PIE-MAN

by Eagle Berns and
Michael Kosaka

You got a late start looking for that summer job, and all you could find was a baker apprentice position at the Automated Bakery Company. Simple enough, since the pies are made by machine... all you have to do is add topping and put the pies away when they come out on the conveyor belt. Shouldn't be too difficult of a summer, you think to yourself...

Penguin Software

830 4th Avenue Geneva, IL 60134 (312) 232-1984

Works with Keyboard, Joystick, or Atari Joystick
Softline's New Policy
2

Directline
3

Master of the Mite:
An Interview with Jim Nitchals
Matthew T. Yuen
6

Pac-Man Champ:
Ready on the Set
9

The Amazing Maze in 3-D
Brian Fitzgerald
10

Atari Sound:
Music of the Spheres
Bill Williams
15

The Case of the Micro Mystery:
The Private Eye’s on Home Computers
Andrew Christie
18

Gameline
22

New Players
27

Adventures in Adventuring:
The Thing’s the Thing
Ken Rose
28

Apple II Graphics:
Byting Off Animation
Ken Williams
31

High Scores and Highlines
35

Sherwin Steffin’s education column will return next issue.
Dear Readers:

This issue begins the second year of Softline magazine and marks a turning point in its existence. Softline will no longer be free.

During the past year, we've sought to focus on entertainment software for the home computer market in its broadest aspects. Our cover stories have ranged far afield to bring a diversity of views to the home gaming phenomenon. Our features have introduced you to up-and-coming programmers and companies and provided sneak previews of innovative software. In this light, we can finally report—much to our relief—the release of the Arcade Machine for the Apple II computer from Broderbund Software, featured in these pages several months ago.

We believe entertainment software is a light subject worthy of weighty consideration. To practice what we preach, we've brought you four regular columnists. Ken Williams has focused on graphics techniques for the Apple II, and Bill Williams has addressed the creation of sound for Atari programs. In addition to the two columns speaking to specific computers, Ken Rose has used Basic to present techniques for writing adventure games for any computer.

And, because gaming is especially for the young, Softline has featured an education column by Sherwin Steffin to help youngsters and parents understand the whys and wherefores of educational software. Steffin has taken this issue off to bring a diversity of views to the home gaming phenomenon.

Softalk Publishing produces three magazines, but it is our feeling for Softline seems to have communicated itself to you. Our circulation has doubled and reader response has always been strong. We bask in the warm comments received.

But it's time to put to bed the myth that Softline is an advertiser-sponsored publication. You don't need an advanced course in number theory to see that Softline has not significantly increased the amount of its advertising. What that has meant is stable revenue in the face of increasing costs, caused by inflation and the doubling of our circulation. Try as we might, we cannot print twice as many copies on the same revenue.

So, sadly, we must turn to you to help bear the expense. We take no pleasure in this act, because it's been a real joy to give you the best magazine we could afford each issue. All of us at Softline can testify that it's more fun to give than to receive.

Nevertheless, Softline will now cost $12 for the six issues representing one year. Until such time as we can efficiently mobilize a mailing effort to send you a renewal notice, we will accept $9 for one year and $18 for two years. These lower prices are our way of thanking those of you who save us the effort of sending you a renewal notice.

We're proud of the value that we pack into Softline and we have fun making the magazine. All of us hope you'll choose to share that fun with us for another year.

Sincerely,

Al Tommervik, Publisher
Margot Comstock Tommervik, Editor
Ken Williams, Associate Publisher
Andrew Christie, Co-Editor
Kurt Wahlner, Art Director
Capturing the Elusive Asterisk

All that is required to do text or graphics printouts from copy-protected games is a monitor ROM switch (the red one) on a firmware card, and a printer that can be controlled without the use of special driver routines. First flip the switch, then press reset. You’re back into Basic (a control-C or a control-B) and can then execute the necessary printout commands. Also, isn’t Bob Bishop’s new game called Dung Beetles rather than Tumblebugs?

Jordan Weinstein, Woodside, CA

Message received, Jordan. Nice work (if you can get it). And yes, it was called Dung Beetles. Evidently the name was not considered sufficiently, uh, marketable.

Hi-Res Multicolor

I own an Apple II Plus. It has been my impression that a shape table drawing is always drawn in the one current hi-res color, but I have begun to see some tables that are in two or more colors. How is this done?

Lyle VerPlanck, Costa Mesa, CA

You can create shape tables on the Apple that contain up to three colors, but there are some limitations. First, the color will be lost if the scale is larger than 1 or if rotation is anything but 0 (right side up) or 32 (upside down). Second, certain factors may conspire to change the color displayed.

There are two important principles behind color shapes. The first is that a shape table, when scale equals 1 and rot equals 0, is a set of instructions for the computer to plot individual dots in a distinct pattern. The second is that the Apple II displays colors when every other dot (horizontally) is on, black when all are off, and white when all are on. Thus, within a single shape, one area can be made green by plotting only on odd X coordinates, another area can be blue by plotting only on even X coordinates, and a third area can be white by plotting on all X coordinates.

Now about those conspiring factors. If with scale = 1, rot = 0, and hcolor = 3, you draw the shape at 140,80, it will come out as planned. If you draw it at 141,80, it will have its greens and blues switched. If you set hcolor = 7 (white2), the shape will have oranges and violets where its greens and blues were. In other words, you can have color in shape tables, but you have to be careful where and how you draw them.

Cracking the Egg

I need help. In Zork I, how do you get the egg open and finish the adventure? In Zork II, how do you get behind the menhir, or get through the door with the wizard head?

Derek Bosch, Williamsville, NY

Write to Zork User Group about their book of invisibilities for the first Zork; they’re at Box 20923, Milwaukee, WI 53220.

And about getting through that door, remember: you can catch more flies with honey than you can with vinegar.

Riddle Me This

I would appreciate hearing from any Wizardry fans who have solved the riddle that is asked before the door at level six in Knight of Diamonds.

D. K. Ullman, Santa Ana, CA

Savory Small a la Modem

It was just a few weeks ago that I first saw a copy of Softline, dedicated to one of my sore weaknesses in life: computer games. One of the first articles I happened to turn to in Softline was “Modem Gamesmanship” by Roe R. Adams III, which describes, among other things, the Post Games bulletin board service on the Source. Lo and behold, the first notice listed is from one TCP072. Well, you guessed it, that’s me!

Just to follow up on Mr. Adams’s article, after I posted that notice about Mystery House I received about a dozen replies by Smail (nickname for the electronic mail service available to all Source subscribers) offering all manner of hints, answers, and advice. An interesting sidelight to this is the fact that after the article in Softline, I now receive Smail letters every week. They say, “Gee, I saw you in Softline. I’m in the same situation! Can you give me a hand?” I am more than happy to respond. In fact, I’d love to talk to anyone about Mystery House, or Apple Adventure, or any other game I’ve played. My current passion, by the way, is Wizardry.

Steve Pisk, New Britain, CT

Incontrovertibly Atari

I enjoyed Journey to the Planet Pincus in the May Softline. I thought you might be interested in how I converted this program to Atari Basic.

Line 15: GRAPHICS 0: REM CLEAR THE SCREEN
Line 35: Change A$(12) to D$(1). String arrays are not used in this version, and D$ must be dimensioned.
Lines 50–65: Delete these lines.
Line 80: Read N, S, E, W, D, U
Atari Basic does not read arrays out of data statements directly.
Line 240: On R gosub 2010, 2020, 2030, 2040, 2050, 2060, 2070, 2080, 2090, 2100, 2110, 2120
These are the room description subroutines.
Line 245: ?: ? “Direction()”: input D$?
Lines 250–360: Change left$(D$,1) to D$.
Lines 2010–2120: Change data to ? and place return at the end of each message. Some messages will have to be placed on more than one line.

Stuart Pierce, Picayune, MS

Message from Middle Earth

Regarding the book Ken Rose cited in May Softline, entitled From Here to There and Back Again, I would like to be the one to tell you that it is not Frodo’s book at all, but his Uncle Bilbo’s. If other Tolkien fans see this I am sure that more letters will be forthcoming.

James Brown, Chicago, IL

Authoritative Relations

I refer to a letter written by Geoffrey Puterbaugh of Sunnyvale, California, “Copyrights and Wrongs” [May 1982], more particularly to the last paragraph which mentions Wizardry, published by Sir-tech. The following should be of interest to your readers, including Mr. Puterbaugh.

Wizardry was designed and coauthored with Mr. Woodhead by Mr. Andrew Greenberg. Mr. Greenberg has contributed to fantasy role-playing on Plato and in other previous fantasy role-playing games, Moria being one of the descendents. Mr. Woodhead is the author of a fantasy role-playing game on that system named Sorcery, now more than five years old. The original prototype of Wizardry preceded the appearance of Moria by a number of months.

It is regrettable that Mr. Puterbaugh, while being interested in Mr. Woodhead’s opinions, was not interested enough to ask and saw fit to pose the question in a form that verges on libel.

F. B. Sirotek, president, Sir-tech, Ogdenberg, NY
WE'RE WRITING
At the rate we're going, we'll have these pages filled by 2083. And by 2084, people will be clamoring for the next Infocom creation.

We hate to disappoint our public. So we keep you waiting. Because while the software factories are cranking out arcade game after arcade game, pulpy adventure after trite fantasy, we're writing and rewriting, honing and perfecting. Before a single person enters one of Infocom's worlds, it must be crafted into a living, riveting, definitive experience.

Judging from the public's reaction, it's worth the wait. For instance, Creative Computing welcomed DEADLINE™ as "thoroughly engrossing and realistic," while a Softalk readers' poll recently voted ZORK™ I and ZORK II the most popular adventures of 1981.

And now, for the moment, your wait is over. ZORK III, your final step in the underground trilogy, and STAR CROSS™, an exploration of a new dimension in science fiction, are ready for you.

Look at them up there, the little worlds of Infocom. As our universe expands, companions will come to help fill that vast expanse of white space. Till then, they'll continue to stand alone as the best of all possible worlds.

Infocom's worlds are available for Apple®, Atari®, IBM, TRS-80®, Commodore, NEC, Osborne, CP/M®, and DEC®.

