Reading, MA: Addison-Wesley Publishing Company, 1980.

Cromie, William J. and Harold A. Rodgers. The Big Squeeze: Packing More Circuitry into Less Space. *Technology Illustrated*. February/March 1982. Page 55.

D'Ignazio, Fred. *Katie and the Computer* (Illustrations by Stan Gilliam). Morristown, NJ: Creative Computing Press, 1979.

D'Ignazio, Fred. *Small Computers: Exploring Their Technology and Future*. New York: Franklin Watts, Inc., 1981

Matisoo, Juri. The Superconducting Computer. *Scientific American*. May 1980. Page 50.

Price, Derek de Solla. 3-D Intelligence. *Science Digest*. January 1982. Page 40.

Science. Are VLSI Microcircuits Too Hard to Design? July 11, 1980. Page 258.

Sweetnam, George. The Incredible Shrinking Microcircuit. *Science Digest*. January 1982. Page 80.

PILOT and Logo Addresses

Apple Logo. Terrapin, Inc., 678 Massachusetts Avenue #205, Cambridge, MA 02139. (617/492-8816)

Atari PILOT. Atari, Inc., 1196 Borregas Avenue, Sunnyvale, CA 94086. (408/745-2000)

TI Logo. Texas Instruments Inc., P.O. Box 10508, M/S 5849, Lubbock, TX 79408. (806/741-2978)

Young Peoples' Logo Association (YPLA), 1208 Hillsdale Drive, Richardson, TX 75081.

Books and Articles

Abelson, Harold. *Logo.* Peterborough, NH: Byte Books/McGraw-Hill, 1982

Abelson, Harold and Andrea diSessa. *Turtle Geometry: The Computer As a Medium for Exploring Mathematics.* Cambridge, MA: The MIT Press, 1981.

BYTE. Special Issue on Smalltalk. August 1981.

Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books, Inc., 1980.

Twilight of the Computer Hacker

Computer hackers were the pioneers of microcomputing. They were the enthusiasts — the zealots, even — who got this industry started, in the 1960s and the 1970s. Now we're in the mid-1980s, and what has happened to the hackers? Many kids are hackers. But how about the adults — the old revolutionaries? They're the computer elders who are being shoved aside by a stampeding horde of new personal computer users who seem to share none of their characteristics.

Enter the Computer Humanists

What are these new personal computer users like?

To find out, I went to two events — the Computer Faire at the Roanoke Valley Science Museum, and the "Computers in the Schools" conference at Hollins College, just outside Roanoke.

The best way to describe the new personal computer users is to say that they are the kind of crowd you'd find at a Roanoke bowling alley on a Friday night — only more diverse. There are a lot of them. And they are young, old, and in between. There are lots of women and lots of kids.

The new personal computer users talk funny, almost like regular people. You don't hear a lot about bits and bytes. Instead, you hear about computer model numbers. And computer components. And computer software. Most of the discussion centers on who has what piece of new software for which machine, and how it's the "neatest thing you've ever seen."

Another thing that distinguishes this new group from the old-line hackers is their energy. Everyone is excited. Parents are excited about bringing all the new educational programs into their homes. Kids are excited about the games. Teachers are excited about the ways they can use the computers in their classrooms. I sensed some fear and anxiety in this group, too — a certain tentativeness about really moving into this strange new world. But mostly I sensed excitement — a lot of it.

And the questions. Here is where the computer hackers and humanists diverged completely. The hackers asked how soldering on certain boards is done and about which section of memory is switch selectable. The humanists asked how computers for very young kids would alter the way kids develop their fine motor skills and the way that they get along with other people.

For example, I conducted a workshop at Hollins College on "Computer Literature." During the workshop a man raised his hand and said, "I am an artist and a teacher. I teach elementary schoolchildren how to paint and draw. If little children learn to paint beautiful pictures on a computer just by pressing a couple of buttons, how will they ever learn to draw on paper?"

How do you answer a question like that? I didn't try, but other people in my workshop did. One woman, an elementary school teacher, said, "Who cares if they draw on paper? In the future, no one will be using paper. They will be using TV screens for everything. They are the new medium for creative expression, not paper. Kids have to learn how to do this to be prepared."

Another teacher disagreed. "If kids only learn how to push buttons," she said, "they'll never develop their bodies. Their gross and fine motor skills will never be developed. These skills are crucial to a child's becoming an adult."

Another teacher added, "And what about the pleasure children have using art materials? Children who only work with TV screens and plastic buttons will never know the joy of using gooey finger-paints, and clay, and paste, and making cutouts out of construction paper. It doesn't sound modern to me, it sounds deprived."

Jiminy Cricket the Computer

During the workshop, I told the teachers about my concept of the computer friend. I said that I thought all personal computers would one day assume humanlike qualities and become intimate friends of their human masters.

One teacher applauded this prospect. "This will be a boon for human relations," she said. "It will let everyone, especially family members and young people, work out their feelings first with their computer. If something is troubling them, they don't have to keep it bottled up inside them. They can tell their computer friend. Then, after the matter is aired, they understand it more clearly, and they feel better about it and can approach members of their families and other human beings. A lot of situations are like powder kegs, ready to go off. Maybe the computer friend will help defuse them."

This point of view was echoed by another member of the workshop, who said she thought the computer friend might be like Pinocchio's friend Jiminy Cricket. The friend could be a person's conscience. The friend would hear a person's deepest secrets, his darkest and most whimsical fantasies and wishes.

At this point, two teenage girls raised their hands. "All this sounds dangerous to me," one of the girls said. "If computers are so friendly, they might lure people away from interacting with each other."

"She's right," said the other girl. "How can a human being be as patient as a computer friend? How can a person give another person their undivided attention the way a computer friend can?"

"And what about babies and toddlers?" added a woman teacher. "They may learn early in their lives to relate to a computer friend. But the friend may become a substitute for other people. This could stunt a child's ability to get along with people around her. Her earliest model for socialization wouldn't be a person, it would be a computer."

Computer Intimacy vs. Computer Literacy

In one of our workshop discussions, I questioned the big focus on computer literacy. I said that computer literacy was fine now while we were still in an age of transition from a noncomputerized society to a completely computerized society. In the future, however, when people have fully accepted computers and when computer technology has matured, we will move beyond computer literacy into computer *intimacy*.

I compared my audience to computer hackers, and said that it was obvious that the change from literacy to intimacy was already occurring. People were worrying less how computers worked and more how to "drive" computers the way they might drive an automobile. Also, they were showing concern about the impact of such a powerful technology on their lives, their students' lives, and the lives of their children. I likened introducing a computer into people's lives to sitting behind the wheel of a powerful race car. The car can take you places at breathtaking speeds, but if mishandled, it can maim and ruin.

Computers are not a technology of the body but a technology of the mind. As carriers, amplifiers, and multipliers of symbols, concepts, and ideas, they are far more powerful than automobiles. Whether they cause us to benefit or suffer depends on how we use them.

Beyond Computer Literacy

A recent national "computers in the schools" survey conducted by the Center for the Social Organization of Schools at Johns Hopkins University found that most secondary schools are using computers to teach programming. (For a copy of the survey, write to Dr. Henry Becker, Center for the Social Organization of Schools, The Johns Hopkins University, Baltimore, MD 21218.) According to the survey, the second most popular use of computers was for drill and practice, primarily for math and language arts. In addition, the majority of the teachers who responded to the survey said that they looked at the computer as a resource rather than as a tool.

I think this concentration on programming, drill, and practice and the image of computers as a "resource" are temporary. I believe that it is time for teachers and parents to start thinking beyond computer programming, beyond drill and practice, and beyond computer literacy.

The Computer Steam Engine

Two factors have caused teachers and parents to concentrate on the computer as a resource and to stress computer literacy. First, most computer courseware turns the computer into an electronic textbook. This kind of courseware is the most popular with teachers because it is the most familiar and the least threatening. The courseware (like a good textbook) introduces a new subject to a student, then drills the student on that subject.

Second, personal computers are still very primitive machines (compared to what they soon will be). They are a young, immature technology. Compared to what they'll be, the personal computers of today are like chugging steam engines, crude wooden plows, or fussy, cranky Model T's.

Despite manufacturers' claims, you *cannot* buy a personal computer and turn it on the way you would turn on a TV, then immediately begin to use it. Some computer literacy is still essential, or you quickly become lost in a nightmarish maze, pursued by horrible creatures like bytes, RAMs, ROMs, K's, RS-232s, modems, interfaces, bauds, Escapes, Breaks, and Resets.

The Age of Computer Literacy

Another recent survey (conducted by the University of Maryland) echoes the Johns Hopkins survey. It found that most schools introduce computers into the curriculum to help students become literate in computer technology.

But what does this literacy entail?

Is "computer literacy" programming? Is it the fundamentals of computer operation? Is it a quick course on using a computer keyboard? Is it drill and practice or the daily use of the computer as an electronic textbook?

Because of the pervasive spread of computers throughout our society, we have all become convinced that computers are important. From what we read and hear, when our kids grow up almost everyone will have to use computers in some aspect of their lives. This makes computers, as a subject, not only important, but also *relevant*.

An important, relevant subject like computers should be part of a school's curriculum. The question is how "Computers" ought to be taught.

Schools could teach about computers the way they teach about dozens of other important, relevant subjects (like math, social studies, geography, and language arts) — *with books*. However, since desktop computers are now relatively cheap, schools are buying computers so students can get a look at the machines themselves. Special computer classes are being set up so that students can play with computers, tinker with them, and learn some basic programming. Thus, on a practical level, computer literacy turns out to be mere computer *exposure*.

But exposure to what? Kids who are now enrolled in elementary and secondary schools are exposed to four aspects of computers. They learn that computers are programmable machines. They learn that computers are being used in all areas of society. They learn that computers make good electronic textbooks. And (something they already knew) they learn that computers are terrific game machines.

The Results

This exposure is worthwhile. It alone justifies a school's purchase of computers for its students. According to the surveys, real educational results have been realized at schools which concentrate on exposing kids to computers. First, students develop a familiarity with computer keyboards, computer operation, and computer concepts.

Second, students in these schools develop a realistic, positive image about computers. Past generations saw computers as electronic brains — abstract, all-powerful, and mysterious. Now kids get to see computers as they really are. Kids get to touch computers, play with them, push their buttons, order them about, and cope with computers' incredible dumbness, their awful pickiness, their exasperating bugs, and their ridiculous quirks.

Third, the surveys show that computers have played a big part in improving kids' (and teachers') attitudes toward school. Kids who use computers during their school day come early and stay late — just to have time on the computers. The whole school day goes better for everyone because it has a rosy glow caused by the computers. There are countless stories of learning-disabled kids, handicapped kids, and near dropouts who got turned on to computers and became model students. Kids with problems warm up to computers and, on their own, use them to improve their academic performance. Bright kids turn to computers as intellectual companions and resources and learn in a more personal, accelerated fashion.

Computers touch a kid's life. And the effect is cumulative. When enough young people are affected by computers, it changes the atmosphere of the entire school. The impact of computers on a school can be psychological. Computers can improve school spirit.

Last, computers make the students less fiercely competitive. Instead, they begin helping each other. Striving for academic excellence is a good thing. But in certain contexts, it can put kids under a great deal of pressure to succeed — with almost no help or support from their friends.

Here, too, the effect of computers on the social organization of learning has been significant. Computer classes have an atmosphere which is different from that found in many other classes. In computer class, teachers don't just teach, and students don't just give answers, write down notes, and take tests. In computer class, *everybody* learns, everybody shares, and everybody learns to be helpful. Teachers tell stories about how big, smart-aleck teenagers in their classes have put their arms around their shoulders, and with great patience and sincerity have explained how to boot a disk, load a program from tape, or master a new piece of software. Roles become reversed, fluid, and fuzzy. But often everyone benefits. And learning occurs at a rapid pace.

Computer Literacy and Beyond

Computers of the Future

Computers in schools have already had a substantial, positive effect.

But I'm still worried.

I think that schools are unintentionally locking their students into the present — the fleeting, short-lived Age of Computer Literacy. This is an age from which computers will emerge very soon. Computer literacy will become irrelevant and unimportant long before most students enter the job market.

Also, in many schools, computers are being taught in separate computer courses. This divorces them from the rest of the school, from the rest of the curriculum, and most importantly, from the other teachers.

Computers in our economy and in our society don't exist as islands of technology. Instead, they have become part of the fabric of everything we do. They are intimately involved with the way we live, move about, play, and do business. Just telling the students that this is so and teaching them BASIC or Logo is not adequate. We need to give them working experience with computers *as they are used in the real world.*

What's more, schools are using their newly acquired computers as an object of curiosity, as a hands-on device to learn the arcane arts of programming and computer operation, and as a teaching aid to learn math and language arts. But in the very near future, computers will be as common as TV sets, computer operation will be simple, and relatively few people will be employed as programmers. In the near future, the most popular, important, and powerful use of computers will be as a *general-purpose tool.*

The Computer Literacy Deep Freeze

Computers are at a crude, nasty, awkward stage in their development. But they are evolving at an incredible pace. Hardware advances occur almost daily. And software, long the bumbling, dim-witted half brother of computer hardware, has at last entered into its own revolution. In 1976 there was almost no software, yet in 1982 200 companies sold more than a billion dollars of software. By 1990, experts predict that people will be buying $12 billion in software, about as much as they now spend on home appliances. We will soon see more software than ever before, and if we weed through all the junk, we will find much software that is good and quickly getting better.

The twin revolutions in computer hardware and (especially) in computer software will insure that computers of today will be transient, short-lived creatures. Trendy, high-income schools that buy up dozens of these computers and inaugurate intensive courses in the art and science of their programming and operation are handicapping their students. They are guaranteeing that these young people will be victims of technology.

In ten years, how important will it be for a student to know how to program in BASIC, or know machine language, or know how many K bytes are available in RAM storage? Or how to format a floppy? Or how to position a tape cassette to a particular program?

BASIC Will Be a Dead Language

In five years computers will be completely different. In ten years they will be black boxes. They will still be programmable, but nobody except the experts will do the programming. The final custom-fitting of all commercial programs will be done by the user, but in English, not in BASIC, Logo, or Pascal. These will be archaic languages, like Greek or Latin, important historically, but of little relevance to students who are entering the job market of the early 1990s.

High schools, vocational-technical schools, and colleges are turning out huge numbers of computer scientists and technicians. But surprisingly, computer jobs are beginning to dry up, especially at the entry level. High-paying computer jobs are still there, but they are reserved for those who have several years of experience or who have combined skills in computers and in some other field such as business, medicine, law, chemistry, or engineering.

Computer classes in schools today are busy turning out the computer "mechanics" and "repair persons" of the future. Persons trained in these areas will find that there are very few jobs awaiting them, and the competition will be incredible. With the huge supply of bodies and the slackening demand, salaries will plummet and so will prestige. By the time young people enter the market as computer specialists, most of the romantic aura about computers will have rubbed off. The glamour will have faded.

It all boils down to how we see computers. Do we see them as finicky appliances that have to be twiddled, scrutinized, and understood? Do we see them as "exer-cycles" and mental jogging machines that stimulate our problem-solving abilities and encourage algorithmic (that is, step-by-step, logical, goal-oriented) thinking? Or are they mechanical chameleons and quick-change artists?

In the near future I think most of us will see computers as Super Tools — like the handy-dandy Swiss pocketknives you can buy that have all those scissors, bottle openers, screwdrivers and twelve different blades stuffed inside. They will do everything. And we won't care how. We'll just pull out a new tool and run it!

For example, we will pop in a cartridge and our computer will become an electronic typewriter, dictionary, or secretary. We'll pop in another cartridge, and the computer will become our personal accountant, tax advisor, or a gourmet chef.

Computers of the near future will be like vaudeville performers who can change their costumes in a flash. One minute they will be patient math tutors for our children. The next moment, they will be our electronic windows to the outside world. We will use them to bring us the latest stock prices, make a plane reservation, or mass mail our Christmas cards.

Or a moment later the computer will become an interactive (videodisc and graphics) TV. We will get to track down a roller-coaster bomber, solve the mystery of a collapsing bridge, or go on a big game hunt in darkest Africa.

We will not care how the computer changes its clothes. We will not be interested in a tour behind the stage, or what the performer's clothes look like from the inside out. Instead, we will want (maybe demand) to learn, to be informed, to be entertained, and to get on with our work. The computer will slip into its rightful position. It will become a marvelous tool that is almost ignored. It will be an almost invisible means to accomplish the *essential* things in life: survival, work, education, and fun.

Computers As Islands

The approach in many schools is to teach about computers in a special "computer science" or "computer lab" or "computer literacy" course. This reminds me of the touch-typing course and the metalworking and other "shop" courses I took when I was in high school.

In all these "technical skill" courses, kids are introduced to machines and instructed in how to develop a certain level of proficiency and familiarity with these machines. *But they aren't told why.*

At some level, students who take these courses must be asking themselves: Why is a computer important? What good does it do me to know how to program a computer, or load a program, or learn about FOR-NEXT loops?

The computer is not an end in itself. It is a means to an end. It is a resource or a tool that can be used to do something else. Computer skills are meaningless to a child unless the child can use them to do something that he or she wants or needs to do. To make computers meaningful, they must be integrated, on a daily basis, into the rest of the curriculum and into a child's life. The child must need or want to do something important that can be done only on a computer.

Computer Literacy and Beyond

Computers As Moon Rocks

In many schools, desktop computers are introduced as oddities and curiosities, like moon rocks.

This is a marvelous approach. It encourages children to see computers as wondrous devices (which they are) and to approach computers with curiosity and fascination.

Since computers are objects of wonder and curiosity, many schools have put them in a special room — a computer *museum*. Everyone can come in and gawk at them, reverently press their buttons, and say ooh and aah.

But after having a computer about six months, a school usually moves beyond this approach. The awe of and magic about computers quickly wear off — especially for the kids. Teachers begin teaching kids how to program — how to master computers, boss them around, and tame them.

The Latest Audiovisual Device

Many schools are leapfrogging right over these first two steps. When schools acquire a computer today, they don't automatically send it off to a tiny lab or unused classroom. Instead, they regard the computer as a new kind of audiovisual device — a godsend for the frazzled, overworked, but forward-thinking teacher of the 1980s.

This approach is being given a big boost by the educational courseware flooding into the market. Dozens of companies are producing hundreds of software packages. A year ago, there was an acute shortage of reputable software. Now, already, there is a glut. There are hundreds of programs out that introduce kids to the alphabet. Dozens more teach them how to add two numbers or spell simple words.

As I walked down the exhibitors' (read *vendors'*) aisle at a recent educational computing conference, I was overwhelmed by the number of glossy, smart-looking packages I saw. It was a kaleidoscopic, mind-numbing experience.

Given this vast amount of courseware, it won't be long before computers move out of their computer museums and isolated labs and into the curricular mainstream. Thousands of math and language arts teachers already use computers as audiovisual aids. Soon history, science, music, and art teachers will use them too.

The Computer As a General-Purpose Tool

The computer will soon become a valuable resource for teachers, no matter what subject they teach. But the computer can be more than a special-purpose resource to help a teacher teach a particular subject. It is also a tool — a magnificent, general-purpose tool that a child can apply to any subject.

If children learn only how to program, decipher bits from bytes, and learn geography on a computer, they are going to be poorly equipped to use computers — in the future, in a job, in the outside world.

To be prepared for the future, youngsters must learn how to use computers as tools. That's the way most computers in our society are used. And that's the way they will be used in the future.

Discovering a Tool

The problem has been that most classroom computers are regarded more as *toys* than as tools. They don't have the speed, memory capacity, or software to make them serious devices. They are also isolated, one from the next, instead of tied into information and programming resources (by phone or direct-wired access to a central, high-speed computer).

But all this is changing.

One of the most popular and well-attended sessions at the 1983 National Educational Computing Conference (NECC) was on using computers in studying literature and English composition. Teachers presented papers on how they taught word processing in the classroom, how they used a computer in writing class, and how they and their students used a computer to study and analyze literature.

Kids in the first two classes used the computer as a tool — as a *word processor*. They found it was easier to write stories, develop ideas, and explore new subjects by using a computer.

Kids in the third class learned programming skills for a purpose: They turned the computer into a tool to help them analyze a book, article, or short story. They used the computer to complete a class assignment.

Right now, word processing is a very popular computer application in schools. But it is just the tip of the computer-tool iceberg. Computers can become powerful word-handling tools for kids. But they can also become all sorts of other kid tools.

With software already available or under development, computers can become kids' powerful data base managers, priority sorters, homework organizers, and calendar schedulers. They can simulate chemistry labs, physics labs, and math labs. They can be used to map out a complicated dance routine for a musical, compose a song, or take the student on a journey inside a volcano, to the center of an atom, or to the outer reaches of the solar system.

New software packages are also needed that are patterned after the second generation software now running on expensive IBM, Apple, and Xerox business computers. These *kid workstations* should be general-purpose tools that help a student process words, perform complicated calculations, create graphs, functions and diagrams, and organize, classify, and summarize huge amounts of data. They should enable students to link their computers and thereby communicate with each other. They should encourage teachers to assign more *team projects* for students in which students and their computers work together to solve problems, do homework, or complete class assignments.

The more students get to use a computer as a *tool* to enable them to do something necessary or desirable, the more meaningful and useful computers will become. Also, this is precisely the type of training that young people will need to prepare them for their future careers. Very few students will find jobs as computer specialists. But a vast majority of today's students will need to use computers as tools in their jobs. They will use computers to help them solve problems, make decisions, analyze information, have fun, create and disseminate new knowledge, and communicate with their fellow human beings.

Chapter 8

Computer Camp for Grownups and Kids

Catie's Grandmother Goes to Camp

Last summer my mom went to computer camp. The camp, The Computer Tutor, was located in Avalon, a small seashore resort town in southern New Jersey. My daughter Catie was also enrolled in a computer camp — Computer FUNdamentals, at Hollins College in Roanoke, Virginia.

My mom is 58. My daughter is 7. Catie and her grandmother are more than a half a century apart in age. Yet, by coincidence, they were enrolled in computer camp at the same time, learning the same thing. I thought that was neat. And significant.

The Candy Store

Millions of adults find themselves on the outside of the computer revolution and they don't know how to get in. They are like a little kid with his face squashed against the display window of a fantastic candy store. They would love to join the other kids inside the store, but they can't find the door.

Computer camps can be one door into computer literacy and computer intimacy for fearful but interested adults. Full-fledged camps for adults are springing up all over the U.S. Adults enroll in the camps for a period of one or two weeks. Many of the camps combine indoor computer instruction with outdoor exercise.

Adult campers usually begin computing slowly, but they quickly pick up the pace. By the end of camp they spend up to ten hours a day in marathon keyboard sessions. And they get hooked. According to the daughter of one 68-year-old camper, "We had to drag mom away from the machine just to make sure she got nourishment."

Computer Day Camp

Not all computer camps are so intense. My mother's computer camp, The Computer Tutor, was only a day camp. My mother attended the camp for three hours a day for five days. Each night when she finished computing, she returned home to my father.

There were four computers, with two people on each computer. The instructor was a teacher at Avalon's lone elementary school. There was one 9-year-old girl enrolled in the camp and a 14-year-old boy with his father in tow. The rest of the campers were women.

The camp's goal was to teach the basics of computers, including the terminology, a little bit of BASIC programming, and use of the computer keyboard. According to my mother: "Our instructor tried to teach us a lot in a short time. I learned how to do some programming and some graphics. We learned about the disk and about copying and saving."

The Joy of Flying Solo

My mother's most exciting day came when she got to use the computer on her own. "One day," she said, "I worked by myself. That was great! I felt that if I had done that every day I would have gotten a lot more out of the class."

According to my mother, one of the drawbacks of the course was that the instructor had so much to teach and so little time. "You're trying to jam so much into a few hours that it gets very confusing. Having a computer to practice things on would have made things easier."

It was interesting for my mother to watch her classmates' reaction to computers and compare their reactions with her own. "My college secretarial courses on touch-typing helped me a lot," she said. "Being a typist eliminated some of my fear of computers right away. But I was still cautious."

"There were others who were more willing to jump right in," she admitted. "Some of the people got right to work on the computer and tried to invent things right away. Others were quite leery and wouldn't do things until they were taught. This group included me."

Life After Computer Camp

"Computer camp was great," my mother said. "It whetted my appetite for computers. After it was over, I wanted to continue learning more.

"The only problem was I couldn't find any more courses that fit my schedule. I began to worry that I would forget everything I had learned at camp. I decided that the only way to keep learning was to get my own computer. My husband was very supportive. He said he'd get me a low-cost computer as a combined birthday and anniversary gift. He was so proud that I had gone to the camp and that I was learning about computers and having fun."

Grandmother Computer Chic

When my mother got her computer home, at first she couldn't decide where to put it. She ended up putting it on the kitchen table. With its software, manuals, and electric cables, the computer took up almost the whole table.

My mother and father now have to eat their meals on a tiny corner of the table opposite the computer. My mother eats her meals while she is studying the gift certificates she got with her computer. While my father is eating, he just stares at the back of the computer.

My mother is very proud of her new computer. When she first got it, she invited all her friends and family over to see it. The reactions were diverse.

Her grandchildren, Shannon (8) and Laurel (5), were fascinated with the computer's voice synthesizer and the Sesame Street computer games my mother bought. Both grandchildren use computers at school and arrived with the impression that they were "way ahead of Mom Mom" on the computer. But Mom Mom held her own. She got a great deal of pleasure out of being an honorary Whiz Kid.

Mom's friends have had all sorts of reactions. They are proud of her and are amazed that she went out and bought a computer. They are even more amazed that she actually seems to be using it — to write letters using *Bank Street Writer* and to create Christmas-card mailing labels using *PFS:File*.

This year Mom's friends are bragging that they all got Christmas cards addressed by Mom's computer. "I could tell it was a computer that did it," one friend told me, "because it put my last name first on the envelope."

Mom's friends think that she uses her computer all the time. When they call her on the phone, if she sounds vague or distracted, they say, "Libby, are you playing with your computer again?"

Most of Mom's friends are very proud of what she is doing, but they are reluctant to follow in her footsteps. "I know computers are the up and coming thing," one friend told her. "But I could never work one. I'd be lost."

My mother laughs at this kind of reaction. "The biggest thing I have found," she says, "is how easy computers are. I thought they would be much harder."

Then she shrugs and frowns. "It's funny, though," she says. "They are easy, but you still need someone to call on — just for help on the simple things. The most frustrating part is when you are working by yourself and you come to a standstill because a certain button doesn't work. It always worked before, but now it doesn't work, and you feel completely lost and you don't know what to do.

"A person needs someone to call when they feel frustrated — someone to help them, someone to follow up. And half the time advice over the phone is not enough. You need someone looking over your shoulder to see exactly what you're doing. The problem is always something small, but it's enough to stop you in your tracks."

How Grandmothers Get Intimate with Their Computers

According to my mother, "The most exciting time I had with my computer was when I made the mailing labels come out on the printer. 'Look!' I shouted. 'It works! I figured it out myself!'

"Using a computer is like eating Chinese food," my mother told me. "You eat Chinese food, and it tastes great. Then, soon after you are done eating, you are hungry for more.

"Computers are the same way. You do something on it, and it feels great. Then you run out of things to do and you say, 'What can I do now?' It's very frustrating. I have so much fun working on it, but then I finish, and I think I have to invent something new. But to do something new I have to learn more about the computer. It's like I am always hungry for more computing. My appetite keeps growing."

I'm Not Leaving My Computer!

Each winter my mother and father fly south along with the birds. They spend the cold, bitter months in a little resort town tucked away in the Florida keys. My mother flees the Philadelphia area around Thanksgiving, usually after the first cold, soggy November rainstorm.

But this year is different. When Thanksgiving came and went and my mother was still in the Philadelphia area, her friends became puzzled and asked her why she wasn't heading south.

"I'd like to get away from this cold weather," my mother replied. "But I'm not going to leave my computer!"

Mom's Bright Idea

By mid-November my mother had so many manuals, cables, software boxes, and add-ons that she would have needed to rent a U-Haul trailer to get her computer down to Florida.

She checked into renting a station wagon and having my brother drive her computer to Florida. Then she realized that her cottage there was barely large enough for her and my father. She and my father have a loving but stormy relationship. She could imagine what would happen if my father arrived in Florida and found that he had been replaced by a computer.

Computer Camp for Grownups and Kids

What was my mother to do? She really wanted to go to Florida, but she couldn't bear to leave her computer in Pennsylvania. Yet she didn't dare take it with her.

My mother is no dumb bunny. She wanted a computer in Florida, so if her Pennsylvania computer wouldn't fit, the answer was obvious: She needed a new computer — a smallish kind of computer that she could squeeze into the Florida cottage along with her and my father.

Once my mother has a good idea she doesn't waste any time putting it into action. As I write this article, she is busy saving up for a new computer — a portable.

And she's keeping busy on her old computer, using her spreadsheet to chart her stocks and bonds and keeping an inventory of furniture and other household possessions. And she's churning out a snowstorm of letters to me on her word processor.

It All Started with Computer Camp

I've been away from home for seventeen years, yet I have gotten more letters from my mother in the last couple of months than in all those seventeen years combined. And all the letters are chock-full of motherly advice (not to mention nagging).

Each morning I go to the mailbox and find two more letters from my mother — all generated on her infernal (excuse me, Mom) word processor.

I'm beginning to wonder if I did the right thing. Was it really such a good idea to get my mother turned on to computers?

She's beginning to talk about electronic mail and linking us up on a Bulletin Board System. This way she can download ideas and advice anytime she wants, maybe even several times a day. It will be just as if she lived next door.

I'm very proud of Mom. And I'm glad that she went to computer camp. The only thing that worries me is where is this going to end?