Apple is a registered trademark of Apple Computer Inc. Atari is a registered trademark of Atari Inc. TRS-80 is a registered trademark of Tandy Corporation. CP/M is a registered trademark of Digital Research Inc. DEC is a trademark of Digital Equipment Corporation.
You already know Jim Nitchals, even if you don’t know who he is.

He was that kid in school that teachers looked at and thought, “He could really be something if he’d only apply himself.”

He’s that superstar at the office who turned down a promotion so he could keep doing what he likes doing and what he’s good at.

He’s the basketball player who sets up the play and feeds the ball to the center who slams it.

If Nitchals had been in The Empire Strikes Back, he wouldn’t have been Han Solo; he would have been Yoda—an unimposing character, yet the one closest to the Force. But that’s the movies.

In real life, Nitchals is the force behind Cavalier Computer, in Del Mar, California.

For a guy whose extracurricular activities at Torrey Pines High School were the science fiction and chess clubs and who loathed schoolwork so much that he took a proficiency examination to finish early (“I never received my diploma, but I did get a certificate that said I was proficient”), Jim Nitchals isn’t doing too badly.

Now twenty years old, Nitchals has programmed five games for the Apple, four of which spent a total of nine months in Softalk’s Top Thirty. The fifth, Ring Raiders, was offered free by mail during December 1981 and found more than three thousand takers.

His company, the publisher of such Apple winners as Bug Attack and Microwave, is raking in close to two hundred thousand dollars a year, and hopes to surpass that very soon.

Nitchals and his fellow programmers at Cavalier have standardized T-shirts and shorts as the official programmers’ uniform—appropriate for the southern California company. Business execs who earn what Nitchals makes can’t boast that luxury.

And what of Nitchals’s peers in high school who shunned and ostracized him as a computer junkie? While they’re struggling to get the necessary prerequisites for their college majors and agonizing over final examinations, Nitchals’s biggest task is deciding whether to design a game for keyboard or paddles.

Beginning on Brand X. Strangely enough, a Radio Shack electronics kit was Nitchals’s first experience with technology. At thirteen he began creating circuits for small electronic devices, one of which was a short-wave receiver. Fiddling with these steered his interest to ham radio. It was through catalogs and circulars from the electronics store that Nitchals began hearing about microprocessors.

Working with a friend from his math class, Nitchals started designing computers and programs of his own. By the tenth grade, he’d landed a job doing the same sort of work at a Del Mar research and development firm.

Later that summer, he went to work for Micro Works, a producer of digital television cameras for the Apple. At that time, Micro Works didn’t know a whole lot about the Apple and neither did Nitchals. But this didn’t pose a great problem; his first hires program was an Apple first, too: the first program written for the Apple that scanned a television screen and produced a video image printout. Nitchals claims to be the first person ever to have a self-portrait taken by an Apple.

While other kids were at practice for after-school sports, Nitchals was at practice with the Apple at his local computer store, which was anything but local. “I wasn’t able to drive yet, so I had to take the bus fifteen miles into San Diego,” Nitchals recalls. There he began experimenting with the computer, getting in some hands-on experience and even helping out customers once in a while.

Now, you’d assume that someone so adept at and interested in electronic technology was probably a lot like those brainy kids you knew in high school. They came in two types: those who studied all the time and did nothing but hit the books, and those natural Univacs who never applied themselves and still landed in the high end of the grade curve. Nitchals was neither one.

“I had heard that some people actually were hitting the books, but I found that fifteen minutes a day would do it. The problem was that I didn’t want to spend even that much time,” he explains. “I took two years of physics and hadn’t received even a semester’s worth of credit. The class was ‘go at your own pace,’ and for me that meant ‘go nowhere.’” And yet that’s when Nitchals started to go somewhere.
Teddy the Salvageman, a teddy bear in overalls. His only weapon is a spaceship, and his mission is not to kill things to score points. He is wrote with Zimmerman, the hero is not a warrior, he doesn't fly a spaceshipe, and from that came 'Cavalier.'

Nitchals's first work for Cavalier came as a part-time endeavor. In the summer of 1980, he picked up a consulting job and designed an interface card for the IBM Selectric, a card that is still selling today. Using Moore's Apple, Nitchals was writing Asteroid Field on the side. "We were all fans of the Asteroid's arcade game," Nitchals says, "and we weren't sure if we could handle designing hardware. So the most obvious thing to do was to write the game for the Apple.

"At the time, I had never seen a hi-res game on the Apple, and I felt confident I could produce something better than the games that were selling." Nitchals was right. On November 16, 1980, he took Asteroid Field to a user group meeting and sold fifteen copies on the spot.

When Nitchals's second game, Star Thief, came out, the objective for the company was for all to make some money and then go their separate ways. However, Printz started seeing the pot at the end of the rainbow and convinced Nitchals there was no point in stopping there.

There is no doubt, with bestsellers like Star Thief, Bug Attack, and Nitchals's latest, Microwave, that Cavalier Computer is doing well financially. But that doesn't necessarily make Nitchals a rich man. "I take only as much as I need to keep up a modest lifestyle. That includes paying rent, driving a beat-up '76 Chevette, and supporting my eating-out habits." Nitchals doesn't like to cook and neither does his sister, with whom he shares his home. "That means I eat out quite often."

Born to Program. Nitchals's involvement in the administrative aspects of Cavalier is now almost nonexistent, and that's fine with him. He'd rather program and help teach others how to program, though he admits that he's not really emotionally suited to being a teacher. "Sometimes I have to yell it into them."

The rewards Nitchals receives for being a mentor are more intrinsic than financial. Despite the strain on his patience, it gives him a chance to see others in the same position he was in a few years ago, learning the same things he was learning, only with a little more help. His working with programmers at Cavalier, Nitchals feels, makes their jobs a bit easier.

"I don't know of many other companies where aid is available at all times for programmers. Of course, there are exceptions. Ken Williams and Doug Carlston are great; though their main occupation is business, they can still take time with the programmers. My main occupation is programming, so I try to spend as much time with Jay [Zimmerman] and Mike [Abbott] as I can."

Nitchals sees himself becoming even less involved in the business administration aspect of Cavalier as time goes on. Though he already has a grasp of what makes a business work, he'd rather write a program to add up a column of numbers than add them up himself.

Nitchals claims not to have the personality for business and negotiations, but he does have a very definite personality that makes itself apparent through his games.

While other game writers were developing ways for us to shoot aliens and enemy spaceships, Nitchals's Bug Attack had us shooting at ants, butterflies, and millipedes while being serenaded by an old campfire song, "The ants go marching two by two, hurrah, hurrah. . . ."

In Microwave, the music Nitchals chose to accompany the maze game included the theme from the Star Wars cantina and a chase tune from Keystone Cops flicks. In this game, which Nitchals cowrote with Zimmerman, the hero is not a warrior, he doesn't fly a spaceship, and his mission is not to kill things to score points. He is Teddy the Salvageman, a teddy bear in overalls. His only weapon is used for defense, and his mission is to go around picking up calculators, hammers, and other assorted objects in order to repair his spaceship.

"We were all at Jay's house one night working on shapes and characters for Microwave, and the teddy bear was the cutest looking shape at the time, so we just stuffed him in there, and it worked and looked good," explains Nitchals.

For a programmer who chooses things because they're cute, it comes as little surprise that one of Nitchals's favorite games is Sneakers, for the sound effects when bonus points are awarded, its variety of characters, and its general pleasantness. "With those little smily faces on the sneakers, it has a bright, colorful, cheerful way about it."

Is There Life after Twenty? Nitchals is young even by programmers' standards. His high school classmates are trying to devise ways of becoming millionaires by the time they're thirty. Nitchals wants to make it by twenty-five. "If I can't find it in programming games, I'll find something else that's more profitable," he projects. "But it has to be fun. Right now it's just for fun, and I want to keep it that way."

What does a twenty-year-old game programmer who's the center of a successful company do in his spare time? No, it's not pushing big bucks into high-style entertainment and fast living but, rather, pushing quarters into his favorite video arcade machines: Ms. Pac-Man, Tempest, and Tron, a game for which he held the high score in the San Diego area.

If he's not in the arcades, you might find him engaging in other favorite pastimes: riding the bumper boats at the local miniature golf park or at home listening to the Police, Styx, Elton John, Heart, or the Electric Light Orchestra.

Nitchals is a simple person with not much to distinguish him from the next person his age—except for his considerable accomplishments at Cavalier. A person who is as softspoken as his games are entertaining, he enjoys what he's doing and sees no reason to change.

Score high on the SAT Computer SAT

A complete program for scoring high on the Scholastic Aptitude Test


- Makes studying easy and enjoyable even for those with no computer experience.
- Computer Software
  - Scores and times your performance.
  - Calculates College Board equivalent score.
  - Diagnoses your strengths and weaknesses.
- Prescribes specific drill and review on computer and in the textbook.
- Special Features
  - 1000 Electronic Vocabulary-Building Flash Cards
  - 560 Specially-Designed Computerized Drill Items

Educator's Edition Available

For further information or to order CALL (714) 231-6616 or send me information on the following

- The 48K Apple II or Apple II Plus
- The TRS-80 COMPUTER SAT
- The Educator's Edition Package

HBJ
Harcout Brace Jovanovich, Inc.
Dept. Computer SAT-202, 1500 68th Avenue, San Diego, CA 92101
World Leader in Educational and Testing Materials
The SNAVE Guarantee:
If you find any lower prices (GOOD LUCK!)
we guarantee to try our best
to beat them. Just ask.

Need a 2nd Drive?
Micro Sci A2
- 100% Apple Compatible
- Fast-Reliable Operation
- Easy on the wallet

$349

Memorex Disks
(box of 10)
only $19.95
(min. 2 boxes)

The ProWriter Printer
- 120 Characters per second
- Tractor and Friction Standard
- A Great Printer
- If you don’t have one, it’s
time to get one.

$494
Interface w/cable $119

SNAVE SYSTEMS
Post Office Box 957
Niles, Illinois 60648
(312) 966-4505

We accept VISA and MASTERCARD. (Include card # and expiration date)
Money Order, Certified Check, Personal Check (allow 10 days clearance)
Add Only $2 for UPS delivery, (hardware and foreign extra)
Illinois residents add 6% sales tax (except on software)
Prices subject to change.
Open 7 days a week.
Pac-Man Champ: Ready on the Set

At fifteen, Matthew Laborteaux already has significant accomplishments to be proud of. He plays Albert Ingalls on "Little House on the Prairie," and, with a score of 1,200,000, he's the national Pac-Man champion.

He took the title April 25 in a warehouse in Santa Monica stocked with Pac-Man machines provided by the tournament's sponsor, People magazine. Last October, he placed tenth in the Centipede playoffs at the Atari world championships in Chicago.

Laborteaux did not attain his current level of expertise overnight. He started with the early Pong games and got a video cartridge system when they were first introduced; but he found the games insufficiently challenging. Wandering back to the arcades, he witnessed the birth of Space Invaders and Asteroids, but he wasn't really hooked until Missile Command.

"It's probably my best game. I enjoy it—that type of action—a bit more than Pac-Man," Laborteaux says. But the maze game, in its current and most bizarre form, is still close to his heart: "My favorite right now is Dig Dug. I tried to get into Zaxxon, but it was just too hard."

And that's a problem. "Once you've got the patterns down, a game is easy. Then they come out with harder games and it takes longer than the standard two minutes of playing time even to figure out how to play. You have to keep plugging in quarters just to see what the game is."

Like many arcade fans with a finite supply of quarters but higher standards for graphics and animation than home game machines can deliver, Laborteaux is gravitating toward a personal computer.

He has taken a course in the Apple II, and, due to his ongoing success with Atari's products, he's considering the purchase of an Atari 800.

In the meantime he has his own personal Pac-Man arcade machine—the spoils of victory—and a plaque commemorating his feat. And he has time to get in shape to meet the inevitable challengers who'll want to see just how fast he really is.

Experience the POWER ...
You sit isolated in the sweltering heat inside "The Pill Box". Your sworn duty is to wipe out the armada of enemy tanks relentlessly pushing forward across the desert sand. Huge steel monsters suddenly sense your position. Steel grinds together as the turret slowly turns toward you. It comes down to a life or death, split second confrontation. Are you calm enough, cool enough to handle the power of ... PILL BOX!

The most exciting high res., super graphic arcade game for your apple in a long time. Features full 3-D coordinates. Track the tanks with radar. Full color features. Requires 48K with joystick or paddles.

Save 1/3 with special introductory mail order price of $19.95 plus $2.00 postage and handling.

SATISFACTION GUARANTEED. Visa and Mastercard accepted.

Check or Money Order to: Lord of the Games P.O. Box 6592 Rochester, MN 55901
All orders shipped within 1 day.
It's a bright and sunny day. You're walking along the street to the store when suddenly a giant pit opens up under your feet, and you tumble head over heels two thousand feet down into the ground, splashing into a vast subterranean lake. Coughing and gasping, you fight your way to the surface and swim frantically for the shore of an island that's less than a hundred feet from you. You're more than halfway there when a giant water snake bursts through the surface and grabs you.

Sound familiar? Let's hope not; it's made up. But that's beside the point, said point being overhype, or overdramatization. Much of what goes on in game software is overpublicized, overrated, or overcriticized. Credit where credit is due, yes; but there is a shady area where the ignorance of the public blends in with the natural wish of an author for recognition, and this puts an aura of mystique around things that don't quite deserve it; people then say, "Wow, that's neat, but I could never do that."

Not so. And we'll show you one case, if you'll follow along quietly so as not to disturb Gadanya, jealous keeper of the crypt of algorithms.

You're Moving through Another Dimension. We live in a three-dimensional world (well, perhaps more, but that's clouding the picture). Yet merely by closing one eye, we can erase the usual third dimension, forcing ourselves to infer it from contrast and shading and perspective, a task most of us perform fairly well. The ephemeral nature of our depth perception, joined with our historic desire to represent three dimensions where there are only two—in paintings, architectural drawings, horror movies, and Viewmaster slides—has made us masters of illusion. As we advance in the field of computer graphics, it's not surprising that we should quickly take on the challenge to create three dimensions on the computer.

How? We can take a couple of the tricks that artists have learned during the last few hundred years, apply them, and see what we get. Let's try shading, contrast, and perspective. Now let's pick something to put them in. Something that's popular and nifty. Something that's not too hard, because we're all a bit lazy. Something like a 3-D maze plotter. Yeah, that sounds good.

Well, let's set some limits because, after all, we have a limited medium to work with. The maze (underground dungeon, house, whatever) will be composed of right-angle tunnels all of the same size, and made of a boringly perfect substance. The only remarkable feature for now will be doors, sans doorknobs (well, you know how easily they break off; besides, the original planning estimate didn't include cost overruns, so the Kobolds left the extras out). And we'll do no shading or other fancy things like that that eat up time and memory.

The basic idea is one of perspective; the farther away something is, the smaller it looks. Eventually, we reach a vanishing point, where objects that stretch out far enough converge to a point as we see it; for an example, we'll look at a corridor. Observe the illuminating figure 1. The picture is that of two parallel rows of blocks dwindling off into the distance; note that the point at which they converge is still within the picture frame. It's possible for the vanishing point to be out of the picture or for there to be more than one vanishing point, but, again, let's not get too complicated right away, okay?

Let's get down to juggling a few electrons along silicon-metal pathways. Objects right in front are full size; something very far away is, for all practical purposes, no size at all; and in between those two extremes an object's size is inversely proportional to its distance from the front. Let's do more simplifying, keeping in mind that computers prefer things to go in multiples of two. If the object is halfway to the vanishing point, it'll be half size. Three-fourths of the way there, and it'll be one-quarter its beginning size. Seven-eighths of the way there, it'll be one-eighth normal size. And so on.

This gives us our first bit of concrete math:

\[
(shrinking\ factor) = 1 - \frac{\text{distance to vanishing point}}{\text{distance to the front}}
\]
The Root of the Corridor Cubed. Now we need to draw some more pictures and see if we can derive some plotting equations from them. Let’s make things even easier. Our corridors will be in cube segments, say eight by eight by eight feet (ten foot cubes are so pedestrian, don’t you think?): this gives us an easy angle on the perspective and proportion.

Look at figure 2 and pretend that you see a corridor stretching off into the distance. The vertical lines are there to help you pretend; each one marks off one of those eight by eight by eight foot cubes, and each wall panel is eight by eight feet. Now type this Applesoft quickie into the Apple and run it:

```
10 HGR : HCOLOR = 1
20 X1 = 41 : X2 = 169 : Y1 = 15 : Y2 = 143 : CX = 105 : CY = 79
30 HPLT X1,Y1 TO X2,Y1 TO X2,Y2 TO X1,Y1
40 HPLT X1,Y1 TO X1,Y2 : HPLT X2,Y1 TO X2,Y2
50 X1 = (X1 + CX)/2 : X2 = (X2 + CX)/2 : Y1 = (Y1 + CY)/2 : Y2 = (Y2 + CY)/2
60 GOTO 40
```

Amazing, isn’t it? All right, you’re only slightly impressed, but we’ve taken a small step on the way and introduced even more simplifications. The vanishing point is in the center of the screen; thus, all the lines that vanish to the center are forty-five degree lines, and that simplifies our math tremendously.

Now, what math have we done so far? We have a center, and we have left and right wall panels that converge toward it. Each panel is half the size of the one before it, and each panel stretches halfway to the center. Look at figure 3 as we introduce some more math.

X1,Y1 through X4,Y4 form the outer boundaries of the two panels, left and right, and X5,Y5 through X8,Y8 form the inner boundaries; CX,CY is the center vanishing point. Drawing the left right panels is easy enough.

Left panel:

```
HPLT X1,Y1 TO X5,Y5 TO X8,Y8 TO X4,Y4 TO X1,Y1
```

Right panel:

```
HPLT X2,Y2 TO X6,Y6 TO X7,Y7 TO X3,Y3 TO X2,Y2
```

But what about the next set of panels? And the next, and so on? And where do all the variables come from?

Y Not? X1,Y1 to X4,Y4 are predefined; we have to start somewhere. They form the outer window within which everything is drawn. And (listen!!) X5,Y5 to X8,Y8 are calculated from X1,Y1 - X4,Y4 and CX,CY. How? Easy. If each panel goes half the remaining distance to the center, then, for example, X5 is:

```
X5 = X1 + (CX - X1)/2 = (X1 + CX)/2
```

where (CX - X1) is the distance from X1 to CX. We divide that by two, and add it to X1 to find X5, and, presto chango, we’ve got it.

The others come similarly:

```
Y5 = Y1 + (CY - Y1)/2 = (Y1 + CY)/2
X6 = X2 - (X2 - CX)/2 = (X2 + CX)/2
X6 = Y2 + (CY - Y2)/2 = (Y2 + CY)/2
X7 = X3 - (X3 - CX)/2 = (X3 + CX)/2
X7 = Y3 - (Y3 - CY)/2 = (Y3 + CY)/2
X8 = X4 + (CX - X4)/2 = (X4 + CX)/2
X8 = Y4 - (Y4 - CY)/2 = (Y4 + CY)/2
```

To interject a note from an efficiency-minded world, X1 and X4 are the same, X2 and X3 the same, Y1 and Y2 the same, and Y3 and Y4 are also the same. It’s just easier to use them all when thinking about this. So X5 and X8, X6 and X7, Y5 and Y6, and Y7 and Y8 are also going to be the same. If you’re still here, we actually only need to calculate X5, X6, Y5, and Y6 and get the others from them:

```
X8 = X5 : X7 = X6 : Y6 = Y5 : Y8 = Y7
```

“Now, Thumper, what did your father say?” “Which one? Oh, yeah; waste not, want not.” So, once we’ve drawn the first set of panels, we can recycle the variables for the next set of panels, thusly:

```
X1 = X5 : Y1 = Y5 : X2 = X6 : Y2 = Y6
X3 = X7 : Y3 = Y7 : X4 = X8 : Y4 = Y8
```

and then recalculate a new set of X5,Y5 . . . X8,Y8 and keep on going.

Easy? Easy. Well, what if a panel is missing? (Call the police.) For one, we don’t draw it. But that just leaves a hole, and to complete the picture we should draw what’s behind the missing panel. To make things easy on us (again), we’ll just draw the forward wall of the next cube over, if there is a forward wall. So look at figure 4, and we’ll discuss drawing those left and right forward walls.