My advice to those of you out there with mothers and grandmothers is simple: Once you start them computing, watch out!

New Directions for Computer Camps

"I thought that this camp would be about programming. I didn't know that it would be so much fun!"

Ashley Bell, age 8

Ashley was one of the youngest campers at the Computer FUNdamentals camp at Hollins College, in Roanoke, Virginia, last summer. Her comments reflect the kind of computer activities she participated in at the camp. However, if she had gone to another camp, she might have learned about computers in a completely different way.

The Changing Face of Computer Camps

Most educators agree that the first computer camp was organized by Dr. Michael Zabinski in Connecticut, in 1978. Now, six years later, Zabinski's organization offers five camps annually, in locations from Simsbury, Connecticut, to Portland, Oregon. In addition to Zabinski's camps there are hundreds of other computer camps throughout the U.S.

The first camps were mostly attended by boys. The boys studied "hard-core" computer subjects like BASIC programming, computer hardware, and hooking up different devices to computers. Compared to today's models, the computers at the first camps were primitive. They consisted of early Apple computers, TRS-80 (Model I's), Commodore PETs, and other computers whose names we have all but forgotten.

Today's campers enter a new world filled with the latest personal computers and peripheral devices such as speech synthesizers, graphics pads, light pens, and robots. They study a variety of subjects, including the impact of computers on society, computers for handicapped people, and computers in the arts and humanities.

Today, girls represent a much larger proportion of the campers. In some camps, they number as many as a third.

At most camps you will also see a few campers who have some sort of mental or physical disability. Campers in wheelchairs are a common sight at many camps.

So are adults. The newest computer camps cater to both youngsters and oldsters. In fact, it's predicted that many of the most avid campers in 1984 will be men and women in their 60s and 70s.

How to Choose a Computer Camp

There are hundreds of computer camps to choose from, each with its own philosophy and personality. And you can find the right one for you, if you look hard enough.

The first thing you should look at is the type of camp. Is it sponsored locally or nationally? Is it for children, adults, or both? Do the counselors concentrate on programming or on computer literacy and applications? Is the camp residential or a day camp?

There are benefits and drawbacks associated with each type of camp. For example, if a camp is locally sponsored, it may be more suited to the needs of the people in your community. But local sponsorship doesn't necessarily mean high-quality sponsorship. Generally speaking, the best local computer camps are affiliated with a community college or university.

Residential computer camps are nice because they take the children away from home for a week or two of fun, physical exercise, and computer instruction. But some educators feel that residential camps are a fad. Their outdoor activities are often an afterthought, and the camps cannot compare, in terms of staff, program, or facilities, to the regular summer camps, which, on their own, are beginning to offer computer activities. Also, residential camps are expensive and relatively inefficient if your main goal is to introduce your child to computers.

In the past, most computer camps were for kids. Now adult camps are springing up all over the country.

Many families send their kids to computer camp so they can come back and tell the family which computer to buy. But why let your kids have all the fun? Why not attend computer camp at the same time as your son or daughter? Then you and your kids can decide together which computer is right for the family.

New "mixed-age" camp classes are springing up that include people of all ages. Being in a class with several bright youngsters can be unnerving, but it can also add a new dimension to your computing. Kids approach computers as *explorers*. By imitating them you can begin computing fearlessly and playfully.

The Need for Continuing Support

The best computer camps offer a balanced approach — some computer programming and some computer activities. But beware. If you get your child started in either side of computing, his or her appetite for more computing is liable to increase. When you look for a computer camp, you should try to find one that will be around to satisfy your and your child's computing interests no matter how sophisticated they become.

Dr. Zabinski, for example, believes that computer camps "breed kids who are sophisticated with computers, so they can't just drop them." His camps emphasize programming as opposed to computer activities. "We train the youngsters in computers, so it is our responsibility to be around when they become more sophisticated and need more advanced training."

Zabinski's philosophy is "to motivate kids and excite them with examples they can relate to and identify with." His camps have been so popular and successful that he and his staff have to revamp their curriculum each year just to keep up with the kids they trained the previous year.

According to Zabinski, "We used to be content teaching kids to program in BASIC and Pascal. Now I feel that teaching new programming languages is just moving sideways. We can't afford to move sideways. Kids can master new languages in just a couple of weeks. Our objective in 1984 is to teach kids how to interface computers with each other and how to interface computers with other machines. We'll teach kids how to create their own computer languages, and how to use modems and bulletin boards and get computers communicating over the telephone."

Zabinski emphasizes that his highly technical curriculum is not aimed at just teenagers and older children. "Take nine-year-olds," he says. "Nine and ten-year-olds are not what they used to be. We have one nine-year-old who learned Assembler and won a national Assembler Language contest on the TRS-80 computer.

"There are plenty of sophisticated kids at all ages," contends Zabinski. "Computer camps are often these kids' only outlet. We've helped to create these kids, so we have to be ready when they come back to us each year. We can't abandon them."

Computer FUNdamentals

Nancy Healy and Dr. Barbara Kurshan run the Computer FUNdamentals Camp at Hollins College. Kurshan and Healy agree with Zabinski that computer camps need to keep upgrading their curriculums to keep up with the newest computers and the increasing sophistication of the average camper. But Kurshan and Healy stress computer applications as opposed to computer programming. And, above all, they want their campers to have fun.

According to Healy, "What makes our camp different is that it is oriented toward fun, and, at the same time, the kids become good computer users. Also, we don't mix physical activities and computer instruction. This lets our handicapped campers do everything that all the other kids do.

"Another reason our camp is different," Healy continues, "is that our camp isn't just for math and science freaks. Kids who love music and the arts are equally interested and involved.

"After the first few days at camp, it is easy to see who knows what. The 'knowers' are those who attract people around them. But the great thing is that each child brings a different skill with him, like typing, music, art, programming, or math. The kids work together and draw on each other's skills and interests. That way everybody gets a chance to shine."

The Computer That Ate Manhattan

Like their counterparts at other camps, computer campers at Hollins spent most of their time last summer using real computers as electronic notebooks, typewriters, telephones, libraries, and mailboxes. But camp counselors also encouraged the children to spend time inventing totally new fantasy computers. Children described these computers and what things they could do. One boy, for example, made up a story about a computer that ate Manhattan.

One of the big projects during the camp was for the children to build their own *junk computers*. The children designed and built the junk computers out of all kinds of things, including buttons, wires, beads, tupperware, TV sets, and aluminum foil. One boy built a computer out of a nonworking TV set and a working walkie-talkie. The boy hid the walkie-talkie inside the TV set. Another boy built a junk computer that played beach music. The cardboard computer had a tape recorder hidden inside.

A local elementary school PTA in Roanoke sent two children to the camp on scholarships. The children were to learn as much as possible about computers during camp so they could help their teachers use the school's two new computers the following fall. The children, one 10 and the other 11, were chosen on the basis of an essay on why they wanted to go to computer camp. They wrote down everything they learned at camp in a spiral notebook, and were among the camp's most conscientious students.

Training a Future Sally Ride

While the camp was in progress at Hollins, America was glued to the TV set watching its first female astronaut, Sally Ride, blast off the earth in the space shuttle. This inspired the kids to create a computer-controlled rocket launching at camp.

The rocket was finally launched on the same day that Ride and her teammates brought the real shuttle back to the earth. It even featured a computer-screen simulation of the rocket taking off and a speech synthesizer, in robot nasal monotone, doing the countdown: 5 ... 4 ... 3 ... 2 ... 1 ... IGNITION!

In honor of Sally Ride, the girl campers got to operate the computer to control the rocket launch. And the local TV station in Roanoke was so excited by this project that they filmed the rocket launch and, on the evening news, mixed the tape with a film of the real space shuttle take-off.

Chapter 9

Children's Computer Languages

A Computer Language for Kids

I am fascinated with the idea of a computer friend for kids. Later in this book I show you how to build a simple Friend program. When the friend program is run, the friend's face appears on the TV screen. At first, the friend is asleep. "Ding! Dong!" goes a bell. The friend wakes up and winks. "I'm Ged," he announces. "Who's out there?"

The child answers by typing in her name. The friend greets the child and asks if she'd like to play a game. If she would, the friend gives her a menu of the games in its repertoire.

If the child is using a disk-based system, the friend starts the game automatically. If the child is using a cassette-based system, the friend helps the child load the game program from tape.

After the game, the friend comes back on the TV screen. "I hope you had fun," it says. It offers to play a new game with the child.

Friendly Computing

The friend is like a simple operating system. It is the interface, the middleman, between your child and the computer. It is a first attempt at making computers warmer, more human and personable.

The computer friends should liven up your child's computing. But they won't help with *programming*.

No matter how many games a friend has up its sleeve, the child is never actually programming the computer. He or she is interfacing with the friend and its programs. But not programming. Instead, in a way, the friend is programming the child.

This is one of the major drawbacks of the computer friends. They don't encourage children to write programs of their own. At least half of the value of the computer is unleashed when you program it yourself. Without that opportunity, your child is missing out on a lot.

Right now the friend is a friendly operating system. What we need is a friend that can also act as a *friendly computer language*. Then the friend can encourage the child to create, save, and run programs.

Children's Computer Languages

Beyond Logo

"Wait a second!" you say. "What about BASIC, PILOT, and Logo? These languages are easy to learn. They are friendly. They are perfect for kids."

My answer to that is: Do you have kids of your own? Do you teach kids? Have you ever tried to teach little kids how to use BASIC? Or PILOT? Or Logo?

I have two kids. My daughter Catie is almost seven. She's a first grader. My son Eric is three and a half. Eric spends his mornings at "Miss Eleven's Castle" (Evelyn's Day Care).

Both kids are whizzes at using the family computers. They have their own disks and tapes. They can turn the computers on and off, boot up disks, run programs, and key in the letters, numbers, and words the programs request. Both kids know all the special-function keys on the computer keyboard.

But try getting them to program? Forget it.

I can understand Eric's reluctance to program. After all, the kid doesn't even know how to read or write. If he gets 6's and 9's mixed up, and M's and W's, how can I expect him to master FOR-NEXT loops, string variables, subroutines, and arrays?

But Catie is a different matter. She reads Nancy Drew mysteries and Choose Your Own Adventure books. She is good at arithmetic, and she loves logic games, puzzles, and mazes. But she has no interest in programming.

Maybe it's just getting over the first hurdle. Unfortunately, Catie and I have stuck on that hurdle for over two years.

The first hurdle is the first line of code in a program.

That first line is invariably a FOR-NEXT loop. The FOR-NEXT loop might do different things. It might print the message CATIE LOVES MOWIE a thousand times, all over the TV screen. (Mowie is Catie's kitty.) It might make the sound of a police siren or a dropping bomb, or the noise of water, or of crashing dishes. Or it might draw a drunken fly wandering across the screen.

What is Catie's reaction to all this? It's not positive, I'll tell you that.

Even if I get the fly to change into 16 different colors, Catie couldn't care less. After the first line of code, her reaction is sudden and dramatic. She gets hungry. Or she has to go to the bathroom. Or she has a headache. Or her spine dissolves and I get to watch her slide out of her seat and collapse into a puddle on the floor.

Or else she begins giggling and acts silly. She begins typing on the computer with her nose. Or her tongue.

Children's Computer Languages

This is an embarrassing situation.

On all sides we hear about friendly computers, computer literacy for kids, teaching kids to speak "computer" along with English. And here I am, a computer expert, a writer, an advocate for teaching computing to kids. So what do I do? I try to drag my kids into the computer age.

But they don't want to go.

Computer Literacy for Whom?

It's not so much that my kids resist me actively. It's just that they don't see the point. They have too much itching powder in their pants to make them sit still long enough to program.

At least using the languages available now.

But what if we created programming languages that incorporated the same ingredients as the best software designed for children? What are these ingredients? Quick response, for one. Other ingredients include action, sound effects, pictures, colors. Quick mastery, a sense of power and control. Progress. Encouragement. Humor. These are qualities found in all good software for kids. But these qualities are not evident in programming languages. Even in PILOT. Even in Logo.

Children's Languages

Children's computer languages should contain the same qualities that make good programs so popular with kids. Maybe the programming will be in terms of colors, or sounds. Maybe in terms of shapes.

However it's done, the kids should be able to create programs themselves. They should be able to save, retrieve, and run those programs. The programs should not be trivial. They should do something. (Of course, they are doing *some*thing if they are teaching a child how to program.)

Most of all, the programming language should be fun for the kids to use. It should teach the kids that programming isn't something ugly that you have to do to get something nice. It's fun in itself. It's a way to express yourself, like coloring or playing music, or dancing.

The language shouldn't deter kids. It should encourage them to sit down and write a whole program. Even a short program.

Turning Logo Upside Down

In the last article I put out a call for new computer languages for kids. I related my experiences with my own children (ages three and seven), trying to teach them to program. Frankly, I admitted that my efforts had failed. Neither child seemed to have the least inclination to learn how to program.

Enter *Delta Drawing*

Recently I received a copy of Spinnaker Software Company's *Delta Drawing* program for the Apple II Plus. (Versions for other computers are also available.) At first, I thought *Delta Drawing* was just another new paint program or simplified turtle graphics program. Then I dug further.

Now, my family and I have spent a dozen hours playing with *Delta Drawing*, and I am convinced that it is something more. I now believe that it is a first, but significant, step toward a powerful new computer language for kids.

I Want to Play Mystery House!

This is how my seven-year-old daughter, Catie, and I were first introduced to *Delta Drawing*.

I took the plastic off the *Delta Drawing* case and handed the disk to Catie. Catie put the disk in the Apple computer's disk drive and booted up the program.

We answered a couple of quick questions: Did we have a color monitor? Yes — a must. Did we have a printer? No. Then a little upside-down V appeared on the center of the screen. Under the V was a blinking dot. Around the edge of the screen was a blue box.

The manual calls the upside-down V the "Delta Drawing cursor." But Catie knew better. "That's a turtle," she said. "The blinking dot is her tail." Catie named the turtle DeeDee (for DD — *Delta Drawing*).

I thought things were going pretty well. That's when Catie got bored. (Catie gets bored easily.)

"I'm tired of this game," she said.

"But we haven't even started," I replied.

"I don't care. I want to play *Mystery House.*"

Mystery House (from On-Line Systems) is one of Catie's favorite adventure games (along with *Cranston Manor,* also from On-Line, and *Copts and Robbers,* from Sirius).

Did I respond to Catie's obstinance with tact and gentle persuasion? Of course not. I did what any normal parent would do. I yelled at her.

Naturally, she responded by crying and I felt guilty. You really blotched it, Fred, I thought to myself.

Catie started banging on the Apple keyboard. All of a sudden, DeeDee came to life and drew a small straight line toward the top of the screen. She made a cute little "blink" or "clink" sound. Catie stopped crying and stared.

"Can I make my own pictures?" she asked.

Encouraged, I pulled out the Fast Start cards that accompany the *Delta Drawing* manual. Catie and I both avoided the manual. At 80 pages of fairly tiny print, it looked too intimidating, both for daughter and daddy.

But the Fast Start cards were different. Each one is made of shiny, plastic-coated, heavy-stock paper. Each one has a picture at the top and a few commands to show you how the picture was made. The cards are two-sided, numbered (with big numbers) from one to fourteen.

One card tells you on one side how to load *Delta Drawing,* and on the back gives you all of the *Delta Drawing* commands. I discovered that Catie had pressed the Apple's D key and caused DeeDee to *draw.*

Catie and I looked at the pictures on the Fast Start cards. We grew excited. It looked like we could get DeeDee to draw all the standard stuff: circles, squares, triangles, and the like. We could also get DeeDee to draw three-dimensional cubes and "paint" the sides, play a game of tic-tac-toe, and draw a colorful scene with an orange house, green grass, a blue sky, and an orange sun.

But how?

I did the adult thing and turned to card #1 and began trying to decipher the command beneath the picture. Meanwhile, Catie did the kid thing and began punching buttons. A couple of minutes later, I was still on card #1, but Catie had discovered that the R button made DeeDee turn right 30 degrees, and the L button made her turn left 30 degrees. Catie showed me how she could press the U button and make DeeDee do a fancy U turn; and the M button to get DeeDee to shoot across the screen with her pen up — that is, she moved without drawing.

Catie squealed. "DeeDee didn't obey me," she said. "She went up and I wanted her to go down." We looked at the card with the command summary. We discovered that by pressing the E button we could make DeeDee backtrack and erase her last step.

I threw the Fast Start cards on the table. From that point on, we began improvising. Occasionally, we picked up the cards and borrowed commands from them, when we needed to make DeeDee do something we wanted her to do.

It sounds really impressive when I say that "Catie and I improvised." Actually, I advised Catie what to do, and she ignored me. This seemed to be a very successful strategy to learn *Delta Drawing*.

After a while, I gave up and let Catie take the lead.

Catie's approach was to do things with commands she already knew. For example, the first thing she did was hit the D (Draw) key until DeeDee drew herself off the top of the screen and popped back onto the bottom of the screen.

Catie wondered why DeeDee could "tunnel" off the screen, like Ms. Pac-Man. Why didn't DeeDee bump her nose on the blue wall at the edge of the screen?

We looked at the Fast Start card with the command summary and found that there is a B command that makes DeeDee "bounce" instead of tunnel. When we pressed the B key, the wall turned green. To get DeeDee back into tunneling mode, we had to press a W (Wraparound) key.

Watching Catie at work was like watching a baby learn to speak for the first time — only in fast motion. She was learning a new language, and the moment she learned a new word in the language, she used it to express herself.

In fact, the reason she learned new words was to be able to express herself. She was motivated to master the language's vocabulary so she could do what she set out to do. And when she learned each new word, she automatically incorporated it into all the other words she knew. She was associating each word and developing word sequences — her own personal grammar in the new language.

Catie's experimental approach was efficient, but it also led us occasionally into dead ends and surprises. For example, Catie got DeeDee to draw a house out of a square and a triangle. She filled the house with purple, by pressing C to choose the color, then by holding the CTRL button down and typing F (Fill). Then she tried to color the grass green. She pressed C and picked green. She typed

CTRL-F, and green started washing like a wave across the screen. Then the green went out of control. It slipped through a tiny hole in the line that separated the grass and the sky, and it filled the sky, too. It ended up filling the entire picture, except for the purple house.

Catie howled!

A Sun Not a Circle

One thing that I immediately liked about *Delta Drawing* was the quick way Catie could make a picture. Also, I liked the precise, geometric way she constructed pictures. I have a problem with paint programs that use joysticks because my fine motor skills never passed the klutz stage. But, looking at Catie create pictures in *Delta Drawing,* I had hopes that even I might be able to make something pretty.

That is, if Catie would ever give me a turn.

Another thing that pleased me about *Delta Drawing* is that when children are using shapes like triangles, squares, and circles, they are not dealing with them in an abstract, adult sense. After all, shapes by themselves are boring. But shapes that resemble real-world objects like hats, planets, boxes, mountains, etc., are interesting. And shapes that can be combined into building blocks to make a new world are even more interesting. With *Delta Drawing,* Catie not only combined the shapes, she created new shapes to act as the proper building blocks for the world she was trying to create.

The Catie Robot Makes a Circle

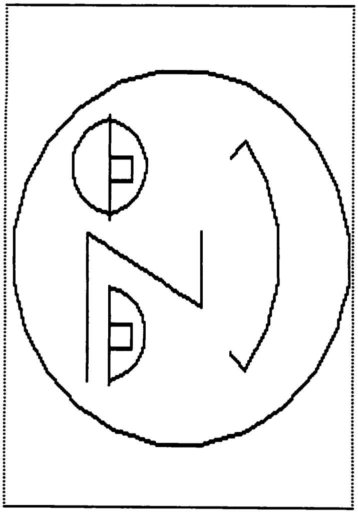
With both Catie and Eric (three years old), I had tried the classic Logo experiment where you get the child to play "turtle" and figure out how to walk in a circle around the floor. I had often tried, but I had always failed.

The easy part was getting Catie and Eric to play turtle and figure out how to draw a circle. They took a baby step forward, then made a small turn to the right. Then they repeated the two steps over and over until they made a circle.

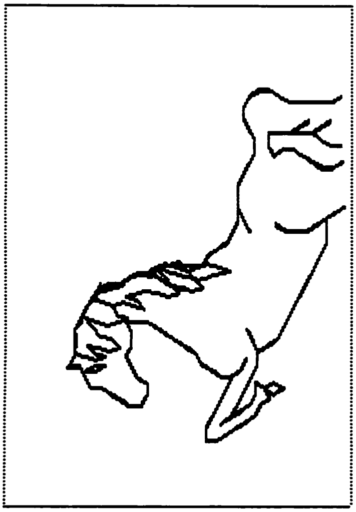
Fine. But then came the hard part — entering a program into the computer to make the turtle do what Catie and Eric had discovered so easily.

This is where I hit a brick wall. Catie and Eric had no interest in creating a circle procedure (program) in Logo. In fact, they never got past the first command — FORWARD or FD.

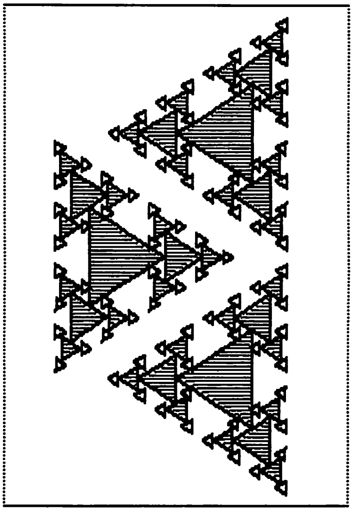
For Catie and Eric, it was too much effort for too little reward. And they didn't want to wait for the computer to learn the procedure. Why couldn't the computer obey them and make the circle immediately?



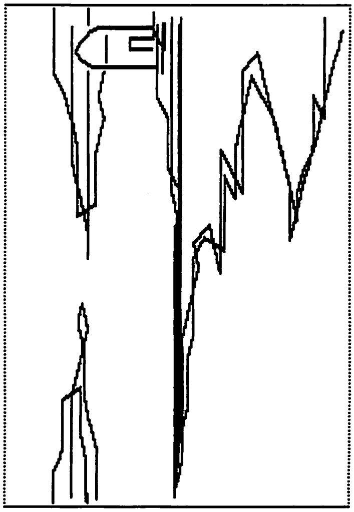
Ziggie by Dennis Purcell



Pegasus by 12-year-old computer summer camper



Triangle experiment by Jock Gill



Seascape by Clifford Wong

With Logo it couldn't, but with *Delta Drawing* it could. To make a circle, Catie typed D (Draw) and R (Right), then D and R, then D and R again. As she typed, DeeDee responded and drew the circle. It was easy to type D and R, and Catie got immediate results. After drawing the circle for the first time, she colored it orange with just a single command — CTRL-F. Then, on her own, she figured out how to make DeeDee move around the edge of the circle and make rays. She had turned her circle into a sun.

Our First *Delta Drawing* Program

Catie and I were doing great — until I accidentally bumped the 1 button on the keyboard and Catie's beautiful picture disappeared.

I thought she was going to kill me.

Quickly, I scanned the card with the command summary, looking for an "Unerase" button. "Why did that happen?" I grumbled. "Stupid program!"

I didn't find an unerase command on the card, but I did discover the T (Text) command. I pushed the T button. DeeDee vanished. The screen filled with words.

There was Catie's program! It was still there.

I flipped through the big manual. A moment later I realized what I had done. By accident, when I pressed the 1 button, I had saved Catie's picture as a *program* — program #1. To get the picture back, I had to call the program. Doing that was unbelievably easy. I just had to push the 1 button again.

We switched back to DeeDee by pressing the G (Graphics) button. Then Catie pushed the 1 button, and, superfast, DeeDee drew and painted her picture. It was good as new.

Automatic Pictures

That's when Catie and I discovered the A button. The A button executes the Automatic command. The Automatic command automatically calls the last saved program and obeys it — over and over until you punch the ESC (escape) button.

I had a brainstorm. I had Catie type in a CTRL-D (a half-draw) and a CTRL-R (a half-turn right). DeeDee did her stuff.

Then I told Catie to press the 1 button. She did. Now we had a program that, when we pressed T, looked like this:

```
1<...^D..^R...>1
```

The program looked puny and not very exciting. What was it good for? To find out, I asked Catie how we could make DeeDee automatically obey program #1 — over and over. With just a moment's thought, Catie pressed the A button.

A couple of seconds later, we had a circle.

To get DeeDee to stop drawing, Catie pressed the ESC button. Catie and I were excited: Using program #1 as a building block, we had created a circle program — program #2. We saved program #2 by pressing the 2 button.

We drew circles all over the picture screen and got DeeDee to paint them different colors. Then we tried something simpler — and neater.

We erased all our current commands by typing CTRL-E. We pressed the T button. Our first two programs were still intact.

We pressed 2 and got a quick circle. Then we pressed the L button. DeeDee turned 30 degrees to the left. We saved these two commands as program #3. The first three programs looked like this:

```
Program #1   1<...^D..^R...>1
Program #2   2<...25<1>...>2
  This came from pressing the A Button.
Program #3   3<...<2>..L...>3
```

When you see a number inside brackets, like <2>, it means you are calling a program — this time program #2. The 25 <1> means you are calling program #1 25 times. We got the computer to do this just by pressing the A button — once — and the ESC button to stop DeeDee.

Next Catie typed the A button. DeeDee drew a circle, turned left 30 degrees, drew another circle, turned left 30 degrees, drew another circle, and so on. In about a minute she had rotated her way around the picture screen. She had drawn a three-dimensional figure: a doughnut. Catie, on her own, moved DeeDee and had her paint the doughnut's center orange and the background violet. *After the doughnut was drawn* (not before), Catie and I pressed T to see what the program looked like. Here it is:

```
Program #4   4<...44<3>..2L..M..C:2
             ..^F..8M..C:5..^F..
```

The most formidable command—44 "calls" of program #3—was achieved by pressing the A button *once*.

Catie and I saved our doughnut in just a few seconds. We pressed CTRL-S, and the computer asked if we wanted to save or recall (load) something. We pressed S for Save. Then the computer asked us if we wanted to save the program (T — text) or the picture (G — graphics). We typed T. The computer told us to load in our own disk.

We did, and it asked us to name our *Delta Drawing* file — we called it DONUT. Then the computer saved it.

Nested Building Blocks

Delta Drawing's real power comes from its ability to save pictures as building blocks, from its ability to combine simple building blocks into blocks that are more and more elaborate and complex. And you can gain access to all these building blocks just by pressing the CTRL-A buttons. When you press CTRL-A, the computer asks you which building block (program) you want. You can choose any number from 1 to 9.

By pushing just a few buttons, Catie and I created our doughnut. We built the doughnut from a draw and turn program, a circle program, and a circle and turn program. Just as easily, we could have created house programs, people programs, tree programs, and so on. We could have formed a picture by positioning DeeDee on the screen and calling the program we wanted.

A Kids' Language

Delta Drawing is, I hope, one of the early representatives of a new generation of children's software that combines simplicity with great power. Also, it is open-ended. It is a language. Once the child learns the language, she can do whatever she wants. And she can do a lot *even as she is learning the language*.

This simplicity, power, and freedom are what made the program a hit with Catie. And when her brother showed up, and learned a few buttons, they made a big hit with him, too. All of a sudden programming becomes an activity with immediate results that are meaningful to the child, controlled by the child, and that challenge and stimulate the child to be original and inventive.

This is certainly a good start toward a kids' language of the future. And it's not a bad adults' language, either. My wife Janet and I have had a ball creating pictures with *Delta Drawing*. Here is a piece of software that is equally fascinating and easy to use for a three-year-old, a first grader, and two jaded adults. The generation gap between the different members of the family disappears when we use *Delta Drawing*. We are all equally caught up in exploring its possibilities. And no single member of the family seems to have an edge. This is a very nice feature of the program.

Upside-Down Logo

Why did I claim that *Delta Drawing* is like Logo turned on its head? Because with Logo (and most other languages), you have to type in the commands in your program before you can run the program and create a picture. With *Delta Drawing* you make the picture first, and in making the picture you create a program. It's just the opposite. It's Logo upside down!

Delta Drawing costs $59.95 and runs on the Apple II+, IIe, Atari 400/800/800XL/1200XL, Commodore 64 and IBM PC and PCjr.

To inquire about *Delta Drawing*, write:

Spinnaker Software Corporation
215 First Street
Cambridge, MA 02142
617-868-4700

Computer Camp Resources

The Computer Camp Book

The Computer Camp Book is published by The Yellow Springs Computer Camp, Inc. It has 224 pages and costs $12.95. To order the book, write:

The Computer Camp Book
1424 Glen View Drive
Yellow Springs, OH 45387

or call 513-767-7717.

The book is a wealth of information about computer camps, including:

- How to start and run your own computer camp
- Ideas and materials for teaching and learning
- Computer literacy activities
- A look at different computer camps
- A nationwide guide to computer camps, courses, and workshops
- A guide to computing resources (including a five-page guide to computing resources for handicapped people)

Parent's Guide to Computers in Education

The *Parent's Guide* was written by David Moursund. It is a real buy — 80 pages for only $3.50. To get the *Guide*, write:

Parent's Guide
Dept. of Computer & Information Science
University of Oregon
Eugene, OR 97403

or call 583-686-4429.

The book covers a lot of ground in a clear, simple style. Some of the subjects covered include: the school of the future, introduction to computers, hardware and software, computers in education, "What You Can Do," a buyer's plan, a glossary, and a list of resources.

A unique feature of the book is that it is really two books in one. A second book, entitled "Here Comes the Dawn (If Only I Can Find the Switch)," written by Merle Marsh, appears in little italicized text boxes at the foot of every few pages. It begins, on page 3: "I tried to enter the Computer Age by quietly sneaking up on the new technology . . . ."