Start with the left missing panel and the front wall behind it (a front or forward facing wall is one that is staring us full in the face, not slanted to one side or the other). We draw it like this:

```
HPLT X1,Y5 TO X5,Y5 TO X8,Y8 TO X1,Y8
```

And the one on the right side will be drawn as:

```
HPLT X2,Y6 TO X6,Y6 TO X7,Y7 TO X2,Y2
```

Wasn’t that fun? You in the front, say yes. Now let’s do some interesting things. Corridors sometimes end, don’t they? And doors are usually convenient things, so we should add those. First, let’s have that corridor end. Look back to figure 4. The corridor ended; oh no, what do we do? Square it off, of course. That would be after an iteration of left-right panel drawing, so the end would be:

```
HPLT X5,Y5 TO X6,Y6 TO X7,Y7 TO X2,Y2
```

Got it? Good. If you don’t have it, we’re all in trouble. That’s quite enough of that.

Getting Around. Now for the doors (the Beatles come later). Ah, doors, those perhaps magical portals to other places and times (yeah, five seconds later on the other side of the silly door). Please to refer to figure 5.

We have three kinds of doors to draw: doors on a side panel, doors on a front facing wall, and a door at the end of a corridor. The last shall be first, because it’s the easiest. So how should it go? A door is going to have some margin between the left and right sides of the wall it’s on and some margin from the top of the wall to the top of the door. Let us give some assignments. YV is the margin from the top of the wall to the top of the door, and XV is the margin from the left side of the wall to the left side of the door and also the margin from the right side of the wall to the right side of the door. Observe figure 5a for enlightenment.
The equation for that door is:

\[
\text{HPLT } X5 + XV, Y8 \text{ TO } X5 + XV, Y5 + YV \text{ TO } X6 - XV, Y5 + YV\]

And so on we trek to the next door, one on a front facing wall, and you should cast an inquiring eye on figure 5b. Since we only see half of the wall, we're only going to see half of the door, and the two plots look like so:

Left door:

\[
\text{HPLT } X8 - XV, Y8 \text{ TO } X5 - XV, Y5 + YV \text{ TO } X1, Y5 + YV
\]

Right door:

\[
\text{HPLT } X7 + XV, Y7 \text{ TO } X7 + XV, Y6 + YV \text{ TO } X2, Y6 + YV
\]

Not too terribly complicated, is it? You'd better say no, because this next part is a little complicated—but fun.

Look at figure 5c. Don't wince; look at it. Scary, isn't it? It needn't be; and we'll endeavor to make this at least painless.

First, let's pull a fact out of the air. In a right triangle with an acute angle of forty-five degrees, the two sides next to the right angle are of equal length, so if you know the length of one side, you then possess the terrible secret of the length of the other side. Now, notice that in figure 5c four right triangles are pointed out; since all the lines form angles of forty-five degrees, we have most of the knowledge needed to draw the door encoded in the picture. Let's dig it out.

The lower left-hand coordinates of the door are \(X4 + XV, Y4 - XV\). That's simple enough. The upper left-hand coordinates are \(X1 - XV, Y1 + YV + XV\). Check to make sure you understand that. Now, the upper right-hand side of the door is \(X5 - XV, Y5 + YV - XV\), because we add the YV but have to back up by the amount XV up the Y axis. And, finally, the lower right-hand side is \(X8 - XV, Y8 + XV\). String it all together and we get:

Left door (side panel):

\[
\text{HPLT } X4 + XV, Y4 - XV \text{ TO } X1 + XV, Y1 + YV + XV \text{ TO } X5 - XV, Y5 + YV - XV \text{ TO } X8 - XV, Y8 + XV\]

The right door is similarly (tortuously):

Right door (side panel):

\[
\text{HPLT } X3 - XV, Y3 - XV \text{ TO } X2 - XV, Y2 + YV + XV \text{ TO } X6 + XV, Y6 + YV \text{ TO } X7 + XV, Y7 + XV\]

And there we have it, all the math we get for now.

Get Back in Your Cell. Therefore, on to the program! But, before that, we have to mention something nasty, so close your ears, data structures.

A maze plotter is actually pretty useless without a maze to plot (but it looks so elegant and pretty!), so we need a way of representing a maze in the machine. Again, let's be easy and make the maze in those chunks. Then we can turn the maze into a matrix, and each cell of the matrix will have a number describing how that eight by eight foot cube looks; where the walls are, and where the doors are. So we have a matrix, \(A(100, 100)\), say. It's pretty large, and it's full of integer numbers. Each number has five parts: one for each of the four directions, north, east, south, and west, and a special number, which we'll put off until later. Now, delving a bit into machine language origins, an integer number is sixteen bits long. So we give each of the directions three bits, and the special number gets four bits. If this all seems meaningless, be patient, because now we'll come back to reality.

This means that a cell edge can be one of eight things. Right now, there are three. They are:

0 wall
1 empty space
2 door

To put a cell number together, do this:

\[
\text{cell } \# = (\text{special}) \times 4096 + (\text{north type}) \times 512 + (\text{east type}) \times 64 + (\text{south type}) \times 8 + (\text{west type})
\]

For example, the cell number for a cell with a wall to the north, doors to the east and west, and space to the south would be:

\[
\text{cell } \# = (0 \times 4096) + (0 \times 512) + (2 \times 64) + (1 \times 8) = 138.
\]

The program, of course, has to be able to reverse this, and there's a subroutine at 1600 that does this; XX, YY is the address of the cell you wish to explode into full form. A discussion of that is for later, also.

So, here at last is the program, and nothing can stop it! Cast an eye on the listing, and read along.

Lines 10 through 80 set up few global variables needed and set up the maze in the matrix. D is the direction you face in the maze: 1 = north, 2 = east, 3 = south, 4 = west. XC, YC are the coordinates of the cell you're standing in. RR is the depth the corridor (or room or whatever) is drawn to; it's impractical actually to try to draw to infinity.

Lines 90 through 130 print the direction you're facing, and the cell you're in. Then line 140 sets up the depth counter and calls the actual maze plotter subroutine.

Lines 190 through 260 ask which direction you'd like to move, and then branch to a likely area to move or turn you, as the case may be. The keys used are: - (turn left), > (turn right), +; (move forward one step), and ?/ (move backward one step).

Lines 270 through 400 are the bits of code that change the direction and/or XY coordinates; simple enough. Lines 410 through 430 are a teleport (why not?), push T.
Lines 1000 through 1840 are the maze plotter and ancillary subroutines. But hold it; there's one last thing, and, really, only one, that we need to discuss. The burning topic that couldn't be held back any longer is: peripheral vision. "Oh," you say, "but how does that...?" Stop! Don't commit yourself. Just read this.

We can plot ahead of us, and to either side of that, but what of the cell we're standing in? Do we cop out and say, "Well, I'm standing on the very frontmost edge of the cell I'm in, and consequently can't see anything of it?" No. No. NO! Let's be a bit more classy, and draw just a bit of the cell we're in, tacking that on to the outside of the window that everything else is in. Look at figures 6a and 6b, and decide which you like better. But pick 6b anyway, okay? Good.

This creates a bit of a problem, because the first bit is a special case and needs special treatment in spots. So we'll explain all that as it comes up.

1000 clear the hires screen and set color to green
1010-1020 set up special case coordinates for peripheral vision cell
1030-1040 set up normal outer window coordinates
1050 define the center coordinates
1060 set up special case XV and YV
1070 get starting cell coordinates
1080 skip recalculating of X5,Y5 X8,Y8 the first time around
1090-1130 calculate X5,Y5 ... X8,Y8 and XV,YV
1140-1150 get left wall number; if space, goto left wall extend
1160 if not door, then skip door plot
1170 if not first cell (peripheral) then skip special case
1180 plot peripheral left panel door and skip to 1200
1190 plot left panel door
1200-1220 plot left panel wall and goto right side plot
1230 get cell to the left of current cell; if no front-facing wall, skip to right side plot
1240 if no door, then skip door plot
1241 if not peripheral cell, then skip special case
1242-1243 plot peripheral left-facing door and skip to 1260
1250 plot left front-facing door
1260 plot left front-facing wall
1270-1390 repeat the same thing, but for the right side instead
1400-1410 reset X1,Y1 ... X4,Y4 to start of next panels
1420-1430 if the cell just plotted has a forward wall, then goto 1500 to finish off the plot
1440-1470 otherwise, get the address of the next cell in the direction D
1480 if depth counter has reached zero, then end
1490 otherwise, go back to 1090, recalculate X5,Y5 ... X8,Y8 and keep going
1500 square off the end of the corridor
1510 if no door then skip door plot
1520 get appropriate XV and YV
1530 make the door
1540 return from maze plotter
1600-1700 expand the cell at XX,YY to the four directions and special number
1800-1840 get address of cell in direction P from the cell at X,Y
3000-3190 set up the maze in matrix A%(100,100)

Wasn't that just ever so? So type it in, or get some other hapless

YOUR MOVE

Kraft puts you in command with high performance Joysticks and Paddles, available for Apple II, TRS-80 Color and IBM personal computers.

• Quick change from Spring Centering to Free Floating Joystick from OUTSIDE THE CASE.
• Joystick trim control on X-Y axis.
• Exclusive ONE YEAR LIMITED WARRANTY for Joysticks and Paddles.
Ask your computer retailer for Kraft Precision Joysticks and Paddles. It's Your Move!

Kraft Systems Company, Computer Products
P.O. Box 1268, Vista, CA 92083 • (714) 726-7146

*Apple II, TRS-80 and IBM are trademarks of Apple Computers, Inc., Tandy Corp., and International Business Machines Corp., respectively.