Two of the leading computer camps in the U.S. are the Atari Computer Camps and the National Computer Camps. You can learn more about them by writing:

Dr. Linda Gordon
Atari Computer Camps
Dept. AL
40 E. 34th Street
New York, NY 10012

Dr. Michael Zabinski
Director
National Computer Camps
P.O. Box 585
Orange, CT 06477

You can learn more about the Hollins College Computer FUNdamentals Camp by writing:

Dr. Barbara Kurshan
Nancy Healy
Computer Fundamentals Camp
Hollins College
Hollins, VA 24020

To find out more about the Hollins camp's robot mascot, you can write:

Bill Glass
Tasman Turtle & Turtle Tot
Harvard Associates, Inc.
260 Beacon Street
Somerville, MA 02143

Chapter 10

Projects for Kids

Computer Adventures

This chapter is for everyone — big kids, little kids, fat kids, skinny kids, and kids with green polka dots on their feet. Who am I? I'm a computer nut. I love computers — even more than pizza, steak sandwiches, or spaghetti.

Once, several years ago, my wife bopped me on the head with a pillow because she was jealous of my computer. Actually it wasn't a computer, it was a computer terminal that talked on a telephone with a big computer thousands of miles away. The terminal was neat because it fit into a briefcase. I used to stay up all night programming the computer on the kitchen table. Lots of nights I got so carried away I forgot to go to bed.

And computer fantasies? Wow! I used to dream of computers even before I ever met one. But now, at long last, my dreams have come true. I have a house full of computers — computers in the bedroom, in the dining room, in my study, and in the playroom. This morning I took a computer with me into the bathroom. Yesterday I was really proud because I got six computers running programs, all at the same time. One was playing music, two were talking, one was spitting out words, one was drawing crazy turtle pictures all over the TV screen, and one was controlling a robot.

A Robot Pterodactyl

I have written 15 books about computers and robots, including *Katie and the Computer, Working Robots,* the *Star Wars Question and Answer Book About Computers,* and a new solve-it-yourself mystery series. The first book in the series is *Chip Mitchell: The Case of the Stolen Computer Brains*.

I really envy the hero of the series, Chip Mitchell. Chip has even more computers than I have. And he has a talking pet robot named Sherwin, 456 snakes, monkeys, lizards, wallaroos, gerbils, and frogs, all living in his bedroom. Of course Chip doesn't have something I have: a pterodactyl robot that sleeps under my desk and nips my knees while I'm writing.

We Were Kids Together

Computers and I go way back. I was born on January 6, 1949, at Bryn Mawr Hospital, just outside Philadelphia. As I was lying in the nursery, crying, slobbering, and gooing along with a dozen other babies, I didn't realize there was another baby just up the road that was already beginning to make history. That baby didn't slobber or cry. With its 20,000 glowing vacuum tubes the size of pickles, it was hot stuff. Who was the baby? It was ENIAC, the world's first modern computer.

As the years went by, computers and I grew up. We were neighbors, but we still hadn't met. I grew bigger, like a normal kid. But while I was growing big, computers grew enormous. By 1959, when I was ten years old, some computers were bigger than a brontosaurus. One computer, the Whirlwind I, in Massachusetts, occupied a two-story building the size of a city block.

During the 1960s I still hadn't heard about computers. But I was getting ready for them. I had a bedroom in the attic of an old stone house. Off the corner of the bedroom was my Monster's Den. It was really a giant closet, but I called it my Monster's Den because that's where I kept all the creatures I had invented — clay aliens, plastic dinosaurs, and robots made out of old go-cart motors, motorcycle batteries, and dryer hoses.

I read comic books, saw monster movies, and stayed up late for the scary movies that came on TV after midnight.

At night I dreamt about monsters, and about building a machine someday that would obey my orders — just like Dr. Frankenstein's creature. Except not like Dr. Frankenstein's exactly, since his creature didn't listen too well.

Little did I know that I was dreaming about computers. And while I was dreaming, computers were undergoing an almost magical change. They were like Alice in Wonderland. At first, they had grown really big. Then one day, after eating a piece of cake, they grew very, very small.

What happened?

In California, an engineer named Ted Hoff did the impossible. He squeezed hundreds of computer circuits onto a tiny square of silicon the size of a baby's big toenail. Ted had invented the world's first chip "brain." He had started the personal computer revolution.

This was about the time I first bumped into a computer. I met the computer at college. I taught it to tell me how many Chinese were visiting Latin America. At the time, I thought this was an important subject. The computer added up the number of Chinese visitors and helped me make lots of pretty graphs to impress my professors.

Dreams to Magic Words

I was hooked. When I programmed the computer, I felt like a lion tamer in a circus. With just a few commands, typed into a typewriter, I was able to control a machine that was bigger than my college dorm, and faster than greased lightning.

I especially liked computers because I could control them with words instead of with screwdrivers, wrenches, pulleys, and gears. It made me feel like a magician. If I said the magic word, the computer would turn cartwheels. Of course, sometimes I said the wrong magic word, and the computer stood on its head or rolled over and played dead.

In the 1970s, I went to work for a company programming computers full-time. I worked on a computer in the basement of the Pentagon, in Washington, D.C. Just to get into the computer room each day I had to wear five security badges hanging from my neck like dog tags. My job was so top secret I never even got to see the real data for my programs. All I knew was that my programs had something to do with sending soldiers' socks and underwear all over the world.

The basement of the Pentagon was a gloomy place. Everyone frowned and carried rifles. Pipes hung from the ceilings. After only a couple of weeks there, I devised a plan to escape. One day, I took my briefcase computer terminal, slipped out the back door, and started visiting elementary schools around Washington.

When I arrived in a classroom, I plugged the terminal into a telephone, and the kids and I called up my company's big computer in Chicago. We bought Bob Albrecht's book *What Do You Do After You Hit Return?*, and we used it to teach us how to program in BASIC.

It was a great thrill. Here we were in a school classroom playing with a company's giant computer. I was supposed to be working. The kids were supposed to be working. Instead we were playing games like *Mugwump, Hurkle,* and *Hunt the Wumpus*. It was a nice break from programming socks and underwear.

My company didn't let me stay in the classrooms too long. They decided to use the top-secret military programming skills I had developed at the Pentagon and put me to work programming Bicentennial military bases. I wrote a program that automatically produced a letter to Congressmen congratulating them on having a new Bicentennial military base in their district. When I finally got the program debugged, it was one of my proudest moments in top-secret military programming.

A Bonk on the Nose

Soon after this success, my wife and I moved to North Carolina. We had a baby daughter named Catie. Shortly after Catie was born, I had her sitting at the computer in my study, banging happily on the keys. We used to spend hours in this creative form of free play. One night, however, all of a sudden, Catie lost her balance, fell forward, and bonked her nose on the picture screen.

Oh, how she howled!

My wife came running and yelled at me for almost letting Catie fall into the computer. This was such a good idea for a book that I immediately sat down to write it. Two years and 99 rejection letters later, the book was published. It was called *Katie and the Computer.* It was about a little girl named Katie who fell inside her family's computer and found a magic land known as Cybernia. Katie's adventures inside the computer included riding down a mountain on a bobsled, meeting a ferocious robot spider, parachuting out of an airplane, and getting fired out of a pirate's cannon in the midst of a big glob of goopy green paint.

Your Wish Is My Command

What can a personal computer do? Nothing — unless you give it orders. The computer may have millions of transistors and be as swift as a bolt of lightning. But it is nothing more than a servant. And you are its king or queen. Like a good king or queen, you want to put your servant to work. But how do you give it orders?

The first computers understood only two numbers — ones and zeros. The ones and zeros that humans fed the old computers represented the pulses of electricity that whizzed through the computers. The ones represented the big pulses. The zeros represented the little pulses. Dozens of ones and zeros, strung together like pearls on a necklace, represented only a single computer command — like ADD 1 PLUS 1.

Today's personal computers can almost understand English. You feed them commands by pressing buttons on a keyboard wired to a chip. If you took off the top of the keyboard, you would see dozens of little chips inside. The chips might be lined up on green plastic cards the size of graham crackers, or hidden inside black plastic cartridges.

Let's say you type the command PRINT "HI" into the computer. First, the computer translates your command into tiny pulses of electricity. Next, it obeys the command. Then it translates the answer back into English and prints it out on the TV screen. "HI" says the computer.

Writing Simple Programs

Now let's imagine that you want to do your math homework on a computer. As part of your homework, you have to multiply pairs of numbers. Your computer is great at multiplication. But you have to teach it how. You have to give it orders.

You think about what you want the computer to do. First, you want it to accept two numbers. Then you want it to multiply those two numbers. Last, you want it to print the answer so you can use it in your homework.

You sit down at the computer keyboard. You have to teach the computer with a language the computer understands. Your computer talks BASIC, like most other small computers.

You type in your commands one at a time. You make sure that you begin each command with a line number. This helps the computer keep the commands separate when it obeys them.

Here are the commands:

```
10 INPUT N1
20 INPUT N2
30 LET ANSWER=N1*N2
40 PRINT "THE ANSWER IS ";ANSWER
50 GOTO 10
```

All the commands work together to do one job — help you with your homework. When commands work together to do one job, they have a special name. They are called a *program*.

When you type in the program, it is stored in the computer's memory chip. To get the computer to obey your program, you have to get the memory chip to send it to the brain chip. To do that is simple. You just type RUN.

When you type RUN, the computer obeys the commands very quickly, but only one command at a time.

First, it obeys the command on line 10. The command on line 10 tells the computer to print a question mark on the TV screen and accept any number you type on the keyboard. Let's say you type 47. The computer stores the 47 in a little memory cubbyhole you've called N1.

Second, the computer obeys the command on line 20. This is just like the command on line 10. Except now you have the computer accept a number and put it into a cubbyhole you've called N2. You type in 82. The computer puts the 82 into the cubbyhole called N2.

Third, the computer obeys the command on line 30. Line 30 is where the computer performs its multiplication. The "times" sign in the computer's language looks like an asterisk (*). The computer takes the first number (the one stored in N1) and the second number (stored in N2) and multiplies them together.

Now the computer has an answer. Where does the computer put the answer? You guessed it: into the memory cubbyhole you've called ANSWER.

Next, the computer obeys line 40 and prints the answer on the TV screen. It looks like this:

```
THE ANSWER IS 3854
```

What does the computer do next? It looks at line 50. Line 50 tells the computer to "go to" line 10. The computer jumps back to line 10 in your program and asks you for two new numbers. You type in the numbers. It multiplies the numbers together, then prints the answer.

Then what does the computer do? It looks at line 50 and jumps back to line 10 and asks you for *two more numbers*. It will keep on multiplying two numbers for you forever — unless you turn it off or pull out its plug. Or else you can type NEW. The NEW command erases all the old commands in the computer's memory. Then you can feed the computer a new program and teach it something new.

Teach the Computer Anything

Our little program reveals lots of secrets about the computer. Now you know what some of the computer's commands look like. You see how you can combine the commands into a program to make the computer do some work for you. You see how the computer can do the same work over and over without ever getting tired or making mistakes. And you see that you can keep teaching the computer new programs.

What kinds of new programs?

Anything you want. There is no limit to what you can teach the computer. What would you like to teach the computer? What can you imagine?

Kids have taught computers to do all sorts of things. A boy named Larry teaches computers to play games. A girl named Claudia loves music, so she teaches her computer to play the songs she hears on the radio.

You can program the computer to teach you the alphabet, quiz you about the presidents of the United States, give you a spelling bee, or draw pictures of triangles and circles on the TV screen.

You can teach the computer to act like a simple calculator and spit out numbers. Or you can teach it to imitate other machines.

The computer is a great pretender. For example, some computers have been taught how to act like airplanes. The computer pretends it is an airplane, and you pretend that you are the pilot. To fly the plane, you push buttons on its keyboard. The TV screen is the cockpit window.

Or you can teach the computer to do biology experiments and breed hundreds of honeybees. The computer speeds up the bees' lifetime until they live just a few seconds. You get to see how the bee parents' colors, shapes, and abilities are passed on to their children. And their children's children.

Or you can play mad scientist and teach the computer to pretend it is your laboratory. You can conduct experiments with different chemicals. But, watch out! If you mix the wrong chemicals together, your lab might blow up. The computer might flash an explosion on the TV screen. From the TV speakers might come a loud BOOM!

But all is not lost. It's all just pretend. The computer is ready for more. On the TV screen it types: WHAT EXPERIMENT SHOULD WE TRY NEXT?

Personal computers are good at experiments. But they can do a lot more, too. You can teach them to play games such as tic-tac-toe, checkers, backgammon, and chess.

Or you can teach them to take you on an adventure game to a make-believe, faraway planet. There are fabulous treasures on the planet. But the treasures are guarded by a fierce dragon. To win the treasures you must fight the dragon. The computer plays the part of the dragon. One warning: Computer dragons can be very, very tricky.

Or if you are tired of playing games, why not teach the computer to draw pictures? Computers can draw pictures in all the colors of the rainbow. And the pictures move — just like in cartoons.

Kids are teaching computers to draw pictures of zooming race cars, running horses, and pictures of spaceships blasting off from Earth. What would you like your computer to do?

A Robot Inside Your Computer

In 1983 I taught a three-week course on robots in London, England. Our classroom looked like a scene from *Dr. Who*. Every student in the room had a computer and a robot. The robots were not humanlike androids like C3PO and R2-D2. They were miniature robot arms mounted on metal boxes that looked like batteries or electric pencil sharpeners.

The arms came in a rainbow of colors — red, orange, blue, and green. They ended in pincerlike claws, or grippers. Each arm was wired to a student's computer. The student programmed the computer to control the arm. When the arms moved, they buzzed. The classroom sounded like a beehive with a thousand buzzing bees.

Arms moved in all directions. Some picked up erasers, others grabbed small metal blocks and cylinders. Two arms worked together, and even managed to shake hands — or claws. Another arm began moving too fast. It spun around in a circle, got dizzy, and fell off the table.

Turtles, Boxes, and Dinosaurs

The miniature, tabletop arms weren't the only robots in the room. A giant arm was over in the corner of the room near the window. When it wove and swooped, it looked like the neck and head of a prehistoric brontosaurus. A Tasmanian turtle robot scooted around on the floor, drawing pictures and saying, "Hello!" A metal-gray, two-foot-high robot (it looked like R2-D2's kid brother) wandered around the room laughing, saying "HA! HA! HA!" and spinning its head in a circle. And a robot that looked like a little red box zipped back and forth across the kitchen floor carefully staying on its robot "sidewalk." The sidewalk was a piece of black tape that wound around the floor like a piece of licorice candy.

Computers in Their Tummies

This classroom was a real place. It was in a small, red-brick building on the edge of London, in southeast England. I was one of the teachers at a course on "Robotics Literacy." Students from all over the world were getting a chance to play with real robots and learn how to program them using computers.

One robot's computer was a small, green circuit board mounted inside the robot's tummy. The other robots' computers sat on top of tables. The tabletop computers included an Apple IIe and two British computers — a BBC and a Jupiter computer.

The computers' electrical signals controlled the robots. The signals flashed along wire "leashes" plugged into the robots' heads and bottoms. The signals were generated by programs stored in the computers' memories. The students wrote the programs in Robot Language, ARM-BASIC, BASIC, and Logo. The programs made the robots talk, walk, roll around, draw cartoons, play games, and make buildings and bridges out of blocks.

Robots for Kids

Our classroom was real, but it really did seem to be more like science fiction than real life. I know of no other classroom in the world where students got to play with so many computers and so many robots.

Few classrooms have their own robots. The same is true for kids at home. Not every kid has his or her own robot. In fact, very few do. Robots, even in kit form, cost hundreds, or thousands, of dollars. Lots of kids are building robots from kits, but this takes a lot of patience and a lot of careful, painstaking work. (If you would like to learn more about building a robot of your own, take a look at my book *Working Robots*, Dutton/Elsevier-Nelson, 1981.)

Even the new home robots are pretty expensive and may cost several hundred dollars. But soon, maybe only in a couple of years, kids are going to have a lot more robots.

While I was in England, I went to a robot convention. There were a thousand robots at the convention, in all shapes and sizes. Most of the robots were big and expensive and designed for work in factories. But a surprising number were small, friendly, and cheap. They were the first generation of *kids' robots*. Like the more expensive robots in our classroom, they came in four styles — arms, boxlike buggies, turtles, and baby R2-D2s. Judging from what I saw at the convention, I'd predict that these kid robots will soon cost the same as a personal computer (less than $100) and come in all shapes, sizes, and colors.

The new low-cost kids' robots will be easy to program (in BASIC, Logo, PILOT, or Pascal). You would train your robot by programming it. You could train your robot like you would train the family dog or cat.

The robots will be able to talk and maybe even listen. They might not be able to understand spoken commands, but you might be able to give them orders by clapping your hands a certain number of times (for example, one clap to make them go forward, and two claps to make them go backward).

Or you might control them by whistling or making funny noises. When you shouted "Boom!," the robot would wave its arm; or when you sang "La! La! La!," it would roll around in a circle.

Kids' robots will also have *sensors* — electrical devices that imitate human senses like seeing, hearing, and touching. Your robot will probably have several eyes, in the form of sensors that detect motion and measure distance.

Robots that can see can become quite independent little creatures. You will be able to turn a seeing robot loose in a room, and it will find its way around chairs, tables, and people's knees. However, if an object is too low or too high, the robot might miss it. Then, watch out! Your robot may trample your dog's tail, run over your toes, or get lost underneath the dining room table.

Robots will also have sensors that detect light, sound, and heat. Some, like the robot turtles and buggies, will even have a sense of touch. Tiny sensors called *microswitches* can be mounted around the base of the robot. When the robot bumps into something, the switches are pressed down and close an electrical circuit. This sends a message to the robot's computer. Using a control program (that you have to write), the robot will "know" it has bumped into something. Depending on the program, the robot may back up and try to go in a different direction, or it may try to bully its way forward, like a tank or bulldozer.

Many robots today have to be plugged into an electrical outlet on the wall or into a tabletop computer. In the future, all robots will get their orders invisibly, through the air, via microwaves or radio signals. And they will have their power source mounted inside their body, as a rechargeable battery. This will make the robots much more independent. You will turn the robot loose and forget about it. You might not even remember that a robot is in the room, until it marches up behind you, crashes into the back of your leg and says, "BOO!"

Microbots

These neat little robots will be here in just a year or two. But you don't have to wait for them. If you have a computer handy, you can create a robot now — as a program inside the computer.

You may be thinking: How can a robot live inside my computer? But think of the videogames you play, on your computer and at the local games arcade. The games are really computer programs. The programs use sound effects and colorful graphics to create the illusion that tiny creatures live inside the machine. You can see these creatures by looking through the picture-screen "window" on your TV or on the big screen in an arcade game. The creatures can be any shape the game designer imagines—aliens from outer space, snorkle-nosed potato heads, vampires, football players, *or robots.*

There's a Creature in My Computer

When I was a kid I used to go to sleep at night with my hands pulled up·inside the sleeves of my pajamas, and my feet tucked inside two layers of socks and a pair of slippers. I did this to hide my fingers and toes from the lobsters that lived under my bed.

These lobsters weren't just average creatures. First, they didn't need to live in water. Instead, they could somehow survive under my bed — along with lint, dust, dirty clothes, copies of *Mad* magazine, science fiction books, and potato chip crumbs.

Second, if they got hungry, they didn't look for regular lobster food. Instead, they liked to munch on crumpled, smelly socks. (There were lots of those under my bed.) But their favorite food was fingers and toes — *dirty* fingers and *dirty* toes.

I went to bed at night convinced that lobsters really did live under my bed. I was afraid that if I fell asleep and accidentally let my hand or foot slip over the side of the bed, one of the lobsters would leap out, pinch it off, gobble it up, and disappear back under the bed.

The lobsters had never been known to attack clean fingers and clean toes. But I never considered taking a bath. Instead, I bundled up my toes and fingers, and slept in bed all scrunched up like a sunburnt spider. If a lobster wanted to make a meal out of me, it was going to have to work for it.

I shared my bedroom with several lobsters. But we weren't alone. There was also a nightmarish creature who lived underneath my dresser. He would come out from under the dresser when my mother turned out the hall light. He always hid in the shadows. In fact, he *was* a shadow. Real slithery, dark, and tricky. He was all body. No head.

And then there was the creature that lived in my closet behind my dress shirts and Sunday school pants. I called him the Closet Beast. He was one of those shy creatures. He only came out at night when I wanted nothing to do with him. During the day he probably killed time pretending to look like a bow tie or the pair of brown dress shoes I hated.

Between me and all the creatures, the bedroom was crowded. I wished that some of the bedroom creatures would move out. But if they had they would have bumped into the creatures in the other parts of the house. The worst of these was the Ghoul who lived in the cellar, underneath the stairs.

I hoped and prayed I would never meet the cellar Ghoul. But one night I came very close. It all happened because I was a sleepwalker. I was so bad that my mother had to bolt all the windows each night before she went to bed. She was afraid I might climb out one of them and try to sleepwalk on the two-foot ledge that rimmed the roof.

Thanks to my mother I never did any sleepwalking on the roof. But I did sleepwalk a lot inside the house. And I sometimes ended up in some pretty strange places.

One night I woke up and immediately knew something wasn't right. I had my pillow and was wrapped up in my blanket like a mummy, but I wasn't in my bedroom any longer. I was someplace else, someplace very, very dark. And damp. And moldy smelling.

I rubbed my fingers on something hard underneath me. I realized I wasn't in bed. I was on a dusty concrete floor — the basement floor. And I was right next to the stairs where the Ghoul lived.

As dark as the basement was, the space under the stairs was even darker. I couldn't see anything, but I could sense that I was not alone. Something was there with me. And it was coming closer.

I screamed. I screamed again. And again.

I woke up the whole house with my screaming. Moments later, the basement light came on. My parents came flying down the stairs and found me huddled under my blanket, wailing like a ninny.

When they dug my head out of the covers, I pointed toward the stairs. My parents investigated. They didn't catch the Ghoul. But, they did find, hiding under the stairs, a very scared kitty cat.

There's a Creature in My Computer!

I used to see creatures in every shadow or dark corner of my life. I saw so many creatures because I had a crazy imagination.

How about you? Do you have a crazy imagination like mine? Do you see ghosts in wisps of smoke? Do you see sleeping giants inside craggy mountains? Do you see fang-toothed monsters staring up at you out of gutters and hollow stumps of trees? Have you seen the skinny creature who lives inside your medicine cabinet — the one that feasts on stale toothpaste?

With a little imagination you can see creatures everywhere. And, with a little imagination, you can create a creature inside your computer. The creature (he, she, or *it*) might even be living there now. You just have to bring it to life.

The Ghosts in the Machine

Your computer is a perfect place for a creature to live. After all, it's already full of ghosts. The ghosts are other people's programs.

Some people think that programs are just abstract lists full of information and commands. These people are wrong. A program is — or can be — much, much more.

Every program that is written has a personality. Most computer programs written in the past had dull personalities. But they don't have to be dull.

Where does a program's personality come from? It comes from its creator, the person who thought it up and typed the commands into the computer.

The program is a reflection of its creator's imagination.

If the person has a dumpy, dull sort of imagination, the program will be dull. It might have the personality of a stuffed shirt or toad. Most business programs have toady personalities.

On the other hand, if the person's imagination is creative, weird, and funny, then the program will be creative, weird, and funny, too. (Does this remind you of a few game programs you have played?)

Programs are the ghosts inside your computer. So why not turn them into real ghosts, goblins, ogres, zombies, dragons, and other creatures? You can take the creatures that live inside your imagination and load them into your computer. To create the creatures you just write a program. To bring them to life you just type RUN.

Turn On Your Imagination

Warning: If your imagination is having a bad day, you'd better stop here and wait. The creature we're going to create this month is 99 percent imagination and only 1 percent program. The creature is simple, but it can still seem real — *if* you use your imagination.

A Simple Creature

Turn on your computer and type:

```
20 PRINT "GRRRR!!" [Press the RETURN key.]
```

You have just created a creature inside your computer. You don't know what it looks like. You don't know if it wears a ski cap and orange polka-dotted socks, or how many warts are on its nose. But you do know two things: It's there and it's not very friendly.

To see if I'm right, type RUN (and press RETURN). What does the creature do? It says:

```
GRRRR!!
```

Not too friendly, is it?

What happens if you add a new line to the creature's program? For example, type:

```
30 GOTO 20
```

Now type RUN. What happens? This time you see:

```
GRRRR!!
GRRRR!!
GRRRR!!
GRRRR!!
GRRRR!!
GRRRR!!
GRRRR!!
GRRRR!!
```

Now you've created a creature that is *really* unfriendly! (To stop the creature from growling, press the RUN/STOP key.)

So far, the only way to get your creature's attention is to type RUN. But you might want to say something to the creature. To do this, you have to teach the creature to *listen*. To make it listen, type:

```
10 INPUT A$
```

Change line 30 to say GOTO 10. Your whole program now looks like this:

```
10 INPUT A$
20 PRINT "GRRRR!!"
30 GOTO 10
```

Type RUN.

The program begins running, and the creature wakes up. He is looking at you. He is waiting for you to say something. (The computer has printed a ? on the display screen. Imagine that the creature is sprawled in a dungeon inside the computer. He is just waking up. He looks dazed, and has a big ? over his head.)

This is your first chance to say something to the creature. In fact, he won't make a move until you say something.

But what do you say to a creature?

You can try insulting him by saying something like:

YOUR FEET SMELL!

Or you might try giving him a command like:

DON'T EAT ME!

Or you can try to be friendly and ask the creature a question, such as:

DO YOU LIKE PIZZA?

Think up a message, type the message, then type RETURN. What is the creature's answer? He says:

GRRRR!!

He says "GRRRR!!" because it's the only thing he knows how to say. He's a very dumb creature. No matter what you tell him, he always growls. He's a real grump.

To make him say something else, you have to teach him. What sort of new things can you teach your creature to say? What sort of things can you teach your creature to do?

Dream up strange, funny, and unusual creatures, then turn them into programs. Be like a magician pulling rabbits out of a hat. Pull the creatures out of your imagination. Then pop them in the computer and bring them to life.

Chapter 11

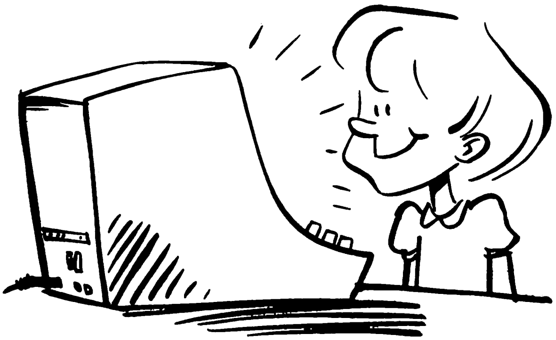
The Computer Friend

Build Your Own Computer Friend

Catie hopped down the steps and jumped out of the bright yellow school bus. She rushed home. She burst into the kitchen, chattered to her father, munched a granola bar, then headed straight for the computer.

She turned the computer on.

A smiling cartoon face appeared on the screen. Under the face, in bright orange, appeared the words, "HI! I'M GED! YOU TURNED ME ON! WHO'S OUT THERE?"



Catie searched the keyboard, then, pecking the keys one at a time, she typed "C-a-t-i-e" and pressed the RETURN button.

"CATIE, BABY!" typed the computer. "I WAS GETTING LONELY. HERE'S A KISS FOR TURNING ME ON. SSSMOOOOCH!"

Catied typed "k-i-s-s" and pressed RETURN.

"YUM. THAT WAS NICE," the computer said. "CATIE, IS TODAY A SCHOOL DAY?"

"Yes," typed Catie.

"IS SCHOOL OVER, CATIE?" asked the computer.

"Yes," typed Catie.

"WHAT DID YOU DO AT SCHOOL TODAY, CATIE?" the computer asked.

Catie told the computer about her field trip to the dairy farm, how she squeezed a cow's udder and milk squirted out, how she got her sneakers muddy in the cow barn, and how the farmer's wife served everyone fresh milk and chocolate chip cookies. "Now I'm home," she concluded.

"I'M GLAD YOU'RE HOME, CATIE. I MISSED YOU," the computer typed. "WANT TO PLAY?"

"Sure," typed Catie. "What game?"

"HOW ABOUT ONE OF THESE," said the computer. A list of Catie's favorite game programs appeared on the screen. Alongside each program was a number. There were word games, number games, storybook games, motor-skill games, and adventure games.

Catie chose game number eight — a "junior" adventure game called *The Castle at the Bottom of the Sea*. She pressed the 8 button.

"GOOD IDEA!" the computer typed. "HAVE FUN! AND REMEMBER: DON'T TRY TO BREATHE UNDERWATER!"

The computer's face briefly appeared, winked, then vanished. Disks lights flashed. The screen turned a deep blue, and the game began.

An Imaginary Playmate

Psychologists say that it is healthy for a child to have an imaginary playmate. That playmate might be in the form of a ratty blanket, a favorite doll, or it might be completely invisible — except in the eyes of the child.

If a blanket, a doll, or an imaginary being can become a friend to a child, why not a computer?

On the surface, most computers are dull. A child can learn how to dredge up all sorts of exciting games and programs from the computer's memory. But, in between games, the computer becomes lifeless, cold, and not especially likable or friendly.

Having a "friend" program or operating system would let you replace the computer's heartless mechanical shell with a personality that is warm, sympathetic, and enjoyable for a child to interact with.

How do you get started? First, the child's computer friend must know certain things about itself.

For example, the computer friend must have a name.