SEPTEMBER 1982 13
200 VTAB 22: HTAB 7: GET A$
210 IF A$ = CHR$ (8) THEN 270
220 IF A$ = CHR$ (21) THEN 290
230 IF A$ = "." THEN 310
240 IF A$ = "T" THEN 410
250 IF A$ = "/'" THEN 330
260 GOTO 200
270 D = D - 1: IF D = 0 THEN D = 4
280 GOTO 90
290 D = D + 1: IF D = 5 THEN D = 1
300 GOTO 90
310 XX = XC:YY = YC: GOSUB 1600: P = D: IF T(D) <> 0
320 PRINT CHR$ (7): GOTO 200
330 XX = XC:YY = YC: GOSUB 1600: P = D - 2: PRINT P:
340 IF P < 1 THEN P = P + 4
350 SP = 0
360 IF P = 1 THEN YY = YY - 1
370 IF P = 2 THEN X = X + 1
380 IF P = 3 THEN YY = YY + 1
390 IF P = 4 THEN X = X - 1
400 GOTO 90
410 HOME: VTAB 22: HTAB 1
420 INPUT "TELEPORT TO (X,Y) ";XC,YC
430 GOTO 90
1000 HGR: HCOLOR= 1
1010 X1 = 27:Y1 = 1:X2 = 183:Y2 = 1
1020 X3 = 183:Y3 = 157:X4 = 27:Y4 = 157
1030 X5 = 41:Y5 = 15:X6 = 169:Y6 = 15
1040 X7 = 169:Y7 = 143:X8 = 41:Y8 = 143
1050 CX = 105:CY = 79
1060 XV = 2:YV = 18
1070 X = XC:Y = YC
1080 GOTO 1140
1090 X5 = (X1 + CX) / 2:X8 = X5
1100 X6 = (X2 + CX) / 2:X7 = X6
1110 Y5 = (Y1 + CY) / 2:Y6 = Y5
1120 Y7 = (Y3 + CY) / 2:Y8 = Y7
1130 XV = (X5 - X1) / 4:YV = (Y4 - Y1) / 8
1140 P = D - 1: IF P = 0 THEN P = 4
1150 X = X:YV = Y: GOSUB 1600: IF T(P) = 1 THEN 1230
1160 IF T(P) = 0 THEN 1200
1170 IF R <> RR THEN 1190
1180 X5 = (X1 + CX) / 2:X8 = X5
1190 X6 = (X2 + CX) / 2:X7 = X6
1200 X7 = 169:Y5 = 143:X8 = 41:Y8 = 143
1210 X1,Y = X: Y: GOSUB 1600: IF T(P) = 1 THEN 1270
1220 GOTO 1270
1230 GOSUB 1600: XX = X:YY = Y: GOSUB 1600: IF T(D) = 1
1240 IF T(D) = 0 THEN 1260
1250 HPLT 29,143 TO 27,31: GOTO 1290
1260 HPLT 29,143 TO 27,31: GOTO 1290
1270 HPLT 29,143 TO 27,31: GOTO 1290
1280 Y = Y: XX = X:YY = Y: GOSUB 1600: IF T(D) = 1
1290 IF T(P) = 0 THEN 1330
1300 IF R <> RR THEN 1320
1310 HPLT 29,143 TO 27,31: GOTO 1290
1320 HPLT 29,143 TO 27,31: GOTO 1290
1330 HPLT 29,143 TO 27,31: GOTO 1290

Survive the perils of the China Seas

An exciting new strategy-adventure game based on the China trade of the 1800's!
Sail between the seven ports of the China Seas and trade general cargo, arms, silk, and opium. Your goal is to make a million in cash. But first, you must...

- battle ferocious pirates
- survive the monstrous storms known as tai-funs
- face merciless thieving street toughs
- evade government officials who seize illegal cargo and impose heavy fines
- bargain with the notorious triad moneylender, Elder Brother Wu

Experience the excitement of the China trade. See TAIPAN at your local computer games store.

Avalanche Productions, Inc.
2460 Embarcadero Way
Palo Alto, California 94303
(415) 856-4881
Building Overwhelming Software
Music plays an important role in our lives. We encounter it at work and at play, traveling down the road—even the telephone hold button plays us music. It should come as no surprise, then, that when presented with a computer equipped with sound generators, the first question most people ask is, “Can it play music?”

**Notes on the Atari.** Creating musical notes with the Atari is pretty easy. A table of pitch values for the notes is provided in the Atari Basic manual and in the Technical User’s Notes. To play middle C, we look it up in the table and find the value 121. Placing this in the audio frequency register of a given channel will produce a middle C, provided we’re using the “pure tone” distortion parameter, 10. (Why this won’t work with other tones is beyond the scope of this discussion; we’ll deal with it at a later date.) If you’re in Basic, the command:

```
SOUND 0,121,10,8
```

will do the job.

Playing chords is a simple matter of looking up each note in the chord and plugging the corresponding value into a different sound channel. With four channels, you can form many complex chords (Cm9, F#maj7, and so on) or stick to a melody line with simpler voicings. To give the music rhythmic structure, preprogrammed music can be stored in a data table with duration values; the program, in turn, uses the duration value as a counter in a delay loop before playing the next note.

**Random Composition.** Producing random music is a little harder. There is a wealth of pitch values that don’t fit into the equally tempered scale; in fact, if we just load random data into the frequency registers, the chances are roughly seven out of eight that we’ll get one of these in-between pitches. The resulting noise will drive any person mad in the space of a few minutes. The problem is compounded by the absence of any obvious correlation between the musical intervals and their numerical equivalents. A half step at C# is a half step, C to D is a whole step): 1,1,1/2,1,1,1,1/2

C# is a half step, C to D is a whole step): 1,1/2,1,1,1/2

Note that the pattern is not symmetrical, which will give the random composer fits in much the same way the nonpredictable pitch values did. The situation becomes really muddled if we want the composer to do something sexy like switch keys or correctly form and resolve common chords within the scale. Some sort of software device is needed to keep all these numbers organized.

**Indirection.** Atari owners should be familiar with the concept of indirection; the color registers are a method of indirectly specifying color. Rather than plotting the number of a color, the Atari plots the name of a color register that specifies the color. For example, putting a two into the screen display area does not mean display color 2, it means display whatever color is in color register 2. If color register 2 contains fifteen, then color 15 will be displayed. This is an extremely powerful graphics tool, for large amounts of previously plotted points can change color instantly with the movement of a single byte.

Indirection is an important concept because it lets us operate on data in terms of categories. We manipulate classes of data grouped together by their similarities, and in so doing we ignore the trivial individual characteristics found inside a class. To return to the color register example, we operate on data in terms of their similarity—they are all specified by register 2—and ignore their individual characteristics, the X and Y coordinates of each point.

**What does this have to do with random music composition?** Indirection, by letting us operate on similarity, allows us to concentrate on function. The notes in a scale, the chords formed in the series, all have specific functions depending on their position in the key. If the random composer is to resolve tones correctly, it must recognize and manipulate notes according to their functions in the scale.

For example, a crucial tone in the scale is the leading tone, the last note in the scale. The leading tone has a strong compulsion to lead into the note above it, the tonic (the first note in the scale). When we follow the leading tone with the tonic, we say we have resolved the tone: we have provided it with a satisfactory conclusion.

In the key of C, the leading tone is B, which means it will have a strong compulsion to resolve to C. In the key of E, however, B is not the leading tone: it is the fifth tone in the E major scale, not the seventh, and has no compulsion to move at all. We have, then, two entirely different functions for the same note, depending on its relationship to the key.

If a random composer is going to produce music with any intelligence, it must recognize the note not according to pitch, which is actually a trivial characteristic, but according to its function in the key. This means we have to deal indirectly with the note; specify it according to its relationship, and let some other software mechanism take care of the trivial pitch specifications.

**Pitching Tonic.** In our sample program, notes are specified by their relationship to the current key. This number will run from zero to six, with seven being the tonic an octave higher. The number is then converted into an offset by reading data from the table at 1100–1107. These data statements convert the sequential note specifications into the correct interval pattern for a major scale.

Before playing a note, the program adds the offset thus obtained to tonic, which specifies the key. This number is the real number of the note to be played, and is converted to a pitch value.
All this footwork enables us to specify notes in any key with the numbers zero through six. In our leading tone example, we would test for tone 6; if true, the note would be resolved to a 7, or alternatively, 0 (the same note an octave lower). This decision is made irrespective of the key being played.

The same idea is applied to the chordal accompaniment. Chords are specified by their roots' relationship to the key, and the following chord is dictated by the rules of resolution set up in the program. For instance, whenever a IV chord is played, there is a 50 percent chance that the key will be changed (lines 1490 through 1510). The new key selected will always be a fifth below the chord (line 1510; we add ten chromatic steps).

**The Listing.** Initialization is located at the end of the program, at lines 1400 through 2020. This was done to speed up execution of the main program loop at lines 100 through 890.

The pitch translation table is read into array `pitch` at line 1420. This line is skipped when initialization is reentered.

Lines 1450 through 1540 write a chord pattern for the piece based on the function of the chord in the previous measure. Add your own rules for resolving chords here.

Lines 1550 through 1590 create an eight-note pattern that is used to give structure to the melody. Lines 1600 through 2020 set up the graphics routine. This routine draws a line from a moving point to a stationary centerpoint in the upper left quarter of the screen, and then mirrors the pattern into the remaining quarters of the screen using the popular kaleidoscope algorithm. No claims to originality are made here, but the pattern produced is pleasing and is a good demonstration of the Atari's ability to maintain graphics and sound at the same time.

Lines 110 through 160 play the chord pattern. Note that the first three entries of the data table at 1100 through 1107 are borrowed to get the offsets from the root necessary to form the chord (the root, a third above, and a fifth above). This double use of the table occurs because the numbers are coincidentally the same.

Lines 170 through 380 play the melody. This melody, unlike the chord pattern, is not figured out ahead of time. Rules for tone resolution are in lines 190 through 340. If the rules don't apply, a random note is selected from the scale, with a 50 percent probability of actually being a note in the chord (line 300).

Lines 400 through 480 take care of moving the graphics points and drawing lines. Because this routine's execution time is a function of the length of the lines being drawn, it also is an interesting tempo control for the music: when the display gets busier, the music plays faster.

When thirty-two bars have been played, the screen is cleared, a new chord pattern is written, and the program repeats. While the chord pattern is being written, line 1450 plays the scale of the last key played to keep the transition smooth.

This program is really a jumping-off point for playing with music theory. The rules defined are the bare minimum necessary to make sonorous music. The more rules you add, the more intelligently the program will compose, and the less random the results will seem. More importantly, creative effort in the definition of a music system will pay off in an increased understanding of music, texture, and style.