Jonathan Long, of Chapel Hill, North Carolina, sent me a "name" program that he wrote for his Atari computer. The program creates random four-letter names by alternating between consonants and vowels. Slightly modified, this is what Jonathan's program looks like:

```
4 DIM A$(1),TITLE$(4),CONSONANT$(2Ø),VOWELS$
  (6),NAME$(2Ø)
5 CONSONANT$="BCDFGHJKLMNPQRSTVWXY"
6 VOWELS$="AEIOUY"
2Ø FOR OPER=1 TO 4
5Ø IF OPER=2 OR OPER=4 THEN LET LETTER=INT(R
   ND(Ø)*6)+1:NAME$=VOWELS$:GOTO 8Ø
6Ø LET LETTER=INT(RND(Ø)*2Ø)+1
7Ø NAME$=CONSONANT$
8Ø A$=NAME$(LETTER,LETTER)
9Ø TITLE$(LEN(TITLE$)+1)=A$
1ØØ NEXT OPER
11Ø GRAPHICS 2:POKE 752,1:SETCOLOR 4,Ø,Ø:SET
    COLOR 2,Ø,Ø
114 ? "{CLEAR}"
115 POSITION 2,1
12Ø PRINT #6;"hello{A}"
121 POSITION 2,3
122 PRINT #6;"I AM A COMPUTER."
123 POSITION 2,5
125 PRINT #6;"my name is {3 N}"
13Ø POSITION 2,7
135 PRINT #6;TITLE$
199 END
```

Jonathan's program could be inserted in a "computer friend" operating system or master program. It would make your computer be a friend with many names. Routinely, Jonathan's program spouts out names that sound alien, or just plain weird.

In addition to knowing its name, the computer friend should know something about itself. What it "knows" can be either factual or completely made up.

Next, the computer friend should know your child's name. And the names of your child's best friends, his brothers or sisters, his pets, and the names of the child's *other* imaginary friends.

Third, the computer friend must know important details about the child's life. For example, the child's birthday, what grade (if any) he is in at school, who his teachers and babysitters are, what color hair he has, and so on.

Is the child a boy or girl? The computer friend should know.
What is the child's address? What's the name of the town?
What are his likes? What does he think is yucky?

The Computer Friend

What TV programs does he watch? What are his favorite books? Does he like ice cream cones? Pizza? Broccoli with ketchup?

All these things and more a good computer friend should know.

What Kind of Friend Am I?

Friends come in all shapes and sizes. Some are silly. Some are serious. Some are smart. Others are not so smart. Some are aggressive, possessive, and bold. Others are soft, gentle, and sympathetic.

Some friends are all these things at different times.

If a computer tried to mimic human friends, it would have an identity crisis. Its programs would become huge, unwieldy, and complex.

Our computer friends need not be super-intelligent. They need not be like human friends at all. Trying to act human is hard for computers.

Let your computer friend act like a computer. Let it do the things that are friendly, yet are also easy for a computer to do, not hard.

For example, your computer friend can be a good listener.

Think about all the times you were anxious to tell someone about something very important. And nobody listened.

Your human friends care about you. But often they are too wrapped up in their own problems and interests to listen to things that are important to you.

And if you talk too long, human friends may interrupt you, or become angry or bored.

But not a computer friend. You can talk to a (properly programmed) computer friend until you run out of things to say — or pull out its plug.

And it's not like pouring all your feelings, your ideas, your joys, and your fears down into the Black Hole of Calcutta. (Which is what it's like when you talk to people sometimes.)

The computer *can* remember. You can teach it to remember everything (within its memory limitations). Or just bits and pieces. Or after code words like "THIS IS VERY IMPORTANT!"

Or you can teach it to spit everything back at you. That way you can see what you sound like.

What other things can *you* think of that are among a computer's strengths and also help make it a good friend?

A Friend, Hero, Teacher, or Pet?

You can program your computer to have one personality or several. It might be a shrewd sleuth like Sherlock Holmes. It might be brilliant

— but absent-minded — like a nutty inventor. It might be stuffy or laid back, happy or moody, gossipy and inquisitive, or secretive and sly. It's up to you.

Or it might not be a friend at all. It might be something more. For example, the computer might be a hero, programmed to act like the child's favorite pro basketball star, or a comic book character, or a rock singer, or a beautiful actress or model.

Or the computer could be a teacher. Not just the types of computers we see today with teaching *programs*. The computer would have some aspects of a good teacher's personality. It would be warm and friendly. It would never lose its patience. It would guide your child toward various learning activities — programs — stored on the machine. It would be a friend with a single goal. And the goal is learning.

Have you ever heard a computer bark? Or meow?

It could, you know. If it were your child's pet.

Computer pets could be a godsend. Especially where real, live animal pets are not allowed.

Just think: You would never have to potty train or paper train your child's computer pet. You would never have to clean its litter box. Or give it baths. Or take it to the vet.

Yet a computer pet could be very rewarding. Like a computer friend, it could be a companion to your child. And your child could train it herself. She might train it to fetch, or solve puzzles, talk, do tricks, or tell jokes.

Granted, a computer pet is not soft and cuddly like a warm kitten or a fluffy rabbit or gerbil. But turtles and goldfish aren't cuddly either, and they make popular pets.

And why stick to pets that are common?

A child can create a computer pet unlike any animal on Earth. He can fabricate it directly from his imagination or his dreams.

How many legs does the pet have?

How many eyes?

Does the pet have feathers, fur, scales, or skin?

What color?

What does the pet eat?

For answers to all these questions, just ask your child.

What does the pet look like? You and your child can draw the pet on the picture screen.

What does the pet sound like? Use your computer's SOUND command.

Your child's pet can be a monster, a dragon, a dinosaur, or a schmoo. Or a nice brown pony. And your child could learn all about ponies and horses as she created her pet.

In fact, having a computer pet could be a first step toward owning and caring for a real, live pet. Your child could learn all about her computer pet. If it lived, thrived, and was happy, perhaps she could graduate to the real thing.

Or a child might have several computer pets and several real pets.

Or just computer pets.

You and your child decide.

The Creature Inside the Computer

I have talked about the shape of the world inside the computer and the shape of the worlds to come.

There are worlds inside your computer. But there are creatures inside your computer, too. In fact, your computer is swarming with them.

The creatures are really programs. You can write them yourself. Once they are written, your child should be able to call them up just like she'd whistle for her dog when she woke up, or cry "Here, kitty, kitty!"

Your computer is like a puppet theater filled with creatures that aren't real, yet are, nevertheless, very lifelike and entertaining. Your child should be able to put his hands inside the computer puppets, then bring them to life, control them, and interact with them. The more puppets, the richer the child's experience with the computer. Sitting your child in front of the computer could become as exciting as turning him loose in a big-city zoo.

Building the Computer Zoo

We have looked at a couple of the issues related to giving your computer a personality. In the rest of this chapter we will turn to specific programs that impart a simple personality to your family or school's computer. We'll build a computer friend. We'll modify and expand the friend to become a friendly teacher.

The Talking Head

In her book *Machines That Think,* Pamela McCorduck described the ancient popularity of talking heads. Wise men built the heads, then consulted them for useful advice. For example, a medieval pope, Sylvester II, supposedly built a talking head that answered only when spoken to. It was, in a sense, an early computer: To all questions, it gave only two answers — yes or no. Yet, like modern computers, it was credited with having great wisdom and the ability to foretell the future.

Conjuring Up a Talking Head

The first talking heads were products of alchemy and magic. They belong alongside all the other creatures of fantasy, myth, and legend.

Today, a thousand years after the first heads appeared, modern technology has made it possible to build talking heads that are real. The heads are computer-controlled robots, mounted on robot bodies. Scientists are building them in their labs. Youthful hobbyists are building them in their workshops and bedrooms.

But if you don't have the time or skill to build a head out of metal, plastic, and servomotors, don't despair. You can conjure up a talking head on a program and call it forth from the world inside your computer. The head you create may not have the wisdom of the ages or be able to foresee the future, but it can become a great friend for your child.

Mirror, Mirror, on the Wall

In *Snow White,* the evil queen had a magical mirror on the wall. The mirror had a face, a voice, and a puckish, irreverent personality.

Your TV picture screen can be like the queen's mirror. When your child turns on the computer and runs the talking head program, the mirror will darken. Then, magically, a sleeping face will appear. Its eyes will blink open. It will wink at the child, talk to the child, and answer his or her questions in a musical voice.

The Face of a Friend

Appendix A includes the "Talking Head" program for Atari, IBM, TI, Apple, Commodore 64, and VIC computers. The talking head program is just the first step toward a much more elaborate "computer friend" program.

Warning and Acknowledgment

Remember, this program is just the beginning. It makes a good talking head. But, as yet, the head is not good at answering. Since the head cannot carry on a conversation with the child, it is not yet a computer friend.

I would like to thank Bruce Mitchell for some valuable programming assistance. Also, thanks to Richard M. Kruse for the doorbell sound.

The Story Game

Have you ever played *Mad Libs? Mad Libs* is an assortment of wacky party books designed to appeal to the six-year-old in all of us. Each book has a theme — monsters, movies, superheroes, current events, geography, mysteries, or whatever. A *Mad Libs* book is a collection of stories, songs, and rhymes with key words left out. You select the words needed to complete the stories. But you must do it blind.

You play *Mad Libs* by first picking a *reader.* The reader selects a *Mad Lib* from the book. Then he (or she) asks people for words to help fill the blanks in the *Mad Lib.* "Give me a plural noun," the reader might say. Or, "I want the name of a person in this room." The reader fills in the blanks, taking care to hide the story from everyone else in the room.

When all the blanks are filled, the reader stops asking questions and reads the completed story. Depending on people's moods and personalities, the completed *Mad Lib* might sound philosophical, ridiculous, funny, or shockingly raunchy. For example, an exchange between Mickey Mouse and Minnie Mouse might go as follows:

MINNIE: Mickey! Will you stop doing those ______ exercises.
MICKEY: Aw, shucks, Minnie. I have to keep my ______ in shape.
MINNIE: Well, watch out for the ______. Just look at the way your ______ hangs down over your ______.

To fill in the above blanks, the reader would ask people to volunteer an adjective, a plural noun, an adjective, and two singular nouns. Depending on people's answers, the above passage could end up anywhere from banal to cute, or from innocent to X-rated.

Computer Mad Libs for Kids

Japanese author Mitsumasa Anno has a book out called *Topsy-Turvies* (Weatherhill, 1970). Anno likes to play games with your vision and sense of perspective. The book is filled with colorful pictures of topsy-turvy buildings and people capable of walking up walls and strolling on ceilings. The book stretches the visual imagination in the same way as the bizarre paintings and drawings by M.C. Escher.

Mad Libs games can do the same thing for kids' *verbal* and *literary* imaginations. And the reader who chooses the stories and asks for words can be a computer.

As a parent or teacher you can choose a MAD LIBS-type skeleton story from just about anywhere: a picture book, the Bible, a fairy tale, a comic book, a TV program, a song, a poem, or your own imagination. Or you can draw from yours or your child's personal experiences — sort of a fill-in-the-blank autobiography.

After you choose the source for a story, pick out a particularly vivid section of only about 25–300 words. You need to keep it short and fast-paced to maintain the child's interest.

Now go through the story and pick the key words you are going to leave out. Vary your choices. Try to take out different parts of speech: proper nouns (names), adjectives, adverbs, verbs, exclamations, plural nouns, and so on.

Next, type the story into your computer. The program will consist mostly of PRINT statements, like:

```
500 PRINT "FOURSCORE AND SEVEN YEARS AGO
    OUR ";NOUN1$
510 PRINT "BROUGHT FORTH ON THIS CONTINENT"
520 PRINT "A NEW ";NOUN2$;"."
```

The variables NOUN 1$ and NOUN2$ contain the child's answers to questions that the computer reader asked earlier. It asked the questions using PRINT and INPUT statements like:

```
200 PRINT "A PLURAL NOUN";:INPUT NOUN1$
210 PRINT "A SINGULAR NOUN;:INPUT NOUN2$
```

No matter what subject you choose, the computer stories are sure to fascinate children. They are likely to play the same story over and over, trying new words each time. And each time children try a new word, they immediately see its effect. The effect might be dramatic, zany, or silly. But it teaches children the different parts of speech and their roles in a sentence or in a story.

This also stretches children's imaginations and increases their confidence in using new words. After all, it's just a game. They can experiment with new words without being afraid of looking dumb. There won't be any all-knowing adults or smart-aleck peers around to laugh at them if the words make the story crazy or absurd. Instead, it will be fun. And they can change the words on the program's next go around.

Dark Stories

We have a family tradition. Each night, my three-year-old and six-year-old take a bath and get into their pajamas. Then they tumble into bed, climb under the covers, and I turn out the lights. Then I tell them a "dark story." (It might be happy or sad, frightening or funny. It's a *dark* story because it can be told only in the dark.)

I make up a new dark story each night. A dark story is usually a heroic fantasy with lots of evil monsters, princes, princesses, spaceships, and adventures. The stories change, but two things remain the same. First, my daughter Catie is always the story's heroine, my son Eric is always the hero. Second, no matter where the stories end up — in a dismal dungeon or on a faraway planet — they always start someplace that is familiar to my children. That someplace might be their bedroom, their school, or their backyard.

Computer Fairy Tales

I tell a new dark story each night. I don't tell a new story just because I love to be creative. I do it because my memory is so bad. By the time bedtime arrives each night, even the previous night's dark story is usually nothing more than a faint smudge in my memory.

You and your family can create a new dark story each night, too, regardless of the state of your memories and imaginations. The storyteller can be your computer.

You can write programs that combine dark stories with our fill-in-the-blank program. What you get are fractured *fairy tales*. The kids can invent the new words to add to the fairy tales each night. If they are old enough, they can type them in themselves. And you can turn off the lights in the room where you keep the computer to make the fairy tales into true-blue dark stories.

Appendix A includes a sample computer "story game" program for you to try. There are versions for Atari, IBM, TI, Apple, Commodore 64, and VIC. It is a very simple, straightforward program that consists mostly of PRINT statements.

Once you see how to create your own computer fairy tale, you can add to the program or change it completely. Right now, for example, the program asks for nouns (proper names, places, things). You can add adjectives, verbs, nonsense words, etc. Also, the story is in a fairy-tale format appropriate for short bedtime dark stories. But it needn't be. You can rewrite the story to be about anything. Whatever appeals to you and your kids.

And if you are a teacher, not a parent, you can use the story idea in your classroom. You can make up a story-writing assignment that combines programming, language arts, and history or social studies. The subject of the story is up to you.

Appendices

Computer Friend

The "Talking Head" program and the "Story Game" program are included in this appendix. There are versions for Atari (disk and tape), Apple (disk), IBM PC and PCjr (disk), TI-99/4A (disk and tape), Commodore 64 (disk and tape), and VIC-20 (disk and tape).

These two programs work together, and for some versions each will load and run the other. It is therefore important that you read the notes that accompany each version.

See Appendix B for an explanation of the special characters which may appear in the program listings.

Atari

There are two separate programs for disk and tape. Be sure you type in the correct versions for your system. Disk users must save the programs with the correct filenames:

The Talking Head program filename—FRIEND
The Story program filename—TELLTALE

Program 1. Talking Head, Atari Tape Version

```
5Ø REM *** WHEN PROGRAM BEGINS,
55 REM *** THE FRIEND'S FACE
6Ø REM *** APPEARS.  THE FRIEND
65 REM *** WAKES UP AND GREETS
7Ø REM *** THE CHILD.  THE FRIEND
75 REM *** THINKS UP COMPUTER GAMES
8Ø REM *** THE CHILD CAN PLAY.
85 REM
9Ø REM *** TAPE VERSION OF FRIEND
95 REM
1ØØ REM *** DIMENSION VARIABLES
11Ø DIM M$(9):REM * MESSAGE
12Ø N=1:REM * INDIVIDUAL-MESSAGE POINTER
125 DATNUM=1ØØØØ:REM * TYPE-OF-MESSAGE POINT
    ER/1ØØØØ = FRIEND WAKE-UP
```

```
13Ø DIM NAME$(9):REM * CHILD'S NAME
5ØØ REM *** FRIEND MASTER
5Ø5 GRAPHICS 2+16
51Ø IF PEEK(1791)=1 THEN GOSUB 7Ø1Ø:GOTO 55Ø
515 GOSUB 1Ø1Ø:REM * FRIEND WAKE-UP
52Ø GOSUB 2Ø1Ø:REM * FRIEND INTRODUCES HIMSE
    LF/HERSELF
53Ø ANSWER=2:GOSUB 321Ø:REM * FRIEND LEARNS
    CHILD'S NAME
54Ø GOSUB 2Ø1Ø:REM * FRIEND HAPPY TO SEE CHI
    LD
55Ø GOSUB 6Ø1Ø:REM * PLAY GAME?
6ØØ END
1ØØØ REM *** FRIEND WAKE-UP
1Ø1Ø GOSUB 5Ø1Ø:REM * DRAW FACE
1Ø2Ø GOSUB 541Ø:REM * DRAW SLEEP EYES
1Ø3Ø GOSUB 521Ø:REM * DRAW CLOSED MOUTH
1Ø35 FOR P=1 TO 8ØØ:NEXT P
1Ø4Ø GOSUB 4Ø1Ø:REM * WAKE-UP BELL
1Ø5Ø GOSUB 546Ø:REM * DRAW OPEN EYES
1Ø6Ø FOR P=1 TO 6ØØ:NEXT P
1Ø7Ø GOSUB 532Ø:REM * WINK EYE
1Ø8Ø FOR P=1 TO 1ØØ:NEXT P
1Ø85 M=Ø:GOSUB 482Ø:REM * WINK NOISE
1Ø9Ø GOSUB 546Ø:REM * DRAW OPEN EYES
11ØØ FOR P=1 TO 8ØØ:NEXT P
112Ø RETURN
2ØØØ REM *** FRIEND TALK
2Ø1Ø RESTORE DATNUM+N*1Ø:REM * SELECT MESSAG
     E
2Ø11 N=N+1:REM * SET POINTER TO NEXT SET OF
     FRIEND MESSAGES
2Ø12 READ SNUM:REM * SNUM = NUMBER OF SCREEN
     S IN CURRENT SET OF FRIEND MESSAGES
2Ø15 FOR K=1 TO SNUM
2Ø2Ø GOSUB 3Ø1Ø:REM * FRIEND TALK--1 SCREEN
2Ø33 FOR P=1 TO 2ØØ:NEXT P
2Ø35 GOSUB 551Ø:REM * CLEAR MESSAGE WINDOW
2Ø4Ø NEXT K
2Ø5Ø RETURN
3ØØØ REM *** FRIEND TALKING--1 SCREEN
3Ø1Ø PY=2:REM * MESSAGE VERTICAL (Y) START L
     OCATION
3Ø2Ø PY=2:REM * MESSAGE VERTICAL (Y) START L
     OCATION
3Ø3Ø PX=14:REM * HORIZONTAL (X) CENTER OF ME
     SSAGE ON SCREEN
3Ø4Ø READ M$
3Ø5Ø IF M$="-1" THEN RETURN
3Ø51 IF M$="*" THEN M$=NAME$
```

```
3055 GOSUB 5260:REM * OPEN MOUTH
3060 POSITION INT(PX-(LEN(M$)/2)+0.5),PY:REM
      * CENTER LINE
3070 PRINT #6;M$
3075 GOSUB 4810:REM * FRIEND SOUND
3080 FOR P=1 TO 10:NEXT P:REM * KEEP MOUTH O
     PEN
3090 GOSUB 5210:REM * CLOSE MOUTH
3095 FOR P=1 TO 50:NEXT P:REM * KEEP MOUTH C
     LOSED
3100 PY=PY+2
3110 GOTO 3040
3200 REM *** FRIEND ASKS CHILD A QUESTION
3210 OPEN #1,4,0,"K:"
3212 M$=""
3215 POSITION 11,4
3217 FOR I=1 TO 9
3220 GET #1,A
3222 IF A=126 AND I=1 THEN 3220
3225 IF A=126 THEN GOSUB 3310
3230 IF A=155 THEN 3265
3240 PRINT #6;CHR$(A);
3250 M$(LEN(M$)+1)=CHR$(A)
3260 NEXT I
3265 FOR P=1 TO 75:NEXT P
3267 GOSUB 5510:REM * CLEAR MESSAGE WINDOW
3270 CLOSE #1
3280 GOSUB 3410:REM * EVALUATE ANSWER
3290 RETURN
3310 POSITION I+9,4:PRINT #6;" ";
3312 POSITION I+9,4
3315 M$(LEN(M$))=""
3317 I=I-1
3320 GET #1,A
3330 IF A<>126 THEN 3390
3350 IF I<2 THEN 3320
3360 GOTO 3310
3390 RETURN
3400 REM *** EVALUATE ANSWER
3410 ON ANSWER GOSUB 3510,3610
3420 RETURN
3500 REM *** NO NEED TO STORE ANSWER
3510 RETURN
3600 REM *** ANSWER=CHILD'S NAME
3610 NAME$=M$
3620 RETURN
4000 REM *** WAKE-UP BELL
4010 BEL=105:TIM=7.5:GOSUB 4040
4020 BEL=132:TIM=8.5:GOSUB 4040
4030 SOUND 0,0,0,0:RETURN
```

```
4040 VLM=15:INC=0.79+TIM/50
4050 SOUND 0,BEL,10,VLM
4060 VLM=VLM*INC
4070 IF VLM>1 THEN 4050
4080 RETURN
4800 REM *** FRIEND VOICE
4810 M=INT(RND(1)*51)+15
4820 FOR A=M+25 TO M STEP -8
4830 SOUND 0,A,10,10
4840 FOR T=1 TO 10
4850 NEXT T
4860 NEXT A
4875 SOUND 0,0,0,0
4880 RETURN
5000 REM *** FRIEND'S FACE
5010 GRAPHICS 2+16
5040 POSITION 2,1:PRINT #6;"{3 SPACES}*"
5050 POSITION 2,2:PRINT #6;"  / \"
5060 POSITION 2,3:PRINT #6;" ====="
5070 POSITION 2,4:PRINT #6;"/{5 SPACES}\"
5090 POSITION 1,6:PRINT #6;"<:  ^  :>"
5100 POSITION 2,9:PRINT #6;"\_____/"
5110 RETURN
5200 REM *** CLOSE MOUTH
5210 POSITION 2,7:PRINT #6;":{5 SPACES}:"
5220 POSITION 2,8:PRINT #6;": --- :"
5230 RETURN
5250 REM *** OPEN MOUTH
5260 POSITION 2,7:PRINT #6;": ___ :"
5270 POSITION 2,8:PRINT #6;": \_/ :"
5280 RETURN
5300 REM *** LEFT EYE WINK
5320 POSITION 2,5:PRINT #6;": O - :"
5330 FOR P=1 TO 150:NEXT P
5340 RETURN
5400 REM *** EYES ASLEEP
5410 POSITION 2,5:PRINT #6;": - - :"
5440 RETURN
5450 REM *** EYES AWAKE
5460 POSITION 2,5:PRINT #6;": O O :"
5470 RETURN
5500 REM *** CLEAR MESSAGE WINDOW
5510 FOR Y=2 TO 8 STEP 2
5520 POSITION 10,Y
5530 PRINT #6;"{9 SPACES}"
5540 NEXT Y
5550 RETURN
6000 REM *** FRIEND'S GAMES
6010 GOSUB 2010:REM * FRIEND ASKS CHILD: PLA
     Y A GAME?
```

```
6Ø2Ø ANSWER=1:GOSUB 321Ø:REM * GET CHILD'S A
     NSWER
6Ø3Ø IF M$(1,1)="N" THEN 6Ø8Ø
6Ø4Ø IF M$(1,1)<>"Y" THEN N=N-1:GOTO 6Ø1Ø
6Ø5Ø GOSUB 611Ø:GOTO 6Ø9Ø:REM * SELECT GAME
6Ø8Ø N=4:DATNUM=13ØØØ:REM * GOOD-BYE
6Ø9Ø GOSUB 2Ø1Ø:REM * FRIEND SAYS GOOD-BYE!
6Ø95 RETURN
61ØØ REM *** SELECT GAME
611Ø DATNUM=12ØØØ:N1=N:N=1:REM * RESET DATA
     POINTERS
612Ø GOSUB 2Ø1Ø:REM * GENIE BEGINS GAME-SELE
     CTION QUESTION
613Ø READ GAMENUM
614Ø N=N+1
615Ø FOR Z=1 TO GAMENUM
616Ø GOSUB 2Ø1Ø:REM * DISPLAY GAME NAME
617Ø GOSUB 321Ø:REM * GET CHILD'S ANSWER
618Ø IF M$(1,1)="Y" THEN GOSUB 631Ø:GOSUB 64
     1Ø:GOTO 622Ø
619Ø IF M$(1,1)<>"N" THEN N=N-1:GOTO 616Ø
62ØØ NEXT Z
621Ø DATNUM=12ØØØ:N=53:RETURN :REM * NO GAME
     S SELECTED/GOOD-BYE!
622Ø DATNUM=13ØØØ:N=4:REM * GOOD-BYE!:RETURN
63ØØ REM *** PREPARE FRIEND'S MEMORY FOR EXI
     T FROM FRIEND PROGRAM
63Ø1 REM *** STORE CHILD'S NAME
63Ø2 REM *** IN LOCATIONS
63Ø3 REM *** 1781-1789
63Ø4 REM *** (LENGTH OF NAME IN 179Ø)
63Ø5 REM *** AND SET LOCATION 1791
63Ø6 REM *** AS FLAG THAT
63Ø7 REM *** FRIEND HAS ALREADY
63Ø8 REM *** BEEN CALLED SINCE
63Ø9 REM *** TURNING ON COMPUTER
631Ø REM
6315 FOR I=1 TO LEN(NAME$)
632Ø POKE 178Ø+I,ASC(NAME$(I,I))
633Ø NEXT I
6335 POKE 179Ø,LEN(NAME$)
634Ø POKE 1791,1
635Ø RETURN
64ØØ REM *** TAPE VERSION OF FRIEND
641Ø DATNUM=13ØØØ:N=1
642Ø GOSUB 2Ø1Ø:REM * ASK IF CHILD NEEDS HEL
     P
643Ø ANSWER=1:GOSUB 321Ø:REM * GET CHILD'S A
     NSWER
644Ø IF M$(1,1)="N" THEN 65ØØ
```

```
6450 IF M$(1,1)<>"Y" THEN 6410
6460 GOSUB 2010:REM * TELL CHILD HOW TO LOAD
      TAPE/REPEAT STEPS?
6470 N=2:GOTO 6430
6500 RETURN
7000 REM *** FRIEND CALLED ON BEFORE
7010 FOR I=1 TO PEEK(1790)
7020 NAME$(LEN(NAME$)+1)=CHR$(PEEK(1780+I))
7030 NEXT I
7040 GOSUB 5010:GOSUB 5210:GOSUB 5460:REM *
     DRAW FRIEND
7050 DATNUM=11000:GOSUB 2010:REM * NEW FRIEN
     D MESSAGES
7060 DATNUM=10000:N=3
7070 RETURN
10000 REM *** WAKE-UP FRIEND
10005 REM *** MESSAGES
10010 DATA 3
10011 DATA HI, I'M, GED,-1
10012 DATA YOU,TURNED,ME,ON,-1
10013 DATA WHO'S,OUT,THERE?,-1
10020 DATA 2
10021 DATA I'M,SO,HAPPY,-1
10022 DATA TO,SEE,YOU,*,-1
10030 DATA 2
10031 DATA DO,YOU,WANT,-1
10032 DATA TO,PLAY,A,GAME?,-1
11000 REM *** FRIEND ALREADY AWAKE MESSAGES
11010 DATA 5
11011 DATA HI,*,-1
11012 DATA I,HOPE,YOU,-1
11013 DATA HAD,FUN!!,-1
11014 DATA I,WONDER,WHAT,-1
11015 DATA WE,SHOULD,DO,NOW.,-1
12000 REM *** GAMES
12001 REM
12002 REM *** LIST GAMES ON
12003 REM *** EVERY 10TH LINE--
12004 REM *** LINES 12030-12520
12005 REM *** FOR A MAXIMUM OF
12006 REM *** 50 GAMES.
12007 REM
12010 DATA 2
12011 DATA DO,YOU,WANT,-1
12012 DATA TO,PLAY,-1
12020 DATA 1
12030 DATA 1
12031 DATA THE,STORY,GAME?,-1
12530 DATA 6
12531 DATA *,I,AM,SORRY,-1
```

```
12532 DATA NONE,OF,THE,GAMES,-1
12533 DATA LOOKED,FUN.,-1
12534 DATA MAYBE,WE,CAN,-1
12535 DATA PLAY,LATER.,-1
12536 DATA BYE!,BYE!,BYE!,-1
13ØØØ REM *** MESSAGE TO HELP
13ØØ5 REM *** CHILD LOAD GAME
13ØØ6 REM *** TAPE
13Ø1Ø DATA 3
13Ø11 DATA ALL,MY,GAMES,-1
13Ø12 DATA ARE,STORED,ON,TAPE.,-1
13Ø13 DATA DO,YOU,NEED,HELP?,-1
13Ø2Ø DATA 12
13Ø21 DATA FIRST,PUT,THE,GAME,-1
13Ø22 DATA TAPE,ON,THE,RECORDER.,-1
13Ø23 DATA SECOND,REWIND,THE,TAPE.,-1
13Ø24 DATA THIRD,TYPE,'CLOAD',THEN,-1
13Ø25 DATA PRESS,THE,'RETURN',BUTTON.,-1
13Ø26 DATA FOURTH,PRESS,THE,'PLAY',-1
13Ø27 DATA BUTTON,ON,THE,RECORDER.,-1
13Ø28 DATA LAST,PRESS,'RETURN',AGAIN.,-1
13Ø29 DATA WHEN,THE,PROGRAM,IS,-1
13Ø3Ø DATA LOADED,TYPE,'RUN'.,-1
13Ø31 DATA WANT,ME,TO,REPEAT,-1
13Ø32 DATA THE,STEPS?,-1
13Ø4Ø DATA 1
13Ø41 DATA BYE!,BYE!,BYE!,-1
```