In the next article, we'll take a look at the internal details of Atari's sound generators and find out why the pitch tables only work with distortion parameter 10.
FOR BAR=0 TO 31
REM UPDATE CHORD
FOR VOICE=0 TO 2
RESTORE 1100+VOICE:READ OFFST
OFFST= OFFST+PATTN(BAR)
GOSUB FETCH:SOUND
VOICE,PITCH(TONIC(BAR)+FOUND),10,4
NEXT VOICE
REM NOW FOR THE MELODY
FOR BEAT=0 TO 3
IF MELODY< >11 THEN 220
IF RND(0)<0.7 THEN MELODY=12:GOTO 370
MELODY=0:GOTO 370
REM PLAY MOTIF?
IF THEMEF=0 OR RND(0)<0.1 THEN 270
THEMEP=THEMEP+1:IF THEMEP > 7 THEN THEMEP=0
OFFSET=PATTN(BAR)+MOTIF(THEMEP)
GOSUB FETCH:MELODY=FOUND:GOTO 370
REM START MOTIF?
IF RND(0)< 0.15 THEN THEMES=1
IF MELODY=5 THEN MELODY=4:GOTO 370
IF RND(0)< 0.5 THEN 350
REM PICK NOTE IN CHORD
RESTORE 1100+INT(RND(0)*3)
READ OFFST:OFFST=OFFST+PATTN(BAR)
GOSUB FETCH:MELODY=FOUND:GOTO 370
REM ANY RANDOM NOTE
RESTORE 1100+INT(RND(0)*6):READ MELODY
REM **PLAY MELODY**
SOUND 3,PITCH(TONIC(BAR)+MELODY+12),10,6
REM **GRAPHICS ROUTINE**
X=X+DX:IF X<0 THEN X=0:DX=-DX
Y=Y+DY:IF Y<0 THEN Y=0
PLOT CX,CY:DRAWTO X,Y
PLOT 319-CX,CY:DRAWTO 319-X,Y
PLOT CX,191-CY:DRAWTO X,191-Y
PLOT 319-CX,191-CY:DRAWTO 319-X,191-Y
IF RND(0)<0.1 THEN
CX= INT(RND(0)*160):CY= INT(RND(0)*96)
NEXT BEAT:NEXT BAR
GOSUB 1440:GOTO 100
IS "COLDSTART"
TONE(31)=0:PATTN(0)=0:FOR L=0 TO 36:READ
A:PITCH(L)=A:NEXT L
IS "WARMSTART"
TONE(0)=TONE(31)
FOR BAR=1 TO 31:RESTORE 1100+INT(BAR/4):READ
A:SOUND 3,PITCH(A+TONE(0)),10,4
TONE(BAR)=TONE(BAR-1)
IF PATTN(BAR-1)=6 THEN PATTN(BAR)=0:GOTO
1520
IF PATTN(BAR-1)=1 AND RND(0)<0.8 THEN
PATTN(BAR)=4:GOTO 1520
IF PATTN(BAR-1)< >3 OR RND(0)<0.5 THEN
PATTN(BAR)=INT(RND(0)*7):GOTO 1520
REM CHANGE KEY
PATTN(BAR)=0:TONE(BAR)=TONE(BAR)+10:IF
TONE(BAR)>11 THEN TONE(BAR)=TONE(BAR)-12
SOUND 3,PITCH(TONE(0)+12),10,4
IF PATTN(BAR)>6 THEN
PATTN(BAR)=PATTN(BAR)-7
NEXT BAR
REM SET UP MOTIF
FOR L=0 TO 7
IF RND(0)<0.7 THEN RESTORE
1100+INT(RND(0)*3):READ
OFFST=MOTIF(L)=INT(RND(0)*7)
NEXT L
REM **SET UP GRAPHICS**
GRAPHICS 24:COLOR 1
IF RND(0)<0.5 THEN 1640
SETCOLOR 2,RND(0)*16,0:SETCOLOR 1,0,RND(0)*8+4:GOTO 1650
SETCOLOR 2,RND(0)*16,RND(0)*6+4:SETCOLOR 1,0,0
CX= INT(RND(0)*160): X= INT(RND(0)*160)
CY=INT(RND(0)*%):Y=INT(RND(0)*96)
GOSUB 2000:DX=VEC
GOSUB 2000:DY=VEC
RETURN
VEC= INT(RND(0)*4)+1
IF RND(0)<0.5 THEN VEC= -VEC
RETURN
WIZMAKER
WIZARDRY * CHARACTER EDITOR
By RON RICHARDS
- Resurrect the dead.
- Recover lost warriors.
- Modify attributes, race & age.
- Change experience and hit points.
- Learn more spells & earn gold.
- CHEAT! Make a 10,000 level Super Hero — also useable in the
  Knight of Diamonds scenario.
  • Supports line printer.
  • Includes maps of dungeon.
  • Requires 48K & Applesoft.
$20.00
California residents add 6½ % Sales tax
ARS PUBLICATIONS
3710 Pacific Ave. #16
Venice, Calif. 90291
by ANDREW CHRISTIE

I had read a good many detective novels, as I found they were excellent to take one's mind off one's worries. After discussing one with my sister, she said it was almost impossible to find a good detective story, where you didn't know who committed the crime. I said I thought I could write one.

—Agatha Christie

It's three a.m. in the city; decent folks are home in bed. The wind is still blowing in from the desert, hot and dry and steady, fraying nerves to a razor's edge; fingers twitch and eyes glance furtively at shadows or jerk toward the sound of a match being struck and catch the sight of a cigarette end glowing in a doorway.

Two stories up, there's a light in a window. A mainframe programmer is hunched over his terminal, long after hours, stealthily punching up hit points, strength, wisdom, charisma, and agility, and cackling over his readouts. It's all luck; random chance. He doesn't know it. It doesn't mean a thing. Poor sap.

Meanwhile, across town, wandering kids are checking out the Pong game at the Burger Chef, or sneaking into the college recreation center to play Asteroids.

That's how it was. It wasn't much. Origins are always crude. Between games that only a programmer could love and games that had only a slight edge over the spin cycle at Laundry King, there was little to indicate future greatness. But in prehistoric Britain, there was little to indicate Shakespeare.

The health of an art form depends upon diversity. Diversity comes with complexity. Complexity comes with culture. If you're gonna play games with a computer, sooner or later you gotta go to the culture. Otherwise, you wind up as cold meat on a slab, because nobody can play the game forever.

It's that kind of business.

It Was a Dark and Stormy Night. Here's the picture. Sometime around 700 A.D., a teller of tales decided to recount the saga of a mighty hero with a fast broadsword who performed great deeds and saved a village by slaying a monster and its mother.

Not thirteen hundred years after Beowulf, someone got the idea for a new kind of open-ended, role-playing game involving dungeons and dragons. And in a much shorter span of time, someone else put it into a computer.

Thus did the adventure make its way through the cultural matrix. The current focus of our concern did not take half so long. Sometime around April 1841, Edgar Allan Poe, casting about for some leisurely occupation "to keep from going mad," something to do in between the penning of his "serious" stories, hit upon a new idea; a genre composed of "tales of ratiocination." (He had already invented the short story sometime previously, making his task that much easier.) The first, "Murders in the Rue Morgue," presented a brilliant detective, a slightly dim but willing detective's assistant, a very dim officer of the law, a lot of false clues, and a climactic revelation of the solution to the mystery, provided by the brilliant detective in a room with all the suspects present, at the end of the story.

Everything old was once young.

Poe's new invention didn't really catch on stateside, but enjoyed a modest vogue across the Atlantic at the hands of Wilkie Collins, Emile Gaboriau, and Charles Dickens.

After the momentous arrival of Sherlock Holmes in 1887, the entire genre tended to coast on the reputation of the best-known character in English literature until the early 1920s, when Dame Agatha had her fateful conversation with her sister, and, along with Dorothy Sayers, Margery Allingham, and Dashiell Hammett, created what is commonly known among aficionados as the golden age of detective fiction.

At which point, somebody decided it would make a good game.
Adventures into computer mystery, however, were first forays into Deadline, hack and try again. The mystery is indeed different from the adventure. "When you confront a problem, you have to say 'What am I gonna do now,' but not two hundred moves from now, when you come back to it later. In Deadline, we wanted to appeal to the nonfantasy people who would rather be part of a real story; people who always wanted to participate when they read the books.

"We designed the game to be open-ended and to have a large vocabulary, but at the same time, we didn't want it too large and too open."

A detective story on a computer is a different proposition from a detective story in a book. Blank and company had to map out their plot twists and build their characters around them. "We couldn't throw in some amazing coincidence at the end that solves everything and that you could never possibly have guessed. You can't run all over town looking for clues, and you can't talk to the suspects about any subject that comes into your head."

Some other Door Slammed. If Deadline, with its twenty-five thousand words of text and machine-independent language, is the Cadillac of the fledgling genre of computer detective fiction, there are currently several attractive economy models to choose from. Inevitably, they must all be compared to Deadline, as most everything else in this new genre is likely to be for some time to come.

Alibi, from Hayden Book Company's Microcomputer Gameware division, offers several levels of play, unlike Deadline, but it's far more limited and highly structured. Alibi is essentially author Ian Trackman's variation on the stalwart old one-of-them-always-lies-and-one-always-tells-the-truth party game. The first discrepancy in testimony generally gives you your murderer/liar, knowledge you then use to ascertain place and time. Getting all three correct promotes you to the next level—more suspects, more rooms, and a longer time span. In play, it resembles Electronic Detective, much simplified. The near-infinite different sets of circumstances produced by combinations of these variables produces a new "case" with every playing, very much in the tradition of Clue.

For the younger set, there's the nonviolent Snooper Troops, a mystery series from Spinnaker Software, the first of which, The Granite Point Ghost, does, in fact, go all over town. Ostensibly (and successfully) a piece of educational software, this one is a delightful surprise. You get to drive around in your Snoopmobile (a '57 Buick, judging by maneuverability and response, but what the hey), choosing your own suspects from a large cast of characters with separate identities and personalities. You have several different ways to obtain clues, and though your interrogatives are limited, they are never repetitive. The first installment of the Snooper Troops series proves itself an original, owing nothing to any other game. For its literary antecedents, its roots are in the classics of juvenile detective fiction, recalling Encyclopedia Brown and the Hardy Boys rather than Hercule Poirot or Sam Spade.

The Maid Screamed; a Woman Fainted. All of the foregoing will find their markets among the fans of the type of game each represents. But for hardcore mystery addicts, the actual competition for Deadline may well be Sherlock Holmes Consulting Detective, released as a bookshelf game last May by Sleuth Publications of San Francisco and now programmed for the Atari computer by
Voyager Software. It's a game that gives new vistas of meaning to the term open-ended.

Scheduled for a release this fall, the Atari Forth version is a scaled-down rendering of Gary Grady's awesomely complex and intricate original game, consisting essentially of scenarios similar to the Thames Murders from the game case book, and using some graphics for a map of Holmes's London, circa 1886, with numbered clue points. Grady, a scholar of the Holmes canon, took two years to write the game, and embarked on the project with the idea of creating something "more complex than the board games. A game like 221 B Baker Street doesn't give a player a choice; you have no control over the clue you're going to get and there's no relationship of the clues to the process of play. We wanted the idea of solving a mystery rather than a puzzle."

For Grady, the difference between a logic game and a mystery is that a mystery gives you unexplained events from which you must deduce a larger context; a logic puzzle tells you what happened and leaves you to infer the specific events that make up that context.

"In real life, if a policeman walks into a room and finds a dead body, that's all he knows."

Like the original, Voyager's computer translation is planned to be a multiplayer game, with players donning the roles of the Baker Street Irregulars and competing with Holmes in the streets of Victorian London to find the solution to each case with maximum dispatch. In the course of investigating the main case, players are liable to stumble onto several other unrelated mysteries; part of the mystery is finding out how many cases you have to solve.

Consulting Detective will be regularly updated with new cases, all requiring a master disk containing the essential information for play.

A Cloud Passed over the Moon. How to follow Sherlock Holmes was a problem before and may prove to be a problem again. (Info-com's next mystery game, planned for release in early 1983, will be an eyewitness murder case in which you may prevent the murder from occurring or possibly die trying.) It has long been noted that the most vital and interesting writers of fiction in America during the last fifty years have been mystery writers; detective novel hacks; graduates of the pulp thriller, once removed: the Raymond Chandlers, Dashiell Hammetts, and Ross MacDonalds.

This new breed of computer game may well bring out the best in designers and programmers, igniting a comparable renaissance of the popular imagination.

Marc Blank is convinced:

"In the movie Deathtrap, they show you the body and tell you it's dead. It turns out not to be.... In these games, you examine the body. Unlike books and movies, there's no cheating. The intent is not to maximize audience surprise, but a player's sense of accomplishment."

Or as Gary Grady recalls, "Dorothy Sayers once said that the job of the mystery writer is to gently take the reader by the hand and lead him in the wrong direction. We don't do that."

That's the essential difference between computer detection and the literary mystery, though it all started in the same cradle of popular culture. One could take that case as far back as 1794 and Ann Radcliffe's Mysteries of Udolpho, which, in recounting the trials of a stalwart English couple who endure an extended stay in an ancient Italian castle, bearing up under threats of death, abduction, the supernatural, and a general air of menace, probably created several genres at once. Lots of secret panels and underground catacombs, too.

Thus do the old and enduring genetic materials return and return again, crossing over from one medium to another to lend several hundred years of literary tradition to your eager little micro. And everything old is new again.
## CALSOFT

**Personal — Entertainment — Business SOFTWARE**

Low Discount Prices / 15 to 25% Discount off List Price  
Fast Convenient Service / We ship same or next day  
Large Selection of Software / Call or Write for our FREE Catalog!

**Call Toll Free:** (800) 423-5290  **In California:** (213) 991-9641

### AVANT-GARDE

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hi-Rise Secrets</td>
<td>29.95</td>
<td>25.45</td>
<td>29.95</td>
</tr>
<tr>
<td>Hi-Rise Computer Golf</td>
<td>29.95</td>
<td>25.45</td>
<td></td>
</tr>
<tr>
<td>Zero Gravity Pippal</td>
<td>29.95</td>
<td>25.45</td>
<td></td>
</tr>
</tbody>
</table>

### BRODERBUND

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Panic</td>
<td>29.95</td>
</tr>
<tr>
<td>Genetic Drift</td>
<td>29.95</td>
</tr>
<tr>
<td>David's Midnight Magic</td>
<td>34.95</td>
</tr>
<tr>
<td>The Arcade Machine</td>
<td>44.95</td>
</tr>
<tr>
<td>Track Attack</td>
<td>29.95</td>
</tr>
<tr>
<td>Star Blazer</td>
<td>31.95</td>
</tr>
<tr>
<td>Labyrinth</td>
<td>29.95</td>
</tr>
<tr>
<td>Chippilin</td>
<td>NEW 34.95</td>
</tr>
<tr>
<td>County Fair</td>
<td>NEW 34.95</td>
</tr>
<tr>
<td>Deadly Secrets</td>
<td>NEW 34.95</td>
</tr>
<tr>
<td>Serpentine</td>
<td>NEW 34.95</td>
</tr>
</tbody>
</table>

### CAVALIER

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Asteroid Field</td>
<td>24.95</td>
</tr>
<tr>
<td>Star Thief</td>
<td>29.95</td>
</tr>
<tr>
<td>Bug Attack</td>
<td>29.95</td>
</tr>
<tr>
<td>Microware</td>
<td>34.95</td>
</tr>
</tbody>
</table>

### CONTINENTAL

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Accountant</td>
<td>74.95</td>
</tr>
<tr>
<td>Home Money Minder</td>
<td>34.95</td>
</tr>
<tr>
<td>CPA Modules #1-4 (each)</td>
<td>250.00</td>
</tr>
<tr>
<td>First Class Mall</td>
<td>NEW 74.95</td>
</tr>
</tbody>
</table>

### DATA MOST

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snap Attack</td>
<td>29.95</td>
</tr>
<tr>
<td>Thief</td>
<td>29.95</td>
</tr>
<tr>
<td>Count Fair</td>
<td>29.95</td>
</tr>
<tr>
<td>Swashbucker</td>
<td>34.95</td>
</tr>
<tr>
<td>Casino</td>
<td>39.95</td>
</tr>
<tr>
<td>Expand-a-Port</td>
<td>69.95</td>
</tr>
<tr>
<td>World Series Baseball</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>Missing Ring</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>Phaser Fire</td>
<td>29.95</td>
</tr>
</tbody>
</table>

### EDU-WARE

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empire I World Builders</td>
<td>32.95</td>
</tr>
<tr>
<td>Empire II Interstellar Sharks</td>
<td>32.95</td>
</tr>
<tr>
<td>Rendezvous</td>
<td>NEW 39.95</td>
</tr>
</tbody>
</table>

### GEBELLI

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firebird</td>
<td>29.95</td>
</tr>
<tr>
<td>Russki Duke</td>
<td>34.95</td>
</tr>
<tr>
<td>Horizon V</td>
<td>34.95</td>
</tr>
<tr>
<td>Neptune</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>Lazer Slik</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>Zenith</td>
<td>NEW 34.95</td>
</tr>
<tr>
<td>Phaser Fire</td>
<td>29.95</td>
</tr>
</tbody>
</table>

### HAYDEN

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sargon II</td>
<td>34.95</td>
</tr>
<tr>
<td>AppleSoft Compiler 3.3</td>
<td>175.00</td>
</tr>
<tr>
<td>Pia Writer</td>
<td>149.95</td>
</tr>
</tbody>
</table>

### INFOCOM

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zork I</td>
<td>39.95</td>
</tr>
<tr>
<td>Zork II</td>
<td>39.95</td>
</tr>
<tr>
<td>Deadline</td>
<td>49.95</td>
</tr>
</tbody>
</table>

### MICROSOFT

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adventure</td>
<td>25.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Typing Tutor</td>
<td>24.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Olympic Decathlon</td>
<td>29.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TASC Compiler</td>
<td>175.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MUSE

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super Text 40/60</td>
<td>175.00</td>
</tr>
<tr>
<td>Castle Wolfenstein</td>
<td>29.95</td>
</tr>
<tr>
<td>Franky</td>
<td>NEW 24.95</td>
</tr>
<tr>
<td>Firebug</td>
<td>NEW 24.95</td>
</tr>
</tbody>
</table>

### ON-LINE SYSTEMS

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hi-Rise Adventures 0-5</td>
<td>15% OFF</td>
</tr>
<tr>
<td>Cross Fire</td>
<td>29.95</td>
</tr>
<tr>
<td>LISA</td>
<td>79.95</td>
</tr>
<tr>
<td>LISA Educational System</td>
<td>119.95</td>
</tr>
<tr>
<td>General Manager</td>
<td>219.95</td>
</tr>
<tr>
<td>Threshold</td>
<td>39.95</td>
</tr>
<tr>
<td>Screen Writer</td>
<td>129.95</td>
</tr>
<tr>
<td>Screen Writer Professional</td>
<td>199.95</td>
</tr>
<tr>
<td>The Dictionary</td>
<td>99.95</td>
</tr>
<tr>
<td>Expeditert II+</td>
<td>134.95</td>
</tr>
<tr>
<td>Java</td>
<td>29.95</td>
</tr>
<tr>
<td>Mousakattack</td>
<td>34.95</td>
</tr>
<tr>
<td>Marauder</td>
<td>34.95</td>
</tr>
<tr>
<td>Ultima II</td>
<td>59.95</td>
</tr>
<tr>
<td>EPF IV</td>
<td>59.95</td>
</tr>
<tr>
<td>Frogger</td>
<td>39.95</td>
</tr>
<tr>
<td>Cannonball Blitz</td>
<td>39.95</td>
</tr>
<tr>
<td>The Artist</td>
<td>79.95</td>
</tr>
<tr>
<td>LostPak</td>
<td>34.95</td>
</tr>
<tr>
<td>Pest Patrol</td>
<td>29.95</td>
</tr>
</tbody>
</table>

### VISICORP

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visiplot</td>
<td>200.00</td>
</tr>
<tr>
<td>VisiTravel/Visiplot</td>
<td>300.00</td>
</tr>
<tr>
<td>Visidex</td>
<td>250.00</td>
</tr>
<tr>
<td>Visilab</td>
<td>100.00</td>
</tr>
<tr>
<td>Visicad 3.3</td>
<td>250.00</td>
</tr>
<tr>
<td>Visifile</td>
<td>250.00</td>
</tr>
</tbody>
</table>