Program 2. The Story Game, Atari Tape Version

```
5Ø REM **************
55 REM THE STORY GAME
6Ø REM **************
65 REM ***
7Ø REM *** PROGRAM HELPS
75 REM *** CHILD AND PARENT
8Ø REM *** INVENT THEIR OWN
85 REM *** FAIRY TALE.
9Ø REM ***
95 REM *** DATA SECTION
96 REM ***
11Ø DIM N1$(1Ø),N2$(1Ø),N3$(1Ø),N4$(1Ø),N5$(
    1Ø),N6$(1Ø)
12Ø DIM ANSWER$(1)
5ØØ REM ***
51Ø REM *** PROGRAM INTRODUCES
52Ø REM *** ITSELF
53Ø REM ***
535 PRINT "{CLEAR}"
```

```
540 SETCOLOR 0,14,10
550 POSITION 8,4:PRINT "*** THE STORY GAME *
    **":PRINT :PRINT
560 FOR I=1 TO 500:NEXT I
800 REM ***
810 REM *** PROGRAM ASKS FOR
812 REM *** KEY INGREDIENTS
814 REM *** OF STORY
815 REM ***
840 POSITION 8,7:PRINT "NAME OF HERO";:INPUT
     N1$
870 POSITION 8,9:PRINT "ENCHANTED PLACE";:IN
    PUT N2$
890 POSITION 8,11:PRINT "VILLAIN";:INPUT N3$
910 POSITION 8,13:PRINT "BAD PLACE";:INPUT N
    4$
930 POSITION 8,15:PRINT "MAGIC THING";:INPUT
     N5$
940 POSITION 8,17:PRINT "SILLY CREATURE";:IN
    PUT N6$
950 REM ***
960 REM *** PROGRAM TELLS STORY
970 REM ***
1000 GRAPHICS 2+16
1001 SETCOLOR 0,6,10
1003 POSITION 6,3:PRINT #6;N1$;" AND"
1004 POSITION 6,5:PRINT #6;"THE MAGIC"
1005 POSITION 6,7:PRINT #6;N5$
1008 FOR I=1 TO 1500:NEXT I
1009 GRAPHICS 2+16
1010 PRINT #6;" ONCE UPON A TIME,"
1015 PRINT #6;" "
1020 PRINT #6;" A BRAVE CHILD"
1025 PRINT #6;" "
1030 PRINT #6;" NAMED ";N1$
1035 PRINT #6;" "
1040 PRINT #6;" WENT EXPLORING"
1045 PRINT #6;" "
1050 PRINT #6;" IN AN ENCHANTED"
1052 PRINT #6;" "
1055 PRINT #6;" ";N2$;"."
1060 FOR I=1 TO 1500:NEXT I
1070 GRAPHICS 2+16
1080 PRINT #6;" IN THE ";N2$
1085 PRINT #6;" "
1090 PRINT #6;" LIVED A HUGE,"
1095 PRINT #6;" "
1100 PRINT #6;" EVIL ";N3$
1105 PRINT #6;" "
```

```
11Ø6 PRINT #6;" WHO LOVED"
11Ø7 PRINT #6;" "
111Ø PRINT #6;" TO EAT CHILDREN."
1115 FOR I=1 TO 15ØØ:NEXT I
1116 GRAPHICS 2+16
112Ø PRINT #6;" THE ";N3$
1125 PRINT #6;" "
1126 PRINT #6;" TRAPPED ";N1$
1127 PRINT #6;" "
113Ø PRINT #6;" AND PUT"
1135 PRINT #6;" "
114Ø PRINT #6;" ";N1$;" IN A DARK,"
1145 PRINT #6;" "
115Ø PRINT #6;" STINKY ";N4$;"."
1155 FOR I=1 TO 15ØØ:NEXT I
1156 GRAPHICS 2+16
116Ø PRINT #6;" ";N1$;" SNEAKED"
1165 PRINT #6;" "
117Ø PRINT #6;" OUT OF THE"
1175 PRINT #6;" "
118Ø PRINT #6;" ";N4$;" AND"
1185 PRINT #6;" "
119Ø PRINT #6;" GRABBED"
1195 PRINT #6;" "
12ØØ PRINT #6;" THE ";N3$;"'S"
12Ø1 PRINT #6;" "
12Ø2 PRINT #6;" MAGIC ";N5$;"."
121Ø FOR I=1 TO 15ØØ:NEXT I
122Ø GRAPHICS 2+16
123Ø PRINT #6;" ";N1$;" WAVED THE"
1235 PRINT #6;" "
124Ø PRINT #6;" ";N5$;" AND"
1245 PRINT #6;" "
1246 PRINT #6;" TURNED THE"
1247 PRINT #6;" "
125Ø PRINT #6;" ";N3$;" INTO "
1255 PRINT #6;" "
126Ø PRINT #6;" A FAT, LAZY"
1265 PRINT #6;" "
1266 PRINT #6;" ";N6$;"."
127Ø FOR I=1 TO 15ØØ:NEXT I
128Ø GRAPHICS 2+16
129Ø PRINT #6;" ";N1$;" CHASED THE"
1295 PRINT #6;" "
13ØØ PRINT #6;" ";N6$;" THROUGH"
13Ø1 PRINT #6;" "
13Ø2 PRINT #6;" THE ";N2$;"."
131Ø PRINT #6;" "
1311 PRINT #6;" THEN ";N1$;" RAN"
```

```
1315 PRINT #6;" "
1320 PRINT #6;" ALL THE WAY"
1325 PRINT #6;" "
1330 PRINT #6;" BACK HOME."
1350 FOR I=1 TO 1500:NEXT I
1360 GRAPHICS 2+16
1365 SETCOLOR 0,6,10
1370 POSITION 2,5:PRINT #6;"**  THE END  **"
1390 FOR I=1 TO 1500:NEXT I
1395 REM ***
1396 REM *** WOULD CHILD LIKE THE
1397 REM *** SAME STORY AGAIN?
1398 REM ***
1400 GRAPHICS Ø
1402 PRINT "WOULD YOU LIKE TO SEE"
1403 PRINT "THE SAME STORY AGAIN";:INPUT ANS
     WER$
1404 IF ANSWER$="Y" THEN 1ØØØ
1405 IF ANSWER$<>"N" THEN 14ØØ
1406 REM ***
1407 REM *** WOULD CHILD LIKE TO
1408 REM *** INVENT A NEW STORY?
1409 REM ***
1410 GRAPHICS Ø
1411 PRINT "WOULD YOU LIKE TO"
1415 PRINT "INVENT A NEW STORY";:INPUT ANSWE
     R$
1420 IF ANSWER$="Y" THEN 535
1430 IF ANSWER$<>"N" THEN 14Ø6
1435 GRAPHICS Ø
1436 REM ***
1437 REM *** RETURN TO FRIEND
1438 REM *** PROGRAM--ON TAPE
1439 REM ***
1440 PRINT "TIME TO WAKE UP YOUR FRIEND!":PR
     INT
1442 PRINT "DO YOU NEED INSTRUCTIONS";:INPUT
      ANSWER$
1443 IF ANSWER$="N" THEN GRAPHICS Ø:GOTO 153
     Ø
1445 IF ANSWER$<>"Y" THEN 1435
1446 GRAPHICS Ø
1450 PRINT "PUT 'FRIEND' TAPE INTO"
1460 PRINT "PROGRAM RECORDER."
1465 PRINT
1470 PRINT "REWIND TAPE TO BEGINNING."
1475 PRINT
1490 PRINT "TYPE 'CLOAD' AND PRESS RETURN."
1495 PRINT
1496 PRINT "PRESS PLAY BUTTON"
```

```
1497 PRINT "ON THE PROGRAM RECORDER."
1498 PRINT
1499 PRINT "PRESS RETURN AGAIN."
1500 PRINT
1505 PRINT "WHEN COMPUTER IS FINISHED"
1506 PRINT "LOADING THE 'FRIEND' TAPE,"
1508 PRINT "THE COMPUTER WILL TYPE 'READY'."
1509 PRINT
1510 PRINT "THEN YOU TYPE 'RUN'"
1515 PRINT "AND PRESS RETURN."
1520 PRINT
1530 END
```

Program 3. Talking Head, Atari Disk Version

```
40 GRAPHICS 2+16
100 REM *** DIMENSION VARIABLES
110 DIM M$(9):REM * MESSAGE
120 N=1:REM * MESSAGE POINTER
125 DATNUM=10000:REM * WHERE TO READ DATA ST
    ATEMENTS
130 DIM NAME$(9):REM * CHILD'S NAME
500 REM *** FRIEND MASTER
510 IF PEEK(1791)=1 THEN GOSUB 7010:GOTO 550
515 GOSUB 1010:REM * FRIEND WAKE-UP
520 GOSUB 2010:REM * FRIEND INTRODUCES HIMSE
    LF/HERSELF
530 ANSWER=2:GOSUB 3210:REM * FRIEND LEARNS
    CHILD'S NAME
540 GOSUB 2010:REM * FRIEND HAPPY TO SEE CHI
    LD
550 GOSUB 6010:REM * PLAY GAME?
600 END
1000 REM *** FRIEND WAKE-UP
1010 GOSUB 5010:REM * DRAW FACE
1020 GOSUB 5410:REM * DRAW SLEEP EYES
1030 GOSUB 5210:REM * DRAW CLOSED MOUTH
1035 FOR P=1 TO 800:NEXT P
1040 GOSUB 4010:REM * WAKE-UP BELL
1050 GOSUB 5460:REM * DRAW OPEN EYES
1060 FOR P=1 TO 600:NEXT P
1070 GOSUB 5320:REM * WINK EYE
1080 FOR P=1 TO 100:NEXT P
1085 M=0:GOSUB 4820:REM * WINK NOISE
1090 GOSUB 5460:REM * DRAW OPEN EYES
1100 FOR P=1 TO 800:NEXT P
1120 RETURN
2000 REM *** FRIEND TALK
2010 RESTORE DATNUM+N*10:REM * SELECT MESSAG
     E
```

```
2011 N=N+1:REM * SET POINTER TO NEXT SET OF
     FRIEND MESSAGES
2012 READ SNUM:REM * SNUM = NUMBER OF SCREEN
     S IN CURRENT SET OF FRIEND MESSAGES
2015 FOR K=1 TO SNUM
2020 GOSUB 3010:REM * FRIEND TALK--1 SCREEN
2033 FOR P=1 TO 200:NEXT P
2035 GOSUB 5510:REM * CLEAR MESSAGE WINDOW
2040 NEXT K
2050 RETURN
3000 REM *** FRIEND TALKING--1 SCREEN
3010 PY=2:REM * MESSAGE VERTICAL (Y) START L
     OCATION
3020 PY=2:REM * MESSAGE VERTICAL (Y) START L
     OCATION
3030 PX=14:REM * HORIZONTAL (X) CENTER OF ME
     SSAGE ON SCREEN
3040 READ M$
3050 IF M$="-1" THEN RETURN
3051 IF M$="*" THEN M$=NAME$
3055 GOSUB 5260:REM * OPEN MOUTH
3060 POSITION INT(PX-(LEN(M$)/2)+0.5),PY:REM
      * CENTER LINE
3070 PRINT #6;M$
3075 GOSUB 4810:REM * FRIEND SOUND
3080 FOR P=1 TO 10:NEXT P:REM * KEEP MOUTH O
     PEN
3090 GOSUB 5210:REM * CLOSE MOUTH
3095 FOR P=1 TO 50:NEXT P:REM * KEEP MOUTH C
     LOSED
3100 PY=PY+2
3110 GOTO 3040
3200 REM *** FRIEND ASKS CHILD A QUESTION
3210 OPEN #1,4,0,"K:"
3212 M$=""
3215 POSITION 11,4
3217 FOR I=1 TO 9
3220 GET #1,A
3222 IF A=126 AND I=1 THEN 3220
3225 IF A=126 THEN GOSUB 3310
3230 IF A=155 THEN 3265
3240 PRINT #6;CHR$(A);
3250 M$(LEN(M$)+1)=CHR$(A)
3260 NEXT I
3265 FOR P=1 TO 75:NEXT P
3267 GOSUB 5510:REM * CLEAR MESSAGE WINDOW
3270 CLOSE #1
3280 GOSUB 3410:REM * EVALUATE ANSWER
3290 RETURN
```

```
3310 POSITION I+9,4:PRINT #6;" ";
3312 POSITION I+9,4
3315 M$(LEN(M$))=""
3317 I=I-1
3320 GET #1,A
3330 IF A<>126 THEN 3390
3350 IF I<2 THEN 3320
3360 GOTO 3310
3390 RETURN
3400 REM *** EVALUATE ANSWER
3410 ON ANSWER GOSUB 3510,3610
3420 RETURN
3500 REM *** NO NEED TO STORE ANSWER
3510 RETURN
3600 REM *** ANSWER=CHILD'S NAME
3610 NAME$=M$
3620 RETURN
4000 REM *** WAKE-UP BELL
4010 BEL=105:TIM=7.5:GOSUB 4040
4020 BEL=132:TIM=8.5:GOSUB 4040
4030 SOUND 0,0,0,0:RETURN
4040 VLM=15:INC=0.79+TIM/50
4050 SOUND 0,BEL,10,VLM
4060 VLM=VLM*INC
4070 IF VLM>1 THEN 4050
4080 RETURN
4800 REM *** FRIEND VOICE
4810 M=INT(RND(1)*51)+15
4820 FOR A=M+25 TO M STEP -8
4830 SOUND 0,A,10,10
4840 FOR T=1 TO 10
4850 NEXT T
4860 NEXT A
4875 SOUND 0,0,0,0
4880 RETURN
5000 REM *** FRIEND'S FACE
5010 GRAPHICS 2+16
5040 POSITION 2,1:PRINT #6;"{3 SPACES}*"
5050 POSITION 2,2:PRINT #6;"  / \"
5060 POSITION 2,3:PRINT #6;" ====="
5070 POSITION 2,4:PRINT #6;"/{5 SPACES}\"
5090 POSITION 1,6:PRINT #6;"<:  ^  :>"
5100 POSITION 2,9:PRINT #6;"\_____/"
5110 RETURN
5200 REM *** CLOSE MOUTH
5210 POSITION 2,7:PRINT #6;":{5 SPACES}:"
5220 POSITION 2,8:PRINT #6;": --- :"
5230 RETURN
5250 REM *** OPEN MOUTH
5260 POSITION 2,7:PRINT #6;": ___ :"
```

```
527Ø POSITION 2,8:PRINT #6;": \_/ :"
528Ø RETURN
53ØØ REM *** LEFT EYE WINK
532Ø POSITION 2,5:PRINT #6;": O - :"
533Ø FOR P=1 TO 15Ø:NEXT P
534Ø RETURN
54ØØ REM *** EYES ASLEEP
541Ø POSITION 2,5:PRINT #6;": - - :"
544Ø RETURN
545Ø REM *** EYES AWAKE
546Ø POSITION 2,5:PRINT #6;": O O :"
547Ø RETURN
55ØØ REM *** CLEAR MESSAGE WINDOW
551Ø FOR Y=2 TO 8 STEP 2
552Ø POSITION 1Ø,Y
553Ø PRINT #6;"{9 SPACES}"
554Ø NEXT Y
555Ø RETURN
6ØØØ REM *** FRIEND'S GAMES
6Ø1Ø GOSUB 2Ø1Ø:REM * FRIEND ASKS CHILD: PLA
     Y A GAME?
6Ø2Ø ANSWER=1:GOSUB 321Ø:REM * GET CHILD'S A
     NSWER
6Ø3Ø IF M$(1,1)="N" THEN 6Ø9Ø
6Ø4Ø IF M$(1,1)<>"Y" THEN N=N-1:GOTO 6Ø1Ø
6Ø5Ø GOSUB 611Ø:REM * SELECT GAME
6Ø9Ø RESTORE 12526:SNUM=1:GOSUB 2Ø15:REM * F
     RIEND SAYS GOOD-BYE!
6Ø95 RETURN
61ØØ REM *** SELECT GAME
611Ø DATNUM=12ØØØ:N1=N:N=1:REM * RESET DATA
     POINTERS
612Ø GOSUB 2Ø1Ø:REM * GENIE BEGINS GAME-SELE
     CTION QUESTION
613Ø READ GAMENUM
614Ø N=N+1
615Ø FOR Z=1 TO GAMENUM
616Ø GOSUB 2Ø1Ø:REM * DISPLAY GAME NAME
617Ø GOSUB 321Ø:REM * GET CHILD'S ANSWER
618Ø IF M$(1,1)="Y" THEN GOSUB 631Ø:GOTO 641
     Ø
619Ø IF M$(1,1)<>"N" THEN N=N-1:GOTO 616Ø
62ØØ NEXT Z
621Ø N=52:GOSUB 2Ø1Ø:REM * NO GAMES SELECTED
     --FRIEND'S SORRY MESSAGE
622Ø RETURN
63ØØ REM *** PREPARE FRIEND'S MEMORY FOR EXI
     T FROM FRIEND PROGRAM
63Ø1 REM *** STORE CHILD'S NAME
```

```
6302 REM *** IN LOCATIONS
6303 REM *** 1781-1789
6304 REM *** (LENGTH OF NAME IN 1790)
6305 REM *** AND SET LOCATION 1791
6306 REM *** AS FLAG THAT
6307 REM *** FRIEND HAS ALREADY
6308 REM *** BEEN CALLED SINCE
6309 REM *** TURNING ON COMPUTER
6310 REM
6315 FOR I=1 TO LEN(NAME$)
6320 POKE 1780+I,ASC(NAME$(I,I))
6330 NEXT I
6335 POKE 1790,LEN(NAME$)
6340 POKE 1791,1
6350 RETURN
6400 REM *** DISK VERSION OF FRIEND
6405 REM ***
6408 REM *** SELECT GAME PROGRAM/EXIT FRIEND
6410 GOTO 6410+Z*10
6420 RUN "D:TELLTALE"
7000 REM *** FRIEND CALLED ON BEFORE
7010 FOR I=1 TO PEEK(1790)
7020 NAME$(LEN(NAME$)+1)=CHR$(PEEK(1780+I))
7030 NEXT I
7040 GOSUB 5010:GOSUB 5210:GOSUB 5460:REM *
     DRAW FRIEND
7050 DATNUM=11000:GOSUB 2010:REM * NEW FRIEN
     D MESSAGES
7060 DATNUM=10000:N=3
7070 RETURN
10000 REM *** WAKE-UP FRIEND
10005 REM *** MESSAGES
10010 DATA 3
10011 DATA HI, I'M, GED,-1
10012 DATA YOU,TURNED,ME,ON,-1
10013 DATA WHO'S,OUT,THERE?,-1
10020 DATA 2
10021 DATA I'M,SO,HAPPY,-1
10022 DATA TO,SEE,YOU,*,-1
10030 DATA 2
10031 DATA DO,YOU,WANT,-1
10032 DATA TO,PLAY,A,GAME?,-1
11000 REM *** FRIEND ALREADY AWAKE MESSAGES
11010 DATA 5
11011 DATA HI,*,-1
11012 DATA I,HOPE,YOU,-1
11013 DATA HAD,FUN!!,-1
11014 DATA I,WONDER,WHAT,-1
11015 DATA WE,SHOULD,DO,NOW.,-1
12000 REM *** GAMES
```

```
12001 REM
12002 REM *** LIST GAMES ON
12003 REM *** EVERY 10TH LINE--
12004 REM *** LINES 12030-12490
12005 REM *** FOR A MAXIMUM OF
12006 REM *** 50 GAMES.
12007 REM
12010 DATA 2
12011 DATA DO,YOU,WANT,-1
12012 DATA TO,PLAY,-1
12020 DATA 1
12030 DATA 1
12031 DATA THE,STORY,GAME?,-1
12520 DATA 6
12521 DATA *,I,AM,SORRY,-1
12522 DATA NONE,OF,THE,GAMES,-1
12523 DATA LOOKED,FUN.,-1
12524 DATA MAYBE,WE,CAN,-1
12525 DATA PLAY,LATER.,-1
12526 DATA BYE!,BYE!,BYE!,-1
```

Program 4. The Story Game, Atari Disk

```
50 REM *************
55 REM THE STORY GAME
60 REM *************
65 REM ***
70 REM *** PROGRAM HELPS
75 REM *** CHILD AND PARENT
80 REM *** INVENT THEIR OWN
85 REM *** FAIRY TALE.
90 REM ***
95 REM *** DATA SECTION
96 REM ***
110 DIM N1$(10),N2$(10),N3$(10),N4$(10),N5$(
    10),N6$(10)
120 DIM ANSWER$(1)
500 REM ***
510 REM *** PROGRAM INTRODUCES
520 REM *** ITSELF
530 REM ***
535 PRINT "{CLEAR}"
540 SETCOLOR 0,14,10
550 POSITION 8,4:PRINT "*** THE STORY GAME *
    **":PRINT :PRINT
560 FOR I=1 TO 500:NEXT I
800 REM ***
810 REM *** PROGRAM ASKS FOR
812 REM *** KEY INGREDIENTS
```

```
814 REM *** OF STORY
815 REM ***
840 POSITION 8,7:PRINT "NAME OF HERO";:INPUT
     N1$
870 POSITION 8,9:PRINT "ENCHANTED PLACE";:IN
    PUT N2$
890 POSITION 8,11:PRINT "VILLAIN";:INPUT N3$
910 POSITION 8,13:PRINT "BAD PLACE";:INPUT N
    4$
930 POSITION 8,15:PRINT "MAGIC THING";:INPUT
     N5$
940 POSITION 8,17:PRINT "SILLY CREATURE";:IN
    PUT N6$
950 REM ***
960 REM *** PROGRAM TELLS STORY
970 REM ***
1000 GRAPHICS 2+16
1001 SETCOLOR 0,6,10
1003 POSITION 6,3:PRINT #6;N1$;" AND"
1004 POSITION 6,5:PRINT #6;"THE MAGIC"
1005 POSITION 6,7:PRINT #6;N5$
1008 FOR I=1 TO 1500:NEXT I
1009 GRAPHICS 2+16
1010 PRINT #6;" ONCE UPON A TIME,"
1015 PRINT #6;" "
1020 PRINT #6;" A BRAVE CHILD"
1025 PRINT #6;" "
1030 PRINT #6;" NAMED ";N1$
1035 PRINT #6;" "
1040 PRINT #6;" WENT EXPLORING"
1045 PRINT #6;" "
1050 PRINT #6;" IN AN ENCHANTED"
1052 PRINT #6;" "
1055 PRINT #6;" ";N2$;"."
1060 FOR I=1 TO 1500:NEXT I
1070 GRAPHICS 2+16
1080 PRINT #6;" IN THE ";N2$
1085 PRINT #6;" "
1090 PRINT #6;" LIVED A HUGE,"
1095 PRINT #6;" "
1100 PRINT #6;" EVIL ";N3$
1105 PRINT #6;" "
1106 PRINT #6;" WHO LOVED"
1107 PRINT #6;" "
1110 PRINT #6;" TO EAT CHILDREN."
1115 FOR I=1 TO 1500:NEXT I
1116 GRAPHICS 2+16
1120 PRINT #6;" THE ";N3$
1125 PRINT #6;" "
```

```
1126 PRINT #6;" TRAPPED ";N1$
1127 PRINT #6;" "
1130 PRINT #6;" AND PUT"
1135 PRINT #6;" "
1140 PRINT #6;" ";N1$;" IN A DARK,"
1145 PRINT #6;" "
1150 PRINT #6;" STINKY ";N4$;"."
1155 FOR I=1 TO 15ØØ:NEXT I
1156 GRAPHICS 2+16
116Ø PRINT #6;" ";N1$;" SNEAKED"
1165 PRINT #6;" "
117Ø PRINT #6;" OUT OF THE"
1175 PRINT #6;" "
118Ø PRINT #6;" ";N4$;" AND"
1185 PRINT #6;" "
119Ø PRINT #6;" GRABBED"
1195 PRINT #6;" "
12ØØ PRINT #6;" THE ";N3$;"'S"
12Ø1 PRINT #6;" "
12Ø2 PRINT #6;" MAGIC ";N5$;"."
121Ø FOR I=1 TO 15ØØ:NEXT I
122Ø GRAPHICS 2+16
123Ø PRINT #6;" ";N1$;" WAVED THE"
1235 PRINT #6;" "
124Ø PRINT #6;" ";N5$;" AND"
1245 PRINT #6;" "
1246 PRINT #6;" TURNED THE"
1247 PRINT #6;" "
125Ø PRINT #6;" ";N3$;" INTO "
1255 PRINT #6;" "
126Ø PRINT #6;" A FAT, LAZY"
1265 PRINT #6;" "
1266 PRINT #6;" ";N6$;"."
127Ø FOR I=1 TO 15ØØ:NEXT I
128Ø GRAPHICS 2+16
129Ø PRINT #6;" ";N1$;" CHASED THE"
1295 PRINT #6;" "
13ØØ PRINT #6;" ";N6$;" THROUGH"
13Ø1 PRINT #6;" "
13Ø2 PRINT #6;" THE ";N2$;"."
131Ø PRINT #6;" "
1311 PRINT #6;" THEN ";N1$;" RAN"
1315 PRINT #6;" "
132Ø PRINT #6;" ALL THE WAY"
1325 PRINT #6;" "
133Ø PRINT #6;" BACK HOME."
135Ø FOR I=1 TO 15ØØ:NEXT I
136Ø GRAPHICS 2+16
1365 SETCOLOR Ø,6,1Ø
137Ø POSITION 2,5:PRINT #6;"**  THE END  **'
```

```
1390 FOR I=1 TO 1500:NEXT I
1395 REM ***
1396 REM *** WOULD CHILD LIKE THE
1397 REM *** SAME STORY AGAIN?
1398 REM ***
1400 GRAPHICS 0
1402 PRINT "WOULD YOU LIKE TO SEE"
1403 PRINT "THE SAME STORY AGAIN";:INPUT ANS
     WER$
1404 IF ANSWER$="Y" THEN 1000
1405 IF ANSWER$<>"N" THEN 1400
1406 REM ***
1407 REM *** WOULD CHILD LIKE TO
1408 REM *** INVENT A NEW STORY?
1409 REM ***
1410 GRAPHICS 0
1411 PRINT "WOULD YOU LIKE TO"
1415 PRINT "INVENT A NEW STORY";:INPUT ANSWE
     R$
1420 IF ANSWER$="Y" THEN 535
1430 IF ANSWER$<>"N" THEN 1410
1435 GRAPHICS 0
1440 POSITION 6,8
1445 POKE 752,1
1450 PRINT "***  BYE!  BYE!  BYE!  ***"
1475 POKE 752,0
1480 REM ***
1490 REM *** RETURN TO FRIEND
1500 REM *** PROGRAM--ON DISK
1510 REM ***
1520 RUN "D:FRIEND"
```

Apple and IBM PC and PCjr

The Apple versions will run on all Apple computers with disk drives. The IBM versions will run on any PC and PCjr with disk drives. The programs must be saved with the following filenames:

The Talking Head program filename—FRIEND
The Story program filename—STORY

Program 5. Talking Head, IBC PC and PCjr Version

```
40 WIDTH 40:KEY OFF:COLOR 7,9,12
50 DIM A1$(30),A2$(10),A3$(11),A4$(20),A5$(3
0):GOSUB 10010
500 REM *** FRIEND WAKE-UP
510 IF NA$<>"" THEN 7040
515 GOSUB 1010:REM * FRIEND WAKE-UP
```

```
520 FOR K=1 TO 5:GOSUB 2015:REM * FRIEND INT
RODUCES HIMSELF/HERSELF
530 IF K=2 OR K=3 THEN GOSUB 3300:IF Z$="N"
THEN K=5: GOSUB 2015
540 IF K=3 AND Z$="Y" THEN K=5:A=K-1:GOTO 63
15
550 NEXT K
600 END
1000 REM *** FRIEND WAKE-UP
1010 GOSUB 5010:REM * DRAW FACE
1020 GOSUB 5410:REM *DRAW SLEEP EYES
1030 GOSUB 5210:REM *DRAW CLOSED MOUTH
1035 FOR P=1 TO 800:NEXT P
1040 GOSUB 4010:REM * WAKE-UP BELL
1050 GOSUB 5460:REM *DRAW OPEN EYES
1060 FOR P=1 TO 600:NEXT P
1070 GOSUB 5320:REM * WINK EYE
1080 FOR P=1 TO 100:NEXT P
1085 GOSUB 4820:REM * WINK NOISE
1090 GOSUB 5460:REM * DRAW OPEN EYES
1100 FOR P=1 TO 800:NEXT P
1120 RETURN
2000 REM *** FRIEND TALKS
2015 FOR I=1 TO SNUM(K)
2020 ON K GOSUB 3005,3025,3035,3045,3056:REM
 * FRIEND TALK * 1 SCREEN
2030 FOR P=1 TO 200:NEXT P
2040 NEXT I
2045 GOSUB 5510:REM * CLEAR MESSAGE WINDOW
2050 RETURN
3000 REM *** FRIEND TALKING -- 1 SCREEN
3005 IF A1$(I)="-1" THEN RETURN
3010 IF A1$(I)="-2" THEN GOSUB 5510:GOSUB 32
17:RETURN
3015 IF A1$(I)="*" THEN A1$(I)=NA$
3020 M$=A1$(I):GOSUB 3060:RETURN
3025 IF A2$(I)="-1" THEN RETURN
3030 M$=A2$(I):GOSUB 3060:RETURN
3035 IF A3$(I)="-1" THEN RETURN
3040 M$=A3$(I):GOSUB 3060:RETURN
3045 IF A4$(I)="-1" THEN RETURN
3050 IF A4$(I)="*" THEN A4$(I)=NA$
3055 M$=A4$(I):GOSUB 3060:RETURN
3056 IF A5$(I)="-1" THEN RETURN
3057 IF A5$(I)="BYE!" THEN M$=A5$(I):GOSUB 3
060:FOR J=1 TO 100:NEXT J:GOSUB 5510:RETURN
3058 IF A5$(I)="*" THEN A5$(I)=NA$
3059 M$=A5$(I):GOSUB 3060:RETURN
3060 GOSUB 5260:REM * OPEN MOUTH
3070 LOCATE 18,17:PRINT STRING$(10,32):LOCAT
E 18,17:PRINT M$
```

```
3075 GOSUB 4510:REM * FRIEND SOUND
3080 FOR P=1 TO 300:NEXT P:REM * KEEP MOUTH
OPEN
3090 GOSUB 5210:REM * CLOSE MOUTH
3095 FOR P=1 TO 300:NEXT P:REM * KEEP MOUTH
CLOSED
3110 RETURN
3200 REM * FRIEND ASKS CHILD A QUESTION
3217 FOR L=1 TO 9
3220 Z$=INKEY$:IF Z$="" THEN 3220
3222 LOCATE 18,16+L:PRINT Z$;:SOUND 400,1:N$
(L)=Z$:IF ASC(Z$)<>13 THEN NEXT L ELSE A=L:L
=9
3225 IF K=1 THEN 3245 ELSE 3267
3245 FOR J=1 TO A:NA$=NA$+N$(J):NEXT J
3267 GOSUB 5510:REM * CLEAR MESSAGE WINDOW
3290 RETURN
3300 Z$=INKEY$:IF Z$="" OR (Z$<>"Y" AND Z$<>
"N") THEN 3300
3310 RETURN
4000 REM LINE # 4000-4880 IS THE MUSIC ROUTI
NE
4010 FOR I=1000 TO 1200:SOUND I,.02:NEXT I:R
ETURN
4510 RANDOMIZE I:SOUND RND*900+100,2:RETURN
4820 SOUND 3000,2:RETURN
5000 REM * FRIEND'S FACE
5010 CLS:LOCATE 6,15,0:PRINT STRING$(11,220)

5020 LOCATE 7,14:PRINT STRING$(13,219)
5030 FOR L=8 TO 13:LOCATE L,14:PRINT CHR$(17
9);SPC(11);CHR$(179):NEXT L
5040 LOCATE 10,20:PRINT CHR$(179):LOCATE 11,
20:PRINT CHR$(4)
5050 LOCATE 12,17:PRINT CHR$(218);STRING$(5,
196);CHR$(191)
5060 LOCATE 13,18:PRINT STRING$(5,223)
5070 LOCATE 14,15:PRINT CHR$(92);SPC(9);CHR$
(47)
5080 LOCATE 15,16:PRINT CHR$(211);STRING$(7,
196);CHR$(189)
5150 RETURN
5200 REM * CLOSE MOUTH
5210 LOCATE 13,18:PRINT STRING$(5,223)
5230 RETURN
5250 REM * OPEN MOUTH
5260 LOCATE 13,18:PRINT CHR$(223);STRING$(3,
220);CHR$(223)
5280 RETURN
5300 REM * LEFT EYE WINK
```

```
5320 LOCATE 9,16:PRINT CHR$(148);SPC(7);CHR$
(95)
5330 FOR P=1 TO 150:NEXT P
5340 RETURN
5400 REM * EYES ASLEEP
5410 LOCATE 9,16:PRINT CHR$(95);SPC(7);CHR$(
95)
5440 RETURN
5450 REM * EYES AWAKE
5460 LOCATE 9,16:PRINT CHR$(148);SPC(7);CHR$
(148)
5470 RETURN
5500 REM * CLEAR MESSAGE WINDOW
5510 LOCATE 18,17:PRINT STRING$(10,32)
5550 RETURN
6000 REM * FRIEND'S GAME
6300 REM * PREPARE FRIEND'S MEMORY FOR EXIT
FREOM THE PROGRAM
6315 COMMON NA$
6335 CHAIN "A:STORY"
7000 REM *** FRIEND CALLED ON BEFORE
7040 GOSUB 5010:GOSUB 5210:GOSUB 5460:REM *
DRAW FRIEND
7050 K=4:GOSUB 2015
7060 FOR K=2 TO 3:GOSUB 2015:GOSUB 3300
7070 IF Z$="N" THEN K=5:GOSUB 2015:END
7080 IF K=3 AND Z$="Y" THEN 6315
7090 NEXT K
10010 FOR I=1 TO 22:READ A1$(I):NEXT I
10020 DATA HI,I'M,GED,-1,YOU,TURNED,ME,ON,-
1,WHO'S,OUT,THERE,-2
10040 DATA I'M,SO,HAPPY,-1,TO,SEE,YOU,*,-1
10050 FOR I=1 TO 8:READ A2$(I):NEXT I
10060 DATA DO,YOU,WANT,-1,TO,PLAY,A,GAME?
10070 FOR I=1 TO 10:READ A3$(I):NEXT I
11020 DATA DO,YOU,WANT,-1,TO,PLAY,-1,THE,STO
RY,GAME?
12009 FOR I=1 TO 19:READ A4$(I):NEXT I
12020 DATA HI,*,-1,I,HOPE,YOU,-1,HAD,FUN !!,
-1,I,WONDER,WHAT,-1,WE,SHOULD,DO,NOW.,-1
12030 FOR I=1 TO 24:READ A5$(I):NEXT I
12040 DATA *,I,AM,SORRY,-1,NONE,OF,THE,GAMES
,-1,LOOKED,FUN.,-1,MAYBE,WE,CAN,-1,PLAY,LATE
R.,-1,BYE!,BYE!,BYE!,-1
12050 FOR I=1 TO 5:READ SNUM(I):NEXT I
12060 DATA 22,8,10,19,24
12070 RETURN
```

Appendix A

Program 6. The Story Game, IBM PC and PCjr Version

```
40 CLS:COLOR 15,4,4
50 REM***************
55 REM THE STORY GAME
60 REM***************
70 REM*** THE PROGRAM HELPS CHILD AND PARENT

80 REM*** INVENT THEIR OWN FAIRY TALES.
510 REM*** PROGRAM INTRODUCES ITSELF
550 CLS:LOCATE 2,9,0:PRINT"*** THE STORY GAM
E ***"
560 FOR I=1 TO 500:NEXT I
810 REM*** PROGRAM ASKS FOR KEY INGREDIENTS
OF STORY
840 LOCATE 8,7:INPUT"NAME OF HERO";N1$
870 LOCATE 10,7:INPUT"ENCHANTED PLACE";N2$
890 LOCATE 12,7:INPUT"VILLAIN";N3$
910 LOCATE 14,7:INPUT"BAD PLACE";N4$
930 LOCATE 16,7:INPUT"MAGIC THING";N5$
940 LOCATE 18,7:INPUT"SILLY CREATURE";N6$
950 REM ***
960 REM *** PROGRAM TELLS STORY
970 REM ***
1003 CLS:LOCATE 10,9:PRINT N1$;" AND":LOCATE
 12,9:PRINT"THE MAGIC":LOCATE 14,9:PRINT N5$
1008 FOR I=1 TO 1500:NEXT I
1010 CLS:LOCATE 8,9:PRINT"ONCE UPON A TIME,"
1020 LOCATE 10,9:PRINT"A BRAVE CHILD NAMED "
;N1$
1040 LOCATE 12,9:PRINT"WENT EXPLORING"
1050 LOCATE 14,9:PRINT"IN AN ENCHANTED ";N2$
;"."
1060 FOR I=1 TO 5000:NEXT I
1080 CLS:LOCATE 8,9:PRINT"IN THE ";N2$
1090 LOCATE 10,9:PRINT"LIVED A HUGE, EVIL ";
N3$
1106 LOCATE 12,9:PRINT"WHO LOVED"
1110 LOCATE 14,9:PRINT"TO EAT CHILDERN."
1115 FOR I=1 TO 5000:NEXT I
1120 CLS:LOCATE 8,9:PRINT"THE ";N3$
1126 LOCATE 10,9:PRINT"TRAPPED ";N1$
1130 LOCATE 12,9:PRINT"AND PUT ";N1$
1150 LOCATE 14,9:PRINT"IN A DARK, STINKY ";N
4$;"."
1154 FOR I=1 TO 5000:NEXT I
1160 CLS:LOCATE 8,9:PRINT N1$;" SNEAKED"
1170 LOCATE 10,9:PRINT"OUT OF THE ";N4$
1200 LOCATE 12,9:PRINT"AND GRABBED THE ";N3$
;"’S"
```

```
1202 LOCATE 14,9:PRINT"MAGIC ";N5$;"."
1210 FOR I=1 TO 5000:NEXT I
1230 CLS:LOCATE 8,9:PRINT N1$;" WAVED THE"
1240 LOCATE 10,9:PRINT N5$;" AND"
1246 LOCATE 12,9:PRINT"TURNED THE ";N3$;" IN
TO"
1266 LOCATE 14,9:PRINT"A FAT, LAZY ";N6$;"."

1270 FOR I=1 TO 5000:NEXT I
1290 CLS:LOCATE 8,9:PRINT N1$;" CHASED THE "
;N6$
1302 LOCATE 10,9:PRINT"THROUGH THE ";N2$;"."

1311 LOCATE 12,9:PRINT"THEN ";N1$;" RAN"
1330 LOCATE 14,9:PRINT"ALL THE WAY BACK HOME
."
1350 FOR I=1 TO 5500:NEXT I
1370 CLS:LOCATE 12,13:PRINT"*** THE END ***"
1390 FOR I=1 TO 3500:NEXT I
1402 CLS:LOCATE 12,3:PRINT"LIKE TO SEE THE S
TORY AGAIN (Y/N)?"
1403 Z$=INKEY$:IF Z$="" OR (Z$<>"Y" AND Z$<>
"N") THEN 1403
1404 IF Z$="Y" THEN 1003
1405 IF Z$<>"N" THEN 1402
1410 CLS:LOCATE 12,3:PRINT"LIKE TO INVENT AN
OTHER STORY (Y/N)?"
1415 Z$=INKEY$:IF Z$="" OR (Z$<>"Y" AND Z$<>
"N") THEN 1415
1420 IF Z$="Y" THEN 550
1430 IF Z$<>"N" THEN 1415
1450 CLS:LOCATE 12,9:PRINT"** BYE! BYE! BYE!
 **"
1520 COMMON NA$:CHAIN "A:FRIEND"
```

Program 7. Talking Head, Apple Version

```
4Ø   HOME : DIM A1$(3Ø),A2$(1Ø),A3$(11),A4$(2
     Ø),A5$(3Ø): GOSUB 1ØØ1Ø
5Ø   GOSUB 12Ø8Ø
5ØØ   REM *** FRIEND WAKE-UP
51Ø   IF  PEEK (8Ø8) = 4 THEN  POKE 8Ø8,Ø: GOTO
      7Ø1Ø
515   GOSUB 1Ø1Ø: REM  * FRIEND WAKE-UP
52Ø   FOR K = 1 TO 5: GOSUB 2Ø15: REM  * FRIE
      ND INTRODUCES HIMSELF/HERSELF
53Ø   IF K = 2 OR K = 3 THEN  GOSUB 33ØØ: IF
      Z$ = "N" THEN K = 5: GOSUB 2Ø15
54Ø   IF K = 3 AND Z$ = "Y" THEN K = 5: NEXT
      K:K = K - 1: GOTO 6315
```

```
550  NEXT K: TEXT : HOME : END
600  END
1000  REM  *** FRIEND WAKE-UP
1010  GOSUB 5010: REM  * DRAW FACE
1020  GOSUB 5410: REM  * DRAW SLEEP EYES
1030  GOSUB 5210: REM  * DRAW CLOSED MOUTH
1035  FOR P = 1 TO 800: NEXT P
1040  GOSUB 4010: REM  * WAKE-UP BELL
1050  GOSUB 5460: REM  * DRAW OPEN EYES
1060  FOR P = 1 TO 600: NEXT P
1070  GOSUB 5320: REM  * WINK EYE
1080  FOR P = 1 TO 100: NEXT P
1085  GOSUB 4820: REM  *  WINK NOISE
1090  GOSUB 5460: REM  * DRAW OPEN EYES
1100  FOR P = 1 TO 800: NEXT P
1120  RETURN
2000  REM  *** FRIEND TALKS
2015  FOR I = 1 TO SNUM(K)
2020  ON K GOSUB 3005,3025,3035,3045,3056: REM
      * FRIEND TALKS --1 SCREEN
2030  FOR P = 1 TO 200: NEXT P
2040  NEXT I
2045  GOSUB 5510: REM  * CLEAR MESSAGE WINDO
     W
2050  RETURN
3000  REM  ** FRIEND TALKING -- 1 SCREEN
3005  IF A1$(I) = "-1" THEN  RETURN
3010  IF A1$(I) = "-2" THEN  GOSUB 5510: GOSUB
     3217: RETURN
3015  IF A1$(I) = "*" THEN A1$(I) = NAME$
3020 M$ = A1$(I): GOSUB 3060: RETURN
3025  IF A2$(I) = "-1" THEN  RETURN
3030 M$ = A2$(I): GOSUB 3060: RETURN
3035  IF A3$(I) = "-1" THEN  RETURN
3040 M$ = A3$(I): GOSUB 3060: RETURN
3045  IF A4$(I) = "-1" THEN  RETURN
3050  IF A4$(I) = "*" THEN A4$(I) = NAME$
3055 M$ = A4$(I): GOSUB 3060: RETURN
3056  IF A5$(I) = "-1" THEN  RETURN
3057  IF A5$(I) = "BYE!" THEN M$ = A5$(I): GOSUB
     3060: FOR J = 1 TO 100: NEXT J: GOSUB 5
     510: RETURN
3058  IF A5$(I) = "*" THEN A5$(I) = NAME$
3059 M$ = A5$(I): GOSUB 3060: RETURN
3060  GOSUB 5260: REM  * OPEN MOUTH
3070  VTAB 21: HTAB 19: PRINT "          ": VTAB
     21: HTAB 19: PRINT M$
3075  GOSUB 4510: REM  * FRIEND SOUND
3080  FOR P = 1 TO 300: NEXT P: REM  * KEEP
     MOUTH OPEN
```

```
3090  GOSUB 5210: REM  * CLOSE MOUTH
3095  FOR P = 1 TO 300: NEXT P: REM  * KEEP
     MOUTH CLOSED
3110  RETURN
3200  REM  * FRIEND ASKS CHILD A QUESTION
3217  FOR L = 1 TO 9
3220  GET Z$: IF Z$ = "" THEN 3220
3222  VTAB 21: HTAB 18 + L: PRINT Z$:N$(L) =
      Z$: IF  ASC (Z$) <  > 13 THEN  NEXT L
3225  IF K = 1 THEN  FOR J = 1 TO L:NAME$ =
      NAME$ + N$(J): NEXT J
3267  GOSUB 5510: REM  ** CLEAR MESSAGE WIND
      OW
3290  RETURN
3300  GET Z$: IF Z$ = "" OR (Z$ <  > "Y" AND
      Z$ <  > "N") THEN 3300
3310  RETURN
4010  FOR I = 1 TO 10: POKE 768,5: POKE 769,
      222: CALL 770: NEXT I: RETURN
4510  POKE 768,2: POKE 769, INT ( RND (1) *
      40 + 60): CALL 770: RETURN
4820  POKE 768,2: POKE 769,252: CALL 770: RETURN

5000  REM  *** FRIEND'S FACE
5010  HGR : HPLOT 112,40 TO 167,40: HPLOT 11
      1,41 TO 168,41: FOR I = 42 TO 44: HPLOT
      110,I TO 169,I: NEXT I
5020  FOR I = 45 TO 50: HPLOT 105,I TO 174,I
      : NEXT I: HPLOT 105,55 TO 110,50: HPLOT
      174,55 TO 169,50
5030  HPLOT 105,55 TO 105,95: HPLOT 174,55 TO
      174,95: HPLOT 105,95 TO 110,100: HPLOT
      174,95 TO 169,100
5040  HPLOT 112,100 TO 112,103: HPLOT 167,10
      0 TO 167,103: HPLOT 112,103 TO 167,103
5050  HPLOT 140,70 TO 140,80: HPLOT 110,62: HPLOT
      111,61: HPLOT 112,60 TO 118,60: HPLOT 1
      19,61: HPLOT 120,62
5060  HPLOT 169,62: HPLOT 168,61: HPLOT 167,
      60 TO 161,60: HPLOT 160,61: HPLOT 159,6
      2
5070  HPLOT 134,85 TO 146,85: HPLOT 134,85 TO
      134,87: HPLOT 146,85 TO 146,87: HPLOT 1
      36,87 TO 144,87
5150  RETURN
5200  REM  * CLOSE MOUTH
5210  HCOLOR= 0: HPLOT 134,89 TO 146,89: HCOLOR=
      3: HPLOT 136,87 TO 144,87
5230  RETURN
```

```
5250  REM  * OPEN MOUTH
5260  HCOLOR= Ø: HPLOT 136,87 TO 144,87: HCOLOR=
     3: HPLOT 134,89 TO 146,89
5280  RETURN
5300  REM  * LEFT EYE WINK
5320  HCOLOR= Ø: HPLOT 163,64 TO 165,66: HPLOT
     165,64 TO 163,66: HPLOT 164,66: HCOLOR=
     3: HPLOT 163,65 TO 165,65
5340  RETURN
5400  REM  * EYES ASLEEP
5410  HPLOT 114,65 TO 116,65: HPLOT 163,65 TO
     165,65
5440  RETURN
5450  REM  * EYES AWAKE
5460  HCOLOR= Ø: HPLOT 114,65 TO 116,65: HPLOT
     163,65 TO 165,65: HCOLOR= 3: HPLOT 114,
     64 TO 116,66: HPLOT 114,66 TO 116,64: HPLOT
     114,65
5465  HPLOT 163,64 TO 165,66: HPLOT 165,64 TO
     163,66: HPLOT 164,66
5470  RETURN
5500  REM  * CLEARN MESSAGE WINDOW
5510  HOME : RETURN
6000  REM  *** FRIEND'S GAMES
6300 REM *** PREPARE FRIEND'S MEMORY FOR E
     XIT FROM FRIEND GAME
6301  REM  *** STORE CHILD'S NAME IN LOCATIO
     NS 768-776
6315  FOR I = 1 TO  LEN (NAME$) - 1: POKE 79
     5 + I, ASC (N$(I)): NEXT I
6335  POKE 8Ø7, LEN (NAME$) - 1: POKE 8Ø8,K -
     1
64ØØ  REM  *** DISK VERSION OF FRIEND
64Ø8  REM  *** SELECT GAME PROGRAM/EXIT FRIE
     ND
642Ø  PRINT "RUN STORY": GOTO 642Ø
7ØØØ  REM  *** FRIEND CALLED ON BEFORE
7Ø1Ø NAME$ = "": FOR I = 1 TO  PEEK (8Ø7)
7Ø2Ø NAME$ = NAME$ +  CHR$ ( PEEK (795 + I))
     : NEXT I
7Ø4Ø  GOSUB 5Ø1Ø: GOSUB 521Ø: GOSUB 546Ø: REM
      * DRAW FRIEND
7Ø5Ø K = 4: GOSUB 2Ø15
7Ø6Ø  FOR K = 2 TO 3: GOSUB 2Ø15: GOSUB 33ØØ

7Ø7Ø  IF Z$ = "N" THEN K = 5: GOSUB 2Ø15: TEXT
     : HOME : END
7Ø8Ø  IF K = 3 AND Z$ = "Y" THEN 642Ø
7Ø9Ø  NEXT K
```

```
10Ø10  FOR I = 1 TO 22: READ A1$(I): NEXT I
10Ø20  DATA  HI,I'M,GED,-1,YOU,TURNED,ME,ON
     ,-1,WHO'S,OUT,THERE?,-2
10Ø40  DATA  I'M,SO,HAPPY,-1,TO,SEE,YOU,*,-1

10Ø50  FOR I = 1 TO 8: READ A2$(I): NEXT I
10Ø60  DATA  DO,YOU,WANT,-1,TO,PLAY,A,GAME?
10Ø70  FOR I = 1 TO 1Ø: READ A3$(I): NEXT I
11Ø20  DATA  DO,YOU,WANT,-1,TO,PLAY,-1,THE,S
     TORY,GAME?
12ØØ9  FOR I = 1 TO 19: READ A4$(I): NEXT I
12Ø1Ø  DATA  HI,*,-1,I,HOPE,YOU,-1,HAD,FUN!!
     ,-1,I,WONDER,WHAT,-1,WE,SHOULD,DO,NOW.,
     -1
12Ø3Ø  FOR I = 1 TO 24: READ A5$(I): NEXT I
12Ø4Ø  DATA  *,I,AM,SORRY,-1,NONE,OF,THE,GAM
     ES,-1,LOOKED,FUN.,-1,MAYBE,WE,CAN,-1,PL
     AY,LATER.,-1,BYE!,BYE!,BYE!,-1
12Ø6Ø  FOR I = 1 TO 5: READ SNUM(I): NEXT I
12Ø65  DATA  22,8,1Ø,19,24
12Ø7Ø  RETURN
12Ø8Ø  FOR I = 77Ø TO 795: READ M: POKE I,M:
      NEXT I: RETURN
12Ø9Ø  DATA  172,Ø1,Ø3,174,Ø1,Ø3,169,Ø4,32,1
     68,252,173,48,192,232,2Ø8,253,136,2Ø8,2
     39,2Ø6,Ø,Ø3,2Ø8,231,96
```

Program 8. The Story Game, Apple Version

```
5Ø  REM  **************
55  REM  THE STORY GAME
6Ø  REM  **************
65  REM  ***
7Ø  REM  *** PROGRAM HELPS CHILD AND PARENT
8Ø  REM  *** INVENT THEIR OWN FAIRY TALES.
51Ø  REM  *** PROGRAM INTRODUCES ITSELF
55Ø  TEXT : HOME : VTAB 2: HTAB 1Ø: PRINT "*
     ** THE STORY GAME ***"
56Ø  FOR I = 1 TO 5ØØ: NEXT
81Ø  REM  *** PROGRAM ASKS FOR KEY INGREDIEN
     TS OF STORY
84Ø  VTAB 6: HTAB 1Ø: INPUT "NAME OF HERO ";
     N1$
87Ø  VTAB 8: HTAB 1Ø: INPUT "ENCHANTED PLACE
      ";N2$
89Ø  VTAB 1Ø: HTAB 1Ø: INPUT "VILLAIN ";N3$
91Ø  VTAB 12: HTAB 1Ø: INPUT "BAD PLACE ";N4
     $
```

```
930  VTAB 14: HTAB 10: INPUT "MAGIC THING ";
     N5$
940  VTAB 16: HTAB 10: INPUT "SILLY CREATURE
      ";N6$
950  REM  ***
960  REM  PROGRAM TELLS STORY
970  REM  ***
1003  HOME : VTAB 10: HTAB 10: PRINT N1$;" A
     ND": VTAB 12: HTAB 10: PRINT "THE MAGIC
     ": VTAB 14: HTAB 10: PRINT N5$
1008  FOR I = 1 TO 1500: NEXT I
1010  HOME : VTAB 8: HTAB 6: PRINT "ONCE UPO
     N A TIME,"
1020  VTAB 10: HTAB 6: PRINT "A BRAVE CHILD
     NAMED ";N1$
1040  VTAB 12: HTAB 6: PRINT "WENT EXPLORING
     "
1050  VTAB 14: HTAB 6: PRINT "IN AN ENCHANTE
     D ";N2$;"."
1060  FOR I = 1 TO 4000: NEXT I
1080  HOME : VTAB 8: HTAB 6: PRINT "IN THE "
     ;N2$
1090  VTAB 10: HTAB 6: PRINT "LIVED A HUGE,
     EVIL ";N3$
1106  VTAB 12: HTAB 6: PRINT "WHO LOVED"
1110  VTAB 14: HTAB 6: PRINT "TO EAT CHILDER
     N."
1115  FOR I = 1 TO 4000: NEXT I
1120  HOME : VTAB 8: HTAB 6: PRINT "THE ";N3
     $
1126  VTAB 10: HTAB 6: PRINT "TRAPPED ";N1$
1130  VTAB 12: HTAB 6: PRINT "AND PUT ";N1$
1150  VTAB 14: HTAB 6: PRINT "IN A DARK, STI
     NKY ";N4$;"."
1154  FOR I = 1 TO 4000: NEXT I
1160  HOME : VTAB 8: HTAB 6: PRINT N1$;" SNE
     AKED"
1170  VTAB 10: HTAB 6: PRINT "OUT OF THE ";N
     4$
1200  VTAB 12: HTAB 6: PRINT "AND GRABBED TH
     E ";N3$;"'S"
1202  VTAB 14: HTAB 6: PRINT "MAGIC ";N5$;".
     "
1210  FOR I = 1 TO 4000: NEXT I
1230  HOME : VTAB 8: HTAB 6: PRINT N1$;" WAV
     ED THE"
1240  VTAB 10: HTAB 6: PRINT N5$;" AND"
1246  VTAB 12: HTAB 6: PRINT "TURNED THE ";N
     3$;" INTO"
```

```
1266  VTAB 14: HTAB 6: PRINT "A FAT, LAZY ";
    N6$;"."
127Ø  FOR I = 1 TO 4ØØØ: NEXT I
129Ø  HOME : VTAB 8: HTAB 6: PRINT N1$;" CHA
    SED THE ";N6$
13Ø2  VTAB 1Ø: HTAB 6: PRINT "THROUGH THE ";
    N2$;"."
1311  VTAB 12: HTAB 6: PRINT "THEN ";N1$;" R
    AN"
133Ø  VTAB 14: HTAB 6: PRINT "ALL THE WAY HO
    ME."
135Ø  FOR I = 1 TO 4ØØØ: NEXT I
137Ø  HOME : VTAB 12: HTAB 14: PRINT "** THE
      END **"
139Ø  FOR I = 1 TO 4ØØØ: NEXT I
14Ø2  HOME : VTAB 12: HTAB 2: PRINT "LIKE TO
      SEE THE STORY AGAIN (Y/N)?"
14Ø3  GET Z$: IF Z$ = "" OR (Z$ <  > "Y" AND
    Z$ <  > "N") THEN 14Ø3
14Ø4  IF Z$ = "Y" THEN 1ØØ3
14Ø5  IF Z$ <  > "N" THEN 14ØØ
141Ø HOME : VTAB 12: HTAB 2: PRINT "LIKE TO
      INVENT ANOTHER STORY (Y/N)?"
1415  GET Z$: IF Z$ = "" OR (Z$ <  > "N" AND
    Z$ <  > "Y") THEN 1415
142Ø  IF Z$ = "Y" THEN 55Ø
143Ø  IF Z$ <  > "N" THEN 1415
145Ø  HOME : VTAB 12: HTAB 12: PRINT "BYE!
    BYE!  BYE!"
152Ø  PRINT "RUN FRIEND"
```

TI-99/4A

The TI version requires the use of the Extended BASIC Cartridge. The version as written will work with the disk drive using the filenames:

The Talking Head Program filename—FRIEND
The Story Program filename—STORY

If you are using a tape rather than a disk, make the following changes to the programs:

The Talking Head Program

Replace 6420 with:

6420 RUN "CS1"

Add the following line:

6410 CALL CLEAR::PRINT "PLEASE INSERT THE STORY CASSETTE."

The Story Program:

Replace 1520 with

1520 CALL CHAR (129,"1111111111111111")::RUN "CS1"

Add the following line:

1510 PRINT "PLEASE INSERT THE FRIEND CASSETTE"

Program 9. Talking Head, TI Extended BASIC Version

```
30 CALL CLEAR
40 CALL COLOR(1,2,8)
50 DIM A1$(24),A2$(10),A3$(11),A4$(20),A5$(2
   5),SNUM(5)
55 RESTORE
60 GOSUB 10010
400 CALL CHARPAT(129,B$):: IF B$="1111111111
    111111" THEN B$="" :: CALL CHAR(129,B$):
    : GOTO 7005
500 REM** FRIEND MASTER**
514 REM*FRIEND WAKE-UP
515 GOSUB 1005
520 FOR K=1 TO 5
522 REM FRIEND INTRODUCES HIMSELF/HERSELF
523 GOSUB 2015
530 IF (K=2)OR(K=3)THEN GOSUB 3300 :: IF Z=7
    8 THEN K=5 :: GOSUB 2015 :: END
540 IF K=3 AND Z=89 THEN K=5
542 NEXT K :: K=K-1 :: GOTO 6315
600 CALL CLEAR :: END
1000 REM**FRIEND WAKE-UP**
1004 REM*DEFINE CHARACTERS
1005 GOSUB 4040
1010 REM*DRAW FACE
1015 GOSUB 5010
1020 REM*DRAW SLEEP EYES
1025 GOSUB 5410
1030 REM*DRAW CLOSED MOUTH
1032 GOSUB 5210
1035 FOR P=1 TO 200
1036 NEXT P
1040 GOSUB 8010 :: REM*WAKE-UP BELL
1045 GOSUB 4010
1050 REM*DRAW OPEN EYES
1055 GOSUB 5460
1060 FOR P=1 TO 600
1061 NEXT P
```

```
1070 REM*WINK EYES
1075 GOSUB 5310
1080 FOR P=1 TO 100
1082 NEXT P
1085 GOSUB 8820 :: REM*WINK NOISE
1088 REM GOSUB 4820
1090 REM*DRAW OPEN EYES
1095 GOSUB 5460
1100 FOR P=1 TO 200
1110 NEXT P
1120 RETURN
2000 REM***FRIEND TALKS
2015 FOR I=1 TO SNUM(K)
2020 ON K GOSUB 3005,3025,3035,3045,3057
2030 GOSUB 5510
2040 NEXT I
2050 RETURN
3000 REM*** FRIEND TALKING --1 SCREEN
3005 IF A1$(I)<>"-1" THEN 3010
3006 RETURN
3010 IF A1$(I)<>"-2" THEN 3015
3011 GOSUB 5510
3012 GOSUB 3217
3013 RETURN
3015 IF A1$(I)<>"*" THEN 3020
3017 A1$(I)=NAME$
3020 M$=A1$(I)
3021 GOSUB 3065
3022 RETURN
3025 IF A2$(I)<>"-1" THEN 3030
3026 RETURN
3030 M$=A2$(I)
3031 GOSUB 3065
3032 RETURN
3035 IF A3$(I)<>"-1" THEN 3040
3036 RETURN
3040 M$=A3$(I)
3041 GOSUB 3065
3042 RETURN
3045 IF A4$(I)<>"-1" THEN 3050
3046 RETURN
3050 IF A4$(I)<>"*" THEN 3052
3051 A4$(I)=NAME$
3052 M$=A4$(I)
3053 GOSUB 3065
3054 RETURN
3057 IF A5$(I)<>"-1" THEN 3059
3058 RETURN
3059 IF A5$(I)<>"*" THEN 3061
3060 A5$(I)=NAME$
```

```
3061 M$=A5$(I)
3062 GOSUB 3065
3063 RETURN
3064 REM*OPEN MOUTH
3065 GOSUB 5260
3070 DISPLAY AT(19,12):M$
3074 REM*FRIEND SOUND
3075 GOSUB 8510
3080 FOR P=1 TO 300
3081 NEXT P
3085 REM*CLOSE MOUTH
3090 GOSUB 5210
3095 REM FOR P=1 TO 300
3096 REM NEXT P
3100 RETURN
3200 REM*FRIEND ASK QUESTION
3217 A$="" :: FOR L=1 TO 9
3220 CALL KEY(0,Z,S)
3222 IF S=0 THEN 3220
3223 N$(L)=CHR$(Z):: A$=A$&STR$(Z)
3224 CALL HCHAR(19,13+L,Z)
3225 IF Z=13 THEN M=L :: L=9
3226 NEXT L :: A$=SEG$(A$,1,LEN(A$)-2)
3227 IF K<>1 THEN 3267
3230 NAME$="" :: FOR J=1 TO M
3235 NAME$=NAME$&N$(J)
3240 NEXT J
3266 REM*CLEAR MESSAGE WINDOW
3267 GOSUB 5510
3290 RETURN
3300 CALL KEY(0,Z,S)
3310 IF S=0 THEN 3300
3320 IF (Z<>89)*(Z<>78)THEN 3300
3330 RETURN
3410 REM*LINE #3420-4020 IS THE SOUND ROUTIN
     E
4010 RETURN
4030 REM*DEFINE CHARACTERS
4040 CALL CHAR(97,"003C424242423C00")
4050 CALL CHAR(99,"000000FF00000000")
4060 CALL CHAR(100,"0808080700000000")
4070 CALL CHAR(101,"101010E000000000")
4080 CALL CHAR(102,"8040201008040201")
4090 CALL CHAR(103,"0102040810204080")
4100 CALL CHAR(104,"55AA55AA55AA55AA")
4110 CALL CHAR(105,"000000000055AA55")
4120 CALL CHAR(106,"80C0E0F0F8FCFEFF")
4130 CALL CHAR(107,"0103070F1F3F7FFF")
4140 CALL CHAR(108,"0000000708080800")
4150 CALL CHAR(109,"000000E010101000")
```

```
4160 CALL CHAR(136,"FFFFFFFFFFFFFFFF")
4170 CALL CHAR(110,"8080808080808080")
4180 CALL CHAR(111,"0101010101010101")
4190 CALL CHAR(112,"FF00000000000000")
4200 RETURN
5000 REM*FRIEND'S FACE
5010 CALL CLEAR
5011 CALL HCHAR(5,13,105,5)
5020 CALL HCHAR(6,12,107)
5022 CALL HCHAR(6,13,136,5)
5024 CALL HCHAR(6,18,106)
5030 CALL HCHAR(7,9,104,13)
5040 CALL HCHAR(8,10,103)
5042 CALL HCHAR(8,20,102)
5044 CALL HCHAR(8,11,108)
5046 CALL HCHAR(8,12,99)
5048 CALL HCHAR(8,13,109)
5050 CALL HCHAR(8,17,108)
5051 CALL HCHAR(8,18,99)
5052 CALL HCHAR(8,19,109)
5054 CALL HCHAR(9,10,110)
5056 CALL HCHAR(9,20,111)
5058 CALL HCHAR(9,12,99)
5060 CALL HCHAR(9,18,99)
5062 FOR I=10 TO 14
5064 CALL HCHAR(I,10,110)
5066 CALL HCHAR(I,20,111)
5068 NEXT I
5070 CALL HCHAR(15,10,102)
5072 CALL HCHAR(15,20,103)
5074 CALL HCHAR(16,11,100)
5076 CALL HCHAR(16,19,101)
5078 CALL HCHAR(16,12,99,7)
5080 CALL HCHAR(11,9,60)
5082 CALL HCHAR(11,21,62)
5084 CALL HCHAR(11,15,94)
5086 CALL HCHAR(13,13,108)
5088 CALL HCHAR(13,14,99,3)
5090 CALL HCHAR(13,17,109)
5100 CALL HCHAR(14,14,112,3)
5150 RETURN
5200 REM*CLOSE MOUTH
5210 CALL HCHAR(14,14,112,3)
5220 RETURN
5250 REM*OPEN MOUTH
5260 CALL HCHAR(14,14,100)
5265 CALL HCHAR(14,15,99)
5270 CALL HCHAR(14,16,101)
5280 RETURN
5300 REM*LEFT EYE WINK
```

```
5310 CALL HCHAR(9,18,99)
5320 FOR P=1 TO 150
5330 NEXT P
5340 RETURN
5400 REM*EYES ASLEEP
5410 CALL HCHAR(9,12,99)
5420 CALL HCHAR(9,18,99)
5440 RETURN
5450 REM*EYES AWAKE
5460 CALL HCHAR(9,12,97)
5465 CALL HCHAR(9,18,97)
5470 RETURN
5500 REM*CLEAR MESSAGE WINDOW
5510 DISPLAY AT(19,12):" "
5520 RETURN
6000 REM*PREPARE FRIEND'S MEMORY FOR EXIT
6300 REM*STORE FRIEND'S NAME IN CALL CHAR(12
     8,A$)
6315 CALL CHAR(128,A$)
6408 REM*SELECT GAME PROGRAM /EXIT COMPUTER
6420 RUN "DSK1.STORY"
7000 REM*FRIEND CALLED ON BEFORE
7005 CALL CHARPAT(128,A$)
7010 NAME$="" :: FOR I=1 TO 16 STEP 2 :: B$=
     CHR$(VAL(SEG$(A$,I,2))):: NAME$=NAME$&B
     $ :: NEXT I
7040 GOSUB 4040 :: GOSUB 5010 :: GOSUB 5210
     :: GOSUB 5460 :: REM*DRAW FRIEND
7050 K=4 :: GOSUB 2015
7060 FOR K=2 TO 3 :: GOSUB 2015 :: GOSUB 330
     0
7070 IF Z=78 THEN K=5 :: GOSUB 2015 :: END
7080 IF (K=3)AND(Z=89)THEN 6420
7090 NEXT K
8010 CALL SOUND(500,494,0,659,0,784,0):: RET
     URN
8510 RANDOMIZE I :: CALL SOUND(700,(RND*331)
     +110,0):: RETURN
8820 CALL SOUND(300,1760,0):: RETURN
10010 FOR I=1 TO 22
10012 READ A1$(I)
10014 NEXT I
10020 DATA HI,I'M,GED,-1,YOU,TURNED,ME,ON,-1
      ,WHO'S,OUT,THERE?,-2
10040 DATA I'M,SO,HAPPY,-1,TO,SEE,YOU,*,-1
10050 FOR I=1 TO 8
10052 READ A2$(I)
10054 NEXT I
10060 DATA DO,YOU,WANT,-1=TO,PLAY,A,GAME?
10070 FOR I=1 TO 10
```

```
10072 READ A3$(I)
10074 NEXT I
11020 DATA DO,YOU,WANT,-1,TO,PLAY,-1,THE,STO
      RY,GAME?
12009 FOR I=1 TO 19
12011 READ A4$(I)
12013 NEXT I
12020 DATA HI,*,-1,I,HOPE,YOU,-1,HAD,FUN !!,
      -1,I,WONDER,WHAT,-1,WE,SHOULD,DO,NOW.,
      -1
12030 FOR I=1 TO 24
12032 READ A5$(I)
12034 NEXT I
12040 DATA *,I,AM,SORRY,-1,NONE,OF,THE,GAMES
      ,-1,LOOKED,FUN.,-1,MAYBE,WE,CAN,-1
12050 DATA PLAY,LATER.,-1,BYE!,BYE!,BYE!,-1
12060 FOR I=1 TO 5
12062 READ SNUM(I)
12064 NEXT I
12065 DATA 22,8,10,19,24
12070 RETURN
```

Program 10. The Story Game, TI Extended BASIC Version

```
50 REM**************
55 REM THE STORY GAME
60 REM***************
70 REM*PROGRAM HELPS CHILD AND PARENT
80 REM*INVENT THEIR OWN FAIRY TALES
510 REM*PROGRAM INTRODUCES ITSELF
550 CALL CLEAR :: DISPLAY AT(2,4):"*** THE S
    TORY GAME ***"
560 FOR I=1 TO 100 :: NEXT I
810 REM*PROGRAM ASKS FOR KEY INGREDIENTS OF
    STORY
840 DISPLAY AT(6,4):"NAME OF HERO ?" :: ACCE
    PT AT(6,19)BEEP SIZE(8):N1$
870 DISPLAY AT(8,4):"ENCHANTED PLACE ?" :: A
    CCEPT AT(8,22)BEEP SIZE(8):N2$
890 DISPLAY AT(10,4):"VILLAIN ?" :: ACCEPT A
    T(10,14)BEEP SIZE(8):N3$
910 DISPLAY AT(12,4):"BAD PLACE ?" :: ACCEPT
     AT(12,16)BEEP SIZE(8):N4$
930 DISPLAY AT(14,4):"MAGIC THING ?" :: ACCE
    PT AT(14,18)BEEP SIZE(8):N5$
```

```
940 DISPLAY AT(16,4):"SILLY CREATURE ?" :: A
    CCEPT AT(16,21)BEEP SIZE(8):N6$
950 REM***
960 REM*** PROGRAM TELLS STORY
970 REM***
1003 CALL CLEAR :: DISPLAY AT(10,8):N1$;" AN
     D" :: DISPLAY AT(12,8):"THE MAGIC" :: D
     ISPLAY AT(14,8):N5$
1008 FOR I=1 TO 300 :: NEXT I
1010 CALL CLEAR :: DISPLAY AT(6,3):"ONCE UPO
     N A TIME, A BRAVE " :: DISPLAY AT(8,3):
     "CHILD NAMED ";N1$ :: DISPLAY AT(10,3):
     "WENT EXPLORING IN AN"
1050 DISPLAY AT(12,3):"ENCHANTED ";N2$;"."
1060 FOR I=1 TO 800 :: NEXT I
1080 CALL CLEAR :: DISPLAY AT(6,3):"IN THE "
     ;N2$;" LIVED" :: DISPLAY AT(8,3):"A HUG
     E EVIL ";N3$ :: DISPLAY AT(10,3):"WHO L
     OVED TO"
1110 DISPLAY AT(12,3):"EAT CHILDERN."
1115 FOR I=1 TO 800 :: NEXT I
1120 CALL CLEAR :: DISPLAY AT(6,3):"THE ";N3
     $;" TRAPPED" :: DISPLAY AT(8,3):N1$;" A
     ND PUT" :: DISPLAY AT(10,3):N1$;" IN A
     DARK,"
1150 DISPLAY AT(12,3):"AND STINKY ";N4$
1154 FOR I=1 TO 800 :: NEXT I
1160 CALL CLEAR :: DISPLAY AT(6,3):N1$;" SNE
     AKED OUT" :: DISPLAY AT(8,3):"OF THE ";
     N4$;" AND" :: DISPLAY AT(10,3):"GRABBED
      THE ";N3$;"'S"
1202 DISPLAY AT(12,3):"MAGIC ";N5$;"."
1210 FOR I=1 TO 800 :: NEXT I
1230 CALL CLEAR :: DISPLAY AT(6,3):N1$;" WAV
     ED THE" :: DISPLAY AT(8,3):N5$;" AND TU
     RNED" :: DISPLAY AT(10,3):"THE ";N3$;"
     INTO"
1266 DISPLAY AT(12,3):"A FAT, LAZY ";N6$;"."
1270 FOR I=1 TO 800 :: NEXT I
1290 CALL CLEAR :: DISPLAY AT(6,3):N1$;" CHA
     SED THE" :: DISPLAY AT(8,3):N6$;" THROU
     GH THE ";N2$;"." :: DISPLAY AT(10,3):"T
     HEN ";N1$;" RAN"
1330 DISPLAY AT(12,3):"ALL THE WAY HOME."
1350 FOR I=1 TO 800 :: NEXT I
1370 CALL CLEAR :: DISPLAY AT(12,6):"*** THE
      END ***" :: FOR I=1 TO 1000 :: NEXT I
1402 DISPLAY AT(12,2):"SEE THE STORY AGAIN (
     Y/N)?"
1403 CALL KEY(0,K,S):: IF S=0 THEN 1403
```

```
1404 IF K=89 THEN 1003
1405 IF K<>78 THEN 1403
1410 DISPLAY AT(12,2):"INVENT A NEW STORY (I
     /Q)?"
1415 CALL KEY(0,K,S):: IF S=0 THEN 1415
1420 IF K=73 THEN 550
1430 IF K<>81 THEN 1415
1450 CALL CLEAR :: DISPLAY AT(12,5):"BYE!
     {3 SPACES}BYE!{3 SPACES}BYE!"
1460 FOR I=1 TO 800 :: NEXT I
1520 CALL CHAR(129,"1111111111111111"):: RUN
      "DSK1.FRIEND"
```

Commodore 64 and VIC-20

The Commodore 64 and VIC-20 versions have been written for disk. The VIC-20 version requires at least a 3K expander (expanders which add 8K or more may also be used). Be sure to save the programs using the following filenames:

The Talking Head filename—K
The Story program filename—T

Make the following changes if you are using tape:

Talking Head program
Replace line 6420 with:

6420 PRINT"{CLR}":POKE 631,76:POKE 632,207:POKE 633,58:
POKE 634,131:POKE 198,4:END

Delete Line 6430

Add line 6410

6410 PRINT"{CLR}INSERT THE STORY CASSETTE IN
RECORDER"

The Story Program
Replace line 1520 with

1520 PRINT"{CLR}":POKE 631,76:POKE 632,207:POKE 633,58:
POKE 634,131:POKE 198,4:END

Add line 1510:

1510 PRINT "{CLR}INSERT THE TALKING HEAD
PROGRAM IN RECORDER"

Program 11. Talking Head, VIC Version

```
4Ø POKE 36879,8
5Ø DIM A1$(3Ø),A2$(1Ø),A3$(11),A4$(2Ø),A5$(3Ø),SNU
   M(5):GOSUB 1ØØ1Ø
6Ø VO=36878:T1=36874:T2=36875:T3=36876
5ØØ REM *** FRIEND MASTER
51Ø IF PEEK(687)=4THENPOKE687,Ø:GOTO 7Ø1Ø
515 GOSUB 1Ø1Ø:REM * FRIEND WAKE-UP
52Ø FOR K=1TO5:GOSUB 2Ø15:REM * FRIEND INTRODUCES
    {SPACE}HIMSELF/HERSELF
53Ø IFK=2ORK=3THENGOSUB 33ØØ:IFZ$="N"THENK=5:GOSUB
    2Ø15
54Ø IFK=3ANDZ$="Y"THENK=5:NEXT:K=K-1:GOTO 6315
55Ø NEXT K
6ØØ END
1ØØØ REM *** FRIEND WAKE-UP
1Ø1Ø GOSUB 5Ø1Ø:REM * DRAW FACE
1Ø2Ø GOSUB 541Ø:REM * DRAW SLEEP EYES
1Ø3Ø GOSUB 521Ø:REM * DRAW CLOSED MOUTH
1Ø35 FOR P=1 TO 8ØØ:NEXT P
1Ø4Ø GOSUB 4Ø1Ø:REM * WAKE-UP BELL
1Ø5Ø GOSUB 546Ø:REM * DRAW OPEN EYES
1Ø6Ø FOR P=1 TO 6ØØ:NEXT P
1Ø7Ø GOSUB 532Ø:REM * WINK EYE
1Ø8Ø FOR P=1 TO 1ØØ:NEXT P
1Ø85 GOSUB 482Ø:REM * WINK NOISE
1Ø9Ø GOSUB 546Ø:REM * DRAW OPEN EYES
11ØØ FOR P=1 TO 8ØØ:NEXT P
112Ø RETURN
2ØØØ REM *** FRIEND TALKS
2Ø15 FOR I=1 TO SNUM(K)
2Ø2Ø ONKGOSUB 3ØØ5,3Ø25,3Ø35,3Ø45,3Ø56:REM * FRIEN
     D TALK * 1 SCREEN
2Ø3Ø FOR P=1 TO 2ØØ:NEXT P
2Ø4Ø NEXT I
2Ø45 GOSUB 551Ø:REM * CLEAR MESSAGE WINDOW
2Ø5Ø RETURN
3ØØØ REM *** FRIEND TALKING -- 1 SCREEN
3ØØ5 IF A1$(I)="-1"THEN RETURN
3Ø1Ø IF A1$(I)="-2"THEN GOSUB551Ø:GOSUB 3217:RETUR
     N
3Ø15 IF A1$(I)="*" THEN A1$(I)=NAME$
3Ø2Ø M$=A1$(I):GOSUB 3Ø6Ø:RETURN
3Ø25 IF A2$(I)="-1"THEN RETURN
3Ø3Ø M$=A2$(I):GOSUB 3Ø6Ø:RETURN
3Ø35 IF A3$(I)="-1"THEN RETURN
3Ø4Ø M$=A3$(I):GOSUB 3Ø6Ø:RETURN
3Ø45 IF A4$(I)="-1"THEN RETURN
3Ø5Ø IFA4$(I)="*"THENA4$(I)=NAME$
```

```
3Ø55 M$=A4$(I):GOSUB 3Ø6Ø:RETURN
3Ø56 IF A5$(I)="-1"THEN RETURN
3Ø57 IFA5$(I)="BYE!"THEN M$=A5$(I):GOSUB3Ø6Ø:FORJ=
     1TO1ØØ:NEXT:GOSUB 551Ø:RETURN
3Ø58 IF A5$(I)="*"THEN A5$(I)=NAME$
3Ø59 M$=A5$(I):GOSUB 3Ø6Ø:RETURN
3Ø6Ø GOSUB 526Ø:LL=LEN(M$):REM * OPEN MOUTH
3Ø65 POKE214,16:PRINT:POKE211,1:PRINT"{22 SPACES}"
3Ø7Ø POKE 214,16:PRINT:POKE 211,INT((22-LL)/2):PRI
     NTM$
3Ø75 GOSUB 451Ø:REM * FRIEND SOUND
3Ø8Ø FOR P=1 TO 3ØØ:NEXT P:REM * KEEP MOUTH OPEN
3Ø9Ø GOSUB 521Ø:REM * CLOSE MOUTH
3Ø95 FOR P=1 TO 3ØØ:NEXTP:REM * KEEP MOUTH CLOSED
31ØØ REM
311Ø RETURN
32ØØ REM * FRIEND ASKS CHILD A QUESTION
3217 POKE214,16:PRINT:FOR L=1 TO 9:
322Ø GET Z$:IF Z$=""THEN 322Ø
3222 PRINTTAB(8+L);Z$;:N$(L)=Z$:IF ASC(Z$)<>13THEN
      NEXTL
3225 IFK=1THENFOR J=1 TO L:NAME$=NAME$+N$(J):NEXT
     {SPACE}J
3267 GOSUB 551Ø:REM * CLEAR MESSAGE WINDOW
329Ø RETURN
33ØØ GETZ$:IF Z$=""OR(Z$<>"Y"ANDZ$<>"N")THEN 33ØØ
331Ø RETURN
4ØØØ REM LINE # 4ØØØ-488Ø IS THE SOUND ROUTINE
4Ø1Ø POKEVO,15:FORM=1TO7:POKET2,17Ø:FORN=1TO6:NEXT
     N:POKET2,Ø
4Ø2Ø FORN=1TO3ØØ:NEXTN:POKET1,19Ø:FORN=1TO6:NEXTN:
     POKET1,Ø
4Ø3Ø FORN=1TO3ØØ:NEXTN:NEXTM:POKEVO,Ø:RETURN
451Ø POKEVO,15:POKET1,INT(35*RND(1)+22Ø):POKET1,Ø:
     POKEVO,Ø:RETURN
482Ø POKEVO,15:POKET3,25Ø:FORM=1TO5:NEXTM:POKET3,Ø
483Ø FORM=1TO5ØØ:NEXTM:POKET3,23Ø:FORM=1TO5:NEXTM:
     POKET3,Ø:POKEVO,Ø:RETURN
5ØØØ REM *** FRIEND'S FACE
5Ø1Ø PRINT"{CLR}":{3 SPACES}POKE 214,Ø:PRINT:POKE
     {SPACE}211,6:PRINT"{RED}[9 £]"
5Ø2Ø POKE 214,1:PRINT:POKE 211,5:PRINT"{RVS}£
     {9 SPACES}[*]{OFF}"
5Ø3Ø POKE 214,2:PRINT:POKE 211,2:PRINT"[M]
     [15 +][G]"
5Ø4Ø POKE 214,3:PRINT:POKE 211,3:PRINT"{WHT}N
     {2 SPACES}U*I{3 SPACES}U*I{2 SPACES}M"
5Ø5Ø POKE 214,4:PRINT:POKE 211,2:PRINT"[M]
     {15 SPACES}[G]"
```

```
5060 POKE 214,5:PRINT:POKE 211,2:PRINT"[M]
     {15 SPACES}[G]"
5070 POKE 214,6:PRINT:POKE 211,2:PRINT"[M]
     {7 SPACES}-{7 SPACES}[G]"
5080 POKE 214,7:PRINT:POKE 211,2:PRINT"<{7 SPACES}
     Z{7 SPACES}>"
5090 POKE 214,8:PRINT:POKE 211,2:PRINT"[M]
     {15 SPACES}[G]"
5100 POKE 214,9:PRINT:POKE 211,2:PRINT"[M]
     {5 SPACES}U***I{5 SPACES}[G]"
5110 POKE 214,10:PRINT:POKE 211,2:PRINT"[M]
     {15 SPACES}[G]"
5120 POKE 214,11:PRINT:POKE 211,3:PRINT"M
     {13 SPACES}N"
5130 POKE 214,12:PRINT:POKE 211,4:PRINT"J
     ***********K"
5140 POKE 214,10:PRINT:POKE 211,4:PRINT"{5 SPACES}
     [3 Y]{2 SPACES}"
5150 RETURN
5200 REM * CLOSE MOUTH
5210 POKE 214,10:PRINT:POKE 211,2:PRINT"{WHT}[M]
     {6 SPACES}[3 Y]{6 SPACES}[G]"
5220 POKE 214,9:PRINT:POKE 211,2:PRINT"[M]
     {5 SPACES}U***I{5 SPACES}[G]"
5230 RETURN
5250 REM * OPEN MOUTH
5260 POKE 214,9:PRINT:POKE 211,2:PRINT"{WHT}[M]
     {5 SPACES}U***I{5 SPACES}[G]"
5270 POKE 214,10:PRINT:POKE 211,2:PRINT"[M]
     {6 SPACES}J*K{6 SPACES}[G]"
5280 RETURN
5300 REM * LEFT EYE WINK
5320 POKE 214,4:PRINT:POKE 211,2:PRINT"[M]
     {4 SPACES}W{5 SPACES}-{4 SPACES}[G]"
5330 FOR P=1 TO 150:NEXT P
5340 RETURN
5400 REM * EYES ASLEEP
5410 POKE 214,4:PRINT:POKE 211,2:PRINT"[M]
     {4 SPACES}-{5 SPACES}-{4 SPACES}[G]"
5440 RETURN
5450 REM * EYES AWAKE
5460 POKE 214,4:PRINT:POKE 211,2:PRINT"[M]
     {4 SPACES}W{5 SPACES}W{4 SPACES}[G]"
5470 RETURN
5500 REM * CLEAR MESSAGE WINDOW
5510 POKE 214,16:PRINT:POKE 211,1:PRINT"
     {19 SPACES}":RETURN
5550 RETURN
6000 REM * FRIEND'S GAMES
```

```
63ØØ REM *** PREPARE FRIEND'S MEMORY FOR EXIT FROM
      FRIEND PROGRAM
63Ø1 REM *** STORE CHILD'S NAME
63Ø5 REM *** AND SET FL AS
63Ø6 REM *** FLAG THAT FRIEND HAS ALREADY
63Ø7 REM *** BEEN CALLED SINCE TURNING ON COMPUTER
6315 FORI=1TOLEN(NAME$)-1:POKE676+I,ASC(N$(I))-64:
     NEXT
6335 POKE686,LEN(NAME$)-1:POKE687,K-1
64ØØ REM *** DISK VERSION OF FRIEND
64Ø8 REM *** SELECT GAME PROGRAM/EXIT FRIEND
642Ø PRINT"{CLR}":POKE631,76:POKE632,2Ø7:POKE633,3
     4:POKE634,84:POKE635,34
643Ø POKE636,44:POKE637,56:POKE638,58:POKE639,131:
     POKE198,9:END
7ØØØ REM *** FRIEND CALLED ON BEFORE
7Ø1Ø NAME$="":FORI=1TOPEEK(686)
7Ø2Ø NAME$=NAME$+CHR$(PEEK(676+I)+64):NEXT
7Ø4Ø GOSUB 5Ø1Ø:GOSUB 521Ø:GOSUB 546Ø:REM * DRAW F
     RIEND
7Ø5Ø K=4:GOSUB 2Ø15:POKE687,Ø
7Ø6Ø FORK=2TO3:GOSUB 2Ø15:GOSUB 33ØØ
7Ø7Ø IF Z$="N"THEN K=5:GOSUB 2Ø15:END
7Ø8Ø IF K=3ANDZ$="Y"THENNEXT:GOTO 642Ø
7Ø9Ø NEXT
1ØØ1Ø FOR I=1 TO 22:READ A1$(I):NEXT I
1ØØ2Ø DATA HI,I'M,GED,-1,YOU,TURNED,ME,ON,-1,WHO'S
      ,OUT,THERE?,-2
1ØØ4Ø DATA I'M,SO,HAPPY,-1,TO,SEE,YOU,*,-1
1ØØ5Ø FOR I=1 TO 8:READ A2$(I):NEXT I
1ØØ6Ø DATA DO,YOU,WANT,-1,TO,PLAY,A,GAME?
1ØØ7Ø FORI=1TO1Ø:READ A3$(I):NEXT I
11Ø2Ø DATA DO,YOU,WANT,-1,TO,PLAY,-1,THE,STORY,GAM
      E?
12ØØ9 FOR I=1 TO 19:READ A4$(I):NEXT I
12Ø1Ø DATA HI,*,-1,I,HOPE,YOU,-1,HAD,FUN !!,-1,I,W
      ONDER,WHAT,-1,WE,SHOULD,DO
12Ø2Ø DATA NOW.,-1
12Ø3Ø FOR I=1 TO 24:READ A5$(I):NEXT I
12Ø4Ø DATA *,I,AM,SORRY,-1,NONE,OF,THE,GAMES,-1,LO
      OKED,FUN.,-1,MAYBE,WE,CAN,-1
12Ø5Ø DATA PLAY,LATER.,-1,BYE!,BYE!,BYE!,-1
12Ø6Ø FORI=1TO5:READ SNUM(I):NEXT
12Ø65 DATA22,8,1Ø,19,24
12Ø7Ø RETURN
```

Program 12. The Story Game, VIC Version

```
40 POKE36879,8
50 REM **************
55 REM THE STORY GAME
60 REM **************
65 REM ***
70 REM *** PROGRAM HELPS CHILD AND PARENT
80 REM *** INVENT THEIR OWN FAIRY TALES.
95 REM *** DATA SECTION
510 REM *** PROGRAM INTRODUCES ITSELF
550 PRINT"{CLR}":PRINT"{WHT}*** THE STORY GAME ***
    "
560 FOR I=1TO 500:NEXT
810 REM *** PROGRAM ASKS FOR KEY INGREDIENTS OF ST
    ORY
840 POKE214,3:PRINT:POKE211,1:INPUT"NAME OF HERO";
    N1$
870 POKE214,6:PRINT:POKE211,1:INPUT"ENCHANTED PLAC
    E";N2$
890 POKE214,9:PRINT:POKE211,1:INPUT"VILLAIN";N3$
910 POKE214,12:PRINT:POKE211,1:INPUT"BAD PLACE";N4
    $
930 POKE214,15:PRINT:POKE211,1:INPUT"MAGIC THING";
    N5$
940 POKE214,18:PRINT:POKE211,1:INPUT"SILLY CREATUR
    E";N6$
950 REM ***
960 REM *** PROGRAM TELLS STORY
970 REM ***
1003 PRINT"{CLR}":POKE214,9:PRINT:POKE211,1:PRINTN
     1$;" AND"
1004 POKE214,11:PRINT:POKE211,1:PRINT"THE MAGIC"
1005 POKE214,13:PRINT:POKE211,1:PRINTN5$
1008 FORI=1TO1500:NEXTI:PRINT"{CLR}"
1010 POKE214,4:PRINT:POKE211,1:PRINT"ONCE UPON A T
     IME,"
1020 POKE214,6:PRINT:POKE211,1:PRINT"A BRAVE CHILD
     "
1030 POKE214,8:PRINT:POKE211,1:PRINT"NAMED ";N1$
1040 POKE214,10:PRINT:POKE211,1:PRINT"WENT EXPLORI
     NG"
1050 POKE214,12:PRINT:POKE211,1:PRINT"IN AN ENCHAN
     TED"
1055 POKE214,14:PRINT:POKE211,1:PRINTN2$;"."
1060 FORI=1TO3500:NEXT
1070 PRINT"{CLR}"
1080 POKE214,4:PRINT:POKE211,1:PRINT"IN THE ";N2$
1090 POKE214,6:PRINT:POKE211,1:PRINT"LIVED A HUGE"
1100 POKE214,8:PRINT:POKE211,1:PRINT"EVIL ";N3$
```

```
11Ø6 POKE214,1Ø:PRINT:POKE211,1:PRINT"WHO LOVED"
111Ø POKE214,12:PRINT:POKE211,1:PRINT"TO EAT CHILD
     REN."
1115 FORI=1 TO 35ØØ:NEXT
1116 PRINT"{CLR}"
112Ø POKE214,4:PRINT:POKE211,1:PRINT"THE ";N3$
1126 POKE214,6:PRINT:POKE211,1:PRINT"TRAPPED ";N1$
113Ø POKE214,8:PRINT:POKE211,1:PRINT"AND PUT ";N1$
114Ø POKE214,1Ø:PRINT:POKE211,1:PRINT"IN A DARK,"
115Ø POKE214,12:PRINT:POKE211,1:PRINT"STINKY ";N4$
     ;"."
1154 FORI=1 TO 35ØØ:NEXTI
1156 PRINT"{CLR}"
116Ø POKE214,4:PRINT:POKE211,1:PRINTN1$;" SNEAKED"
117Ø POKE214,6:PRINT:POKE211,1:PRINT"OUT OF THE"
118Ø POKE214,8:PRINT:POKE211,1:PRINTN4$;" AND"
119Ø POKE214,1Ø:PRINT:POKE211,1:PRINT"GRABBED"
12ØØ POKE214,12:PRINT:POKE211,1:PRINT"THE ";N3$;"'
     S"
12Ø2 POKE214,14:PRINT:POKE211,1:PRINT"MAGIC ";N5$;
     "."
121Ø FORI=1TO35ØØ:NEXT
122Ø PRINT"{CLR}"
123Ø POKE214,4:PRINT:POKE211,1:PRINTN1$;" WAVED TH
     E"
124Ø POKE214,6:PRINT:POKE211,1:PRINTN5$;" AND"
1246 POKE214,8:PRINT:POKE211,1:PRINT"TURNED THE"
125Ø POKE214,1Ø:PRINT:POKE211,1:PRINTN3$;" INTO"
126Ø POKE214,12:PRINT:POKE211,1:PRINT"A FAT, LAZY"
1266 POKE214,14:PRINT:POKE211,1:PRINTN6$;"."
127Ø FORI=1TO35ØØ:NEXT
128Ø PRINT"{CLR}"
129Ø POKE214,4:PRINT:POKE211,1:PRINTN1$;" CHASED T
     HE"
13ØØ POKE214,6:PRINT:POKE211,1:PRINTN6$;" THROUGH"
13Ø2 POKE214,8:PRINT:POKE211,1:PRINT"THE ";N2$;"."
1311 POKE214,1Ø:PRINT:POKE211,1:PRINT"THEN ";N1$;"
      RAN"
132Ø POKE214,12:PRINT:POKE211,1:PRINT"ALL THE WAY"
133Ø POKE214,14:PRINT:POKE211,1:PRINT"BACK HOME."
135Ø FORI=1TO35ØØ:NEXT
137Ø PRINT"{CLR}":POKE214,12:PRINT:POKE211,4:PRINT
     "** THE END **"
139Ø FORI=1TO35ØØ:NEXT
14ØØ PRINT"{CLR}"
14Ø2 POKE214,12:PRINT:POKE211,Ø:PRINT"LIKE TO SEE
     {SPACE}THE STORY{6 SPACES}AGAIN (Y/N)?"
14Ø3 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THEN 14Ø3
14Ø4 IFZ$="Y"THEN 1ØØ3
```

```
1405 IFZ$<>"N"THEN 1400
1410 PRINT"{CLR}":POKE214,12:PRINT:POKE211,0:PRINT
     "LIKE TO INVENT A STORY{3 SPACES}(Y/N)?"
1415 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THEN 1415
1420 IFZ$="Y"THEN 550
1430 IFZ$<>"N"THEN 1410
1450 PRINT"{CLR}":POKE214,12:PRINT:POKE211,0:PRINT
     "** BYE!{2 SPACES}BYE!{2 SPACES}BYE! **"
1460 FORI=1TO1500:NEXT
1520 PRINT"{CLR}":POKE631,76:POKE632,207:POKE633,3
     4:POKE634,75:POKE635,34:
1530 POKE636,44:POKE637,56:POKE638,58:POKE639,131:
     POKE198,9
```

Program 13. Talking Head, Commodore 64 Version

```
40 POKE 53280,0:POKE 53281,0
50 DIM A1$(30),A2$(10),A3$(11),A4$(20),A5$(30),SNU
   M(5):GOSUB 10010
60 FORI=54272TO54296:POKEI,0:NEXT:POKE54296,15:VO=
   54276:AD=54277:SR=54278
70 HF=54273:LF=54272:POKEAD,15:POKESR,204
500 REM *** FRIEND MASTER
510 IF PEEK(49163)=4THEN 7010
515 GOSUB 1010:REM * FRIEND WAKE-UP
520 FOR K=1TO5:GOSUB 2015:REM * FRIEND INTRODUCES
    {SPACE}HIMSELF/HERSELF
530 IFK=2ORK=3THENGOSUB 3300:IFZ$="N"THENK=5:GOSUB
    2015
540 IFK=3ANDZ$="Y"THENK=5:NEXT:K=K-1:GOTO 6315
550 NEXT K
600 END
1000 REM *** FRIEND WAKE-UP
1010 GOSUB 5010:REM * DRAW FACE
1020 GOSUB 5410:REM * DRAW SLEEP EYES
1030 GOSUB 5210:REM * DRAW CLOSED MOUTH
1035 FOR P=1 TO 800:NEXT P
1040 GOSUB 4010:REM * WAKE-UP BELL
1050 GOSUB 5460:REM * DRAW OPEN EYES
1060 FOR P=1 TO 600:NEXT P
1070 GOSUB 5320:REM * WINK EYE
1080 FOR P=1 TO 100:NEXT P
1085 GOSUB 4820:REM * WINK NOISE
1090 GOSUB 5460:REM * DRAW OPEN EYES
1100 FOR P=1 TO 800:NEXT P
1120 RETURN
2000 REM *** FRIEND TALKS
2015 FOR I=1 TO SNUM(K)
```

```
2020 ONKGOSUB 3005,3025,3035,3045,3056:REM * FRIEN
     D TALK * 1 SCREEN
2030 FOR P=1 TO 200:NEXT P
2040 NEXT I
2045 GOSUB 5510:REM * CLEAR MESSAGE WINDOW
2050 RETURN
3000 REM *** FRIEND TALKING -- 1 SCREEN
3005 IF A1$(I)="-1"THEN RETURN
3010 IF A1$(I)="-2"THEN GOSUB5510:GOSUB 3217:RETUR
     N
3015 IF A1$(I)="*" THEN A1$(I)=NAME$
3020 M$=A1$(I):GOSUB 3060:RETURN
3025 IF A2$(I)="-1"THEN RETURN
3030 M$=A2$(I):GOSUB 3060:RETURN
3035 IF A3$(I)="-1"THEN RETURN
3040 M$=A3$(I):GOSUB 3060:RETURN
3045 IF A4$(I)="-1"THEN RETURN
3050 IF A4$(I)="*"THEN A4$(I)=NAME$
3055 M$=A4$(I):GOSUB 3060:RETURN
3056 IF A5$(I)="-1"THEN RETURN
3057 IFA5$(I)="BYE!"THEN M$=A5$(I):GOSUB3060:FORJ=
     1TO100:NEXT:GOSUB 5510:RETURN
3058 IF A5$(I)="*"THEN A5$(I)=NAME$
3059 M$=A5$(I):GOSUB 3060:RETURN
3060 GOSUB 5260:REM * OPEN MOUTH
3070 POKE 214,16:PRINT:POKE 211,19:PRINT"{YEL}
     {9 SPACES}";:PRINT"{9 LEFT}";M$
3075 GOSUB 4510:REM * FRIEND SOUND
3080 FOR P=1 TO 300:NEXT P:REM * KEEP MOUTH OPEN
3090 GOSUB 5210:REM * CLOSE MOUTH
3095 FOR P=1 TO 300:NEXTP:REM * KEEP MOUTH CLOSED
3100 REM
3110 RETURN
3200 REM * FRIEND ASKS CHILD A QUESTION
3217 POKE214,16:PRINT:FOR L=1 TO 9:
3220 GET Z$:IF Z$=""THEN 3220
3222 PRINTTAB(18+L);Z$;:N$(L)=Z$:IF ASC(Z$)<>13THE
     N NEXTL
3225 IFK=1THENFOR J=1 TO L:NAME$=NAME$+N$(J):NEXT
     {SPACE}J
3267 GOSUB 5510:REM * CLEAR MESSAGE WINDOW
3290 RETURN
3300 GETZ$:IF Z$=""OR(Z$<>"Y"ANDZ$<>"N")THEN 3300
3310 RETURN
4000 REM LINE # 4000-4880 IS THE SOUND ROUTINE
4010 POKEHF,70:POKELF,75:POKEVO,17:FORS=1TO400:NEX
     T:POKEVO,16:RETURN
4510 POKEHF,INT(20*(RND(0))+80):POKELF,INT(10*(RND
     (0))+40):POKEVO,17
```

```
4520 FORS=1TO25:NEXT:POKEVO,16:RETURN
4820 POKEHF,7Ø :POKELF,195:POKEVO,33:FORS=1TO3Ø:NE
     XT:POKEVO,32:RETURN
5ØØØ REM *** FRIEND'S FACE
5Ø1Ø PRINT"{CLR}":{3 SPACES}POKE 214,2:PRINT:POKE
     {SPACE}211,16:PRINT"{RED}⁅9 £⁆"
5Ø2Ø POKE 214,3:PRINT:POKE 211,15:PRINT"{RVS}£
     {9 SPACES}⁅*⁆{OFF}"
5Ø3Ø POKE 214,4:PRINT:POKE 211,12:PRINT"⁅M⁆
     ⁅15 +⁆⁅G⁆"
5Ø4Ø POKE 214,5:PRINT:POKE 211,13:PRINT"{WHT}N
     {2 SPACES}U*I{3 SPACES}U*I{2 SPACES}M"
5Ø5Ø POKE 214,6:PRINT:POKE 211,12:PRINT"⁅M⁆
     {15 SPACES}⁅G⁆"
5Ø6Ø POKE 214,7:PRINT:POKE 211,12:PRINT"⁅M⁆
     {15 SPACES}⁅G⁆"
5Ø7Ø POKE 214,8:PRINT:POKE 211,12:PRINT"⁅M⁆
     {7 SPACES}-{7 SPACES}⁅G⁆"
5Ø8Ø POKE 214,9:PRINT:POKE 211,12:PRINT"<
     {7 SPACES}Z{7 SPACES}>"
5Ø9Ø POKE 214,1Ø:PRINT:POKE 211,12:PRINT"⁅M⁆
     {15 SPACES}⁅G⁆"
51ØØ POKE 214,11:PRINT:POKE 211,12:PRINT"⁅M⁆
     {5 SPACES}U***I{5 SPACES}⁅G⁆"
511Ø POKE 214,12:PRINT:POKE 211,12:PRINT"⁅M⁆
     {15 SPACES}⁅G⁆"
512Ø POKE 214,13:PRINT:POKE 211,13:PRINT"M
     {13 SPACES}N"
513Ø POKE 214,14:PRINT:POKE 211,14:PRINT"J
     ***********K"
514Ø POKE 214,12:PRINT:POKE 211,14:PRINT"
     {5 SPACES}⁅3 Y⁆{2 SPACES}"
515Ø RETURN
52ØØ REM * CLOSE MOUTH
521Ø POKE 214,12:PRINT:POKE 211,12:PRINT"{WHT}
     ⁅M⁆{6 SPACES}⁅3 Y⁆{6 SPACES}⁅G⁆"
522Ø POKE 214,11:PRINT:POKE 211,12:PRINT"⁅M⁆
     {5 SPACES}U***I{5 SPACES}⁅G⁆"
523Ø RETURN
525Ø REM * OPEN MOUTH
526Ø POKE 214,11:PRINT:POKE 211,12:PRINT"{WHT}
     ⁅M⁆{5 SPACES}U***I{5 SPACES}⁅G⁆"
527Ø POKE 214,12:PRINT:POKE 211,12:PRINT"⁅M⁆
     {6 SPACES}J*K{6 SPACES}⁅G⁆"
528Ø RETURN
53ØØ REM * LEFT EYE WINK
532Ø POKE 214,6:PRINT:POKE 211,12:PRINT"⁅M⁆
     {4 SPACES}W{5 SPACES}-{4 SPACES}⁅G⁆"
533Ø FOR P=1 TO 15Ø:NEXT P
```

```
5340 RETURN
5400 REM * EYES ASLEEP
5410 POKE 214,6:PRINT:POKE 211,12:PRINT"{M}
     {4 SPACES}-{5 SPACES}-{4 SPACES}{G}"
5440 RETURN
5450 REM * EYES AWAKE
5460 POKE 214,6:PRINT:POKE 211,12:PRINT"{M}
     {4 SPACES}W{5 SPACES}W{4 SPACES}{G}"
5470 RETURN
5500 REM * CLEAR MESSAGE WINDOW
5510 POKE 214,16:PRINT:POKE 211,19:PRINT"
     {10 SPACES}":RETURN
5550 RETURN
6000 REM * FRIEND'S GAMES
6300 REM *** PREPARE FRIEND'S MEMORY FOR EXIT FROM
      FRIEND PROGRAM
6301 REM *** STORE CHILD'S NAME
6302 REM *** IN LOCATIONS
6303 REM *** 49152-49161
6304 REM *** (LENGTH OF NAME IN 49162)
6305 REM *** AND SET LOCATION 2 AS
6306 REM *** FLAG THAT FRIEND HAS ALREADY
6307 REM *** BEEN CALLED SINCE TURNING ON COMPUTER
6315 FOR I=1 TO LEN(NAME$)-1:POKE49151+I,ASC(N$(I)
     )-64:NEXT
6335 POKE 49162,LEN(NAME$)-1:POKE 49163,K-1
6400 REM *** DISK VERSION OF FRIEND
6408 REM *** SELECT GAME PROGRAM/EXIT FRIEND
6420 PRINT"{CLR}":POKE631,76:POKE632,207:POKE633,3
     4:POKE634,84
6430 POKE635,34:POKE636,44:POKE637,56:POKE638,58:P
     OKE639,131:POKE198,9:END
7000 REM *** FRIEND CALLED ON BEFORE
7010 NAME$="":FORI=1 TO PEEK(49162)
7020 NAME$=NAME$+CHR$(PEEK(49151+I)+64):NEXT
7040 GOSUB 5010:GOSUB 5210:GOSUB 5460:REM * DRAW F
     RIEND
7050 K=PEEK(49163):POKE49163,0:GOSUB 2015
7060 FORK=2TO3:GOSUB 2015:GOSUB 3300
7070 IF Z$="N"THEN K=5:GOSUB 2015:END
7080 IF K=3ANDZ$="Y"THENNEXT:GOTO 6420
7090 NEXT
10010 FOR I=1 TO 22:READ A1$(I):NEXT I
10020 DATA HI,I'M,GED,-1,YOU,TURNED,ME,ON,-1,WHO'S
      ,OUT THERE?,-2
10040 DATA I'M,SO,HAPPY,-1,TO,SEE,YOU,*,-1
10050 FOR I=1 TO 8:READ A2$(I):NEXT I
10060 DATA DO,YOU,WANT,-1,TO,PLAY,A,GAME?
10070 FORI=1TO10:READ A3$(I):NEXT I
```

```
11Ø2Ø DATA DO,YOU,WANT,-1,TO,PLAY,-1,THE,STORY,GAM
      E?
12ØØ9 FOR I=1 TO 19:READ A4$(I):NEXT I
12Ø1Ø DATA HI,*,-1,I,HOPE,YOU,-1,HAD,FUN !!,-1,I,W
      ONDER,WHAT,-1,WE,SHOULD,DO
12Ø2Ø DATA NOW.,-1
12Ø3Ø FOR I=1 TO 24:READ A5$(I):NEXT I
12Ø4Ø DATA *,I,AM,SORRY,-1,NONE,OF,THE,GAMES,-1,LO
      OKED,FUN.,-1,MAYBE,WE,CAN,-1
12Ø5Ø DATA PLAY,LATER.,-1,BYE!,BYE!,BYE!,-1
12Ø6Ø FORI=1TO5:READ SNUM(I):NEXT
12Ø65 DATA22,8,1Ø,19,24
12Ø7Ø RETURN
```

Program 14. The Story Game, Commodore 64 Version

```
4Ø POKE5328Ø,Ø:POKE53281,Ø
5Ø REM **************
55 REM THE STORY GAME
6Ø REM **************
65 REM ***
7Ø REM *** PROGRAM HELPS CHILD AND PARENT
8Ø REM *** INVENT THEIR OWN FAIRY TALES.
95 REM *** DATA SECTION
51Ø REM *** PROGRAM INTRODUCES ITSELF
55Ø PRINT"{CLR}":POKE214,1:PRINT:POKE211,9:PRINT"*
    ** THE STORY GAME ***"
56Ø FOR I=1TO 5ØØ:NEXT
81Ø REM *** PROGRAM ASKS FOR KEY INGREDIENTS OF ST
    ORY
84Ø POKE214,5:PRINT:POKE211,12:INPUT"NAME OF HERO"
    ;N1$
87Ø POKE214,7:PRINT:POKE211,12:INPUT"ENCHANTED PLA
    CE";N2$
89Ø POKE214,9:PRINT:POKE211,12:INPUT"VILLAIN";N3$
91Ø POKE214,11:PRINT:POKE211,12:INPUT"BAD PLACE";N
    4$
93Ø POKE214,13:PRINT:POKE211,12:INPUT"MAGIC THING"
    ;N5$
94Ø POKE214,15:PRINT:POKE211,12:INPUT"SILLY CREATU
    RE";N6$
95Ø REM ***
96Ø REM *** PROGRAM TELLS STORY
97Ø REM ***
1ØØ3 PRINT"{CLR}":POKE214,12:PRINT:POKE211,9:PRINT
     N1$;" AND THE MAGIC ";N5$
1ØØ8 FORI=1TO15ØØ:NEXTI:PRINT"{CLR}"
1Ø1Ø POKE214,8:PRINT:POKE211,11:PRINT"ONCE UPON A
     {SPACE}TIME,"
```

```
1020 POKE214,10:PRINT:POKE211,11:PRINT"A BRAVE CHI
     LD NAMED ";N1$
1040 POKE214,12:PRINT:POKE211,11:PRINT"WENT EXPLOR
     ING"
1050 POKE214,14:PRINT:POKE211,11:PRINT"IN AN ENCHA
     NTED ";N2$;"."
1060 FORI=1TO3500:NEXT
1070 PRINT"{CLR}"
1080 POKE214,8:PRINT:POKE211,11:PRINT"IN THE ";N2$
1090 POKE214,10:PRINT:POKE211,11:PRINT"LIVED A HUG
     E, EVIL ";N3$
1106 POKE214,12:PRINT:POKE211,11:PRINT"WHO LOVED"
1110 POKE214,14:PRINT:POKE211,11:PRINT"TO EAT CHIL
     DREN."
1115 FORI=1 TO 3500:NEXT
1116 PRINT"{CLR}"
1120 POKE214,8:PRINT:POKE211,11:PRINT"THE ";N3$
1126 POKE214,10:PRINT:POKE211,11:PRINT"TRAPPED ";N
     1$
1130 POKE214,12:PRINT:POKE211,11:PRINT"AND PUT ";N
     1$
1150 POKE214,14:PRINT:POKE211,11:PRINT"IN A DARK,
     {SPACE}STINKY ";N4$;"."
1154 FORI=1 TO 3500:NEXTI
1156 PRINT"{CLR}"
1160 POKE214,8:PRINT:POKE211,11:PRINTN1$;" SNEAKED
     "
1170 POKE214,10:PRINT:POKE211,11:PRINT"OUT OF THE
     {SPACE}";N4$
1200 POKE214,12:PRINT:POKE211,11:PRINT"AND GRABBED
      THE ";N3$;"'S"
1202 POKE214,14:PRINT:POKE211,11:PRINT"MAGIC ";N5$
     ;"."
1210 FORI=1TO3500:NEXT
1220 PRINT"{CLR}"
1230 POKE214,8:PRINT:POKE211,11:PRINTN1$;" WAVED T
     HE"
1240 POKE214,10:PRINT:POKE211,11:PRINTN5$;" AND"
1246 POKE214,12:PRINT:POKE211,11:PRINT"TURNED THE
     {SPACE}";N3$;" INTO"
1266 POKE214,14:PRINT:POKE211,11:PRINT"A FAT, LAZY
      ";N6$;"."
1270 FORI=1TO3500:NEXT
1280 PRINT"{CLR}"
1290 POKE214,8:PRINT:POKE211,11:PRINTN1$;" CHASED
     {SPACE}THE ";N6$
1302 POKE214,10:PRINT:POKE211,11:PRINT"THROUGH THE
      ";N2$;"."
1311 POKE214,12:PRINT:POKE211,11:PRINT"THEN ";N1$;
     " RAN"
```

```
1330 POKE214,14:PRINT:POKE211,11:PRINT"ALL THE WAY
      BACK HOME."
1350 FORI=1TO3500:NEXT
1370 PRINT"{CLR}":POKE214,12:PRINT:POKE211,13:PRIN
     T"** THE END **"
1390 FORI=1TO3500:NEXT
1400 PRINT"{CLR}"
1402 POKE214,12:PRINT:POKE211,3:PRINT"LIKE TO SEE
     {SPACE}THE STORY AGAIN (Y/N)?"
1403 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THEN 1403
1404 IFZ$="Y"THEN 1003
1405 IFZ$<>"N"THEN 1400
1410 PRINT"{CLR}":POKE214,12
1412 PRINT:POKE211,3:PRINT "LIKE TO INVENT A NEW S
     TORY (Y/N)?"
1415 GETZ$:IFZ$=""OR(Z$<>"Y"ANDZ$<>"N")THEN 1415
1420 IFZ$="Y"THEN 550
1430 IFZ$<>"N"THEN 1410
1450 PRINT"{CLR}":POKE214,12:PRINT:POKE211,8:PRINT
     "** BYE!{2 SPACES}BYE!{2 SPACES}BYE! **"
1520 PRINT"{CLR}":POKE631,76:POKE632,207:POKE633,3
     4:POKE634,75:POKE635,34
1530 POKE636,44:POKE637,56:POKE638,58:POKE639,131:
     POKE198,9
```

How to Type In Programs

Some of the programs listed in this book contain special control characters (cursor control, color keys, inverse video, etc.). To make it easy to tell exactly what to type when entering one of these programs into your computer, we have established the following listing conventions. There is a separate key for each computer. Refer to the appropriate tables when you come across an unusual symbol in a program listing. If you are unsure how to actually enter a control character, consult your computer's manuals.

Atari

Characters in inverse video will appear like: INVERSE VIDEO
Enter these characters with the Atari logo key, {⅄}.

When you see	Type	See	
{CLEAR}	ESC SHIFT <	↖	Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE	◀	Backspace
{DELETE}	ESC CTRL DELETE	⍃	Delete Character
{INSERT}	ESC CTRL INSERT	⍄	Insert Character
{DEL LINE}	ESC SHIFT DELETE	⍓	Delete Line
{INS LINE}	ESC SHIFT INSERT	⍌	Insert Line
{TAB}	ESC TAB	▶	TAB key
{CLR TAB}	ESC CTRL TAB	⍇	Clear TAB
{SET TAB}	ESC SHIFT TAB	⍈	Set TAB stop
{BELL}	ESC CTRL 2	⍁	Ring Buzzer
{ESC}	ESC ESC	␛	ESCape key

Graphics characters, such as CTRL-T, the ball character ● will appear as the "normal" letter enclosed in braces, e.g., {T}.

A series of identical control characters, such as 10 spaces, three-cursor-lefts, or 20 CTRL-R's, will appear as {10 SPACES}, {3 LEFT}, {20 R}, etc. If the character in braces is in inverse video, that character or characters should be entered with the Atari logo key. For example, {5 ▣ } means to enter five inverse-video CTRL-U's.

Commodore 64 and VIC

Program listings will contain words within braces which spell out any special characters: {DOWN} would mean to press the cursor down key. {5 SPACES} would mean to press the space bar five times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listings. For example, S̲ would mean to type the S key while holding the SHIFT key. If you find an underlined key enclosed in braces (e.g., {10 N̲}), you should type the key as many times as indicated (in our example, you would enter ten shifted N's).

If a key is enclosed in special brackets, |< >|, you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as indicated.

Rarely, you'll see in a Commodore 64 program a solitary letter of the alphabet enclosed in braces. These characters can be entered by holding down the CTRL key while typing the letter in the braces. For example, {A} would indicate that you should press CTRL-A.

About the *quote mode*: You should know that you can move the cursor around the screen with the CRSR keys. Sometimes a programmer will want to move the cursor under program control. That's why you see all the {LEFT}'s, {HOME}'s, and {BLU}'s in our programs. The only way the computer can tell the difference between direct and programmed cursor control is the quote mode.

Once you press the quote (the double quote, SHIFT-2), you are in the quote mode. If you type something and then try to change it by moving the cursor left, you'll only get a bunch of reverse-video lines. These are the symbols for cursor left. The only editing key that isn't programmable is the DEL key; you can still use DEL to back up and edit the line. Once you type another quote, you are out of quote mode.

You also go into quote mode when you INserT spaces into a line. In any case, the easiest way to get out of quote mode is to just press RETURN. You'll then be out of quote mode and you can cursor up to the mistyped line and fix it.

Use the following tables when entering special characters:

Appendix B

When You Read:	Press:		See:	When You Read:	Press:		See:
{CLR}	SHIFT	CLR/HOME		{GRN}	CTRL	6	
{HOME}		CLR/HOME		{BLU}	CTRL	7	
{UP}	SHIFT	↑ CRSR ↓		{YEL}	CTRL	8	
{DOWN}		↑ CRSR ↓		{F1}	f1		
{LEFT}	SHIFT	← CRSR →		{F2}	f2		
{RIGHT}		← CRSR →		{F3}	f3		
{RVS}	CTRL	9		{F4}	f4		
{OFF}	CTRL	0		{F5}	f5		
{BLK}	CTRL	1		{F6}	f6		
{WHT}	CTRL	2		{F7}	f7		
{RED}	CTRL	3		{F8}	f8		
{CYN}	CTRL	4		←	←		
{PUR}	CTRL	5		↑	SHIFT	↑	

If you've enjoyed the articles in this book, you'll find the same style and quality in every monthly issue of **COMPUTE!** Magazine. Use this form to order your subscription to **COMPUTE!.**

For Fastest Service,
Call Our **Toll-Free** US Order Line
800-334-0868
In NC call 919-275-9809

COMPUTE!
P.O. Box 5406
Greensboro, NC 27403

My Computer Is:
□ Commodore 64 □ TI-99/4A □ Timex/Sinclair □ VIC-20 □ PET
□ Radio Shack Color Computer □ Apple □ Atari □ Other ________
□ Don't yet have one...

□ $24 One Year US Subscription
□ $45 Two Year US Subscription
□ $65 Three Year US Subscription

Subscription rates outside the US:
□ $30 Canada
□ $42 Europe, Australia, New Zealand/Air Delivery
□ $52 Middle East, North Africa, Central America/Air Mail
□ $72 Elsewhere/Air Mail
□ $30 International Surface Mail (lengthy, unreliable delivery)

Name ____

Address ____

City ______________________ State __________ Zip __________

Country ____

Payment must be in US Funds drawn on a US Bank; International Money Order, or charge card.

□ Payment Enclosed □ VISA
□ MasterCard □ American Express
Acc t. No. ______________________ Expires ____ / ____

Computers for People

Computers are for people. Little people as well as big people. Youngsters like computers too, and they should be encouraged to use them. *Computing Together: A Parents and Teachers Guide to Using Computers with Young Children* will show you how your whole family—adults and children alike—can get the most from your home computer.

What's new in home education software? What kinds of programs are suitable for young children? What games are available that involve the entire family with the computer? Are there word processing programs designed for young children that adults will find just as useful? *Computing Together* answers these questions, and much more.

Author Fred D'Ignazio lives in Roanoke, Virginia with his wife, two children, eighteen computers, and seven robots. He has written fifteen books, including *Katie and the Computer, The Star Wars Questions and Answer Book,* and *Atari in Wonderland.*

ISBN 0-942386-51-5