### SENSIBLE

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super Disk Copy III</td>
<td>30.00</td>
</tr>
<tr>
<td>Multi-Disk Catalog</td>
<td>25.00</td>
</tr>
<tr>
<td>Sensible Speller</td>
<td>125.00</td>
</tr>
<tr>
<td>Image Printer</td>
<td>NEW 40.00</td>
</tr>
</tbody>
</table>

### SIR-TECH

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wizardry</td>
<td>49.95</td>
</tr>
<tr>
<td>Knight of Diamonds</td>
<td>34.95</td>
</tr>
<tr>
<td>20th Century Software</td>
<td>29.70</td>
</tr>
</tbody>
</table>

### SOFTWARE PUBLISHING

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFS</td>
<td>125.00</td>
</tr>
<tr>
<td>PFS Report</td>
<td>95.00</td>
</tr>
<tr>
<td>PFS Graph</td>
<td>NEW 125.00</td>
</tr>
</tbody>
</table>

### STONEWARE

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Master</td>
<td>229.00</td>
</tr>
<tr>
<td>DB Master Utility Pak</td>
<td>99.00</td>
</tr>
<tr>
<td>DB Master Utility Pak 2</td>
<td>99.00</td>
</tr>
<tr>
<td>DB Master start pak</td>
<td>99.00</td>
</tr>
<tr>
<td>Graphics Processing System</td>
<td>59.95</td>
</tr>
</tbody>
</table>

### SIRIUS SOFTWARE

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gogon</td>
<td>39.95</td>
</tr>
<tr>
<td>Sneakers</td>
<td>29.95</td>
</tr>
<tr>
<td>Epoch</td>
<td>34.95</td>
</tr>
<tr>
<td>Cop's &amp; Robbers</td>
<td>34.95</td>
</tr>
<tr>
<td>Beer Run</td>
<td>34.95</td>
</tr>
<tr>
<td>Haddy</td>
<td>34.95</td>
</tr>
<tr>
<td>The Joypoint</td>
<td>74.95</td>
</tr>
<tr>
<td>Twerp</td>
<td>29.95</td>
</tr>
<tr>
<td>Snake Bite</td>
<td>29.95</td>
</tr>
<tr>
<td>Audex</td>
<td>29.95</td>
</tr>
<tr>
<td>Kabul Spy</td>
<td>34.95</td>
</tr>
<tr>
<td>Minotaur</td>
<td>34.95</td>
</tr>
<tr>
<td>Bandits</td>
<td>34.95</td>
</tr>
<tr>
<td>Escape from Rungistan</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>Lemmings</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>JellyFish</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>Cycloid</td>
<td>NEW 29.95</td>
</tr>
<tr>
<td>Fly Wars</td>
<td>NEW 29.95</td>
</tr>
</tbody>
</table>

### STRATEGIC SIMULATIONS

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartels &amp; Outlaws</td>
<td>39.95</td>
</tr>
<tr>
<td>Computer Quarterback (2nd Ed)</td>
<td>39.95</td>
</tr>
<tr>
<td>The Shattered Alliance</td>
<td>39.95</td>
</tr>
<tr>
<td>Computer Baseball</td>
<td>39.95</td>
</tr>
<tr>
<td>Operation Apocalypse</td>
<td>39.95</td>
</tr>
<tr>
<td>Southern Command</td>
<td>39.95</td>
</tr>
<tr>
<td>Napoleon's Campaigns</td>
<td>39.95</td>
</tr>
<tr>
<td>Cytom Master's</td>
<td>39.95</td>
</tr>
<tr>
<td>Galactic Gladiators</td>
<td>39.95</td>
</tr>
<tr>
<td>SELIES</td>
<td>NEW 39.95</td>
</tr>
<tr>
<td>The Cosmic Balance</td>
<td>NEW 39.95</td>
</tr>
</tbody>
</table>

### TG PRODUCTS

<table>
<thead>
<tr>
<th>Retail Price</th>
<th>Our Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game Paddles</td>
<td>39.95</td>
</tr>
<tr>
<td>JoySteer</td>
<td>59.95</td>
</tr>
<tr>
<td>Select-a-Port</td>
<td>59.95</td>
</tr>
</tbody>
</table>

We also carry complete lines from the following companies:

- **ADVENTURE INTERNATIONAL**
- **ARTSCI**
- **ASHTON-TATE**
- **AUTOMATED**
- **BEAGLE BUDGECO**
- **CAL PACIFIC**
- **HOWARD**
- **INNOVATIVE**
- **MICROLAB**
- **MICROPRO**
- **PENGUIN**
- **PHOENIX**
- **PICCADILLY**
- **QUALITY**
- **SENTIENT**
- **SIERRA SOUTHEASTERN**
- **SUBLIGIC SOUTHWESTERN**
- **VIDEX SYNERGISTIC**
- **SYSTEMS PLUS**

If you don't see it, Ask Us!

### CALSOFT

346 N. Kanan Rd. #103  
Agoura, CA 91301

Call Toll Free: (800) 423-5290  **In California:** (213) 991-9641

We accept Mastercard & Visa (include # and Expiration Date), check, COD ($1.50 extra), or Money Order. California residents add 8% sales tax. Include $2.00 for shipping (UPS Blue Label $3.00, Canada $6.00, other foreign countries $10.00).

* Sale prices are through October only! Prices subject to change without notice.
Gameline

Seipentine
By David Snider.

The beauty of Seipentine is twofold. To begin with, the game is devilishly addicting, being endowed with the quality that makes arcade games worth the bother. You know that if you hadn’t made that one wrong move you would still be alive, so you find yourself trying again and again. There is a definite carrot-dangling-on-a-stick quality to this game. The other nice thing about the program is how well done it is.

Imagine Chinese dragons squaring off in the midst of a parade—tails whipping around corners, tight circular chases with sinuous ninety-degree turns, and some beautifully exasperating film-flam as rival serpents cross paths and come out heads at each end.

Seipentine is a fresh twist in maze games. You chase snakes’ tails and they chase yours. It stars two teams of three snakes each: your team is blue and theirs is orange. An interesting aspect of the competition is that both sides are at root identically matched. Only fate (and the author, who also wrote Midnight Magic) has made your team the good guys.

Of course, it would only be a wrestling match if both sides weren’t given handicaps. Orange serpents turn green when you’re more pesky snakes if not eaten by you. Yours are blue, and when you keep an eye on your hero and everyone’s lengths. You’ll have to determine when the enemy is head-on edible by knowing when it’s shorter than you are.

A maze game played on a joystick can be a slippery chore. Seipentine responds crisply to joystick as well as keyboard. It’s good, clean fun and it’ll make you reckless. Heavy objects lying near the computer should be removed to lessen the temptation to halt the inexorable march of the slinky green bandits.

Between the frogs, the speckled eggs, and the serpents’ heads, you can score a bellyful of points if you’re clever. You can also get your little tail nipped off quite often if you’re not. So stay hungry, pray the frogs are running, and keep your hind end out of trouble.

Choplifter was a hard act to follow. Seipentine is not quite the graphics tour de force that its whirlwind predecessor was, but it’s twice the game. And that’s no mean feat.

Canyon Climber
By Tim Ferris.

Danger lurks as you begin your journey from the bottom of the Colorado River up the walls of the Grand Canyon in Canyon Climber. In your quest to reach the top of the canyon, you must overcome crazy mountain goats, Indian archers, and eagle-eyed birds who drop bricks. Good athletic skills and plenty of luck are the two keys to success in this challenging one-player Atari game.

As you start your dangerous ascent, your first task is to lay explosives across four towering bridges. It might sound easy, but there are these big-horned mountain goats that patrol the bridges in pairs. When the confrontation between man and animal occurs, your only defense is to attempt to hurdle the tops of the goats’ sharp horns. If your timing isn’t perfect, the goat will send you tumbling down to the pits of the river. Once all the charges are set, you can gleefully push the plunger and everything goes!

On the next level, mighty Indian warriors constantly bombard you with arrows. You have the option of jumping the arrows or grabbing a shield to block them. If you choose the athletic option, you must jump with great precision or be knocked off the edge of the cliff. Although the shields provide good protection, they can disappear at any time between three and twelve seconds. It is quite easy to be struck right between the eyes as the shield vanishes with no warning.

Should you survive the Indians, you reach the gorges and can see the spacious blue sky at the top of the canyon. But the sky is full of birds who are dropping bricks with excellent aim! With no rest or protection, you must jump with great precision or be knocked off the edge of the cliff. Although the shields provide good protection, they can disappear at any time between three and twelve seconds. It is quite easy to be struck right between the eyes as the shield vanishes with no warning.

On all the levels are ladders to climb, which provide occasional protection from the goats and arrows. Don’t get caught on the ladders in the gorges or the falling bricks will get you. The longer it takes to complete the three levels, the faster the obstacles come at you, be quick and sure-footed. With each successful journey up the canyon the natives become even more protective of their peaceful environment. Canyon Climber fits right in with the “climbing craze” sweeping the arcades today. Although the graphics are not as good as some of their other games, Datasoft has given us a game that is fun to play and makes good use of the computer’s animation and musical capabilities.

Atari 400 or 800; 16K; disk or cassette. $29.95 from Datasoft, 19519 Business Center Drive, Northridge, CA 91324; (213) 701-5161.
Galaxy of Games

By John Savolaine, Don Dilley, and Jerry and Lori Wilkerson.

For those who want to leave the world of adventuring, target shooting, and dot-gobbling games, Galaxy of Games offers a potpourri of four games to relax with. There are no time limits, no high scores to beat, nothing to kill—although you wouldn’t know it from the names: Hangman, One-Armed Bandit, Skunk, and Jacks.

Hangman is the traditional game in which the player tries to guess a word the computer has chosen by filling in letters. If you guess right, the appropriate blanks are filled in. Each wrong guess adds a part to the body being hung. If the body is complete, you lose. In this version of Hangman, the computer shows the words and letters that have been tried.

One-Armed Bandit simulates a slot machine; the object is to win money. The computer shows the results of the last spin and the player’s cash. Each turn, you bet between one and a hundred dollars. If you lose all your money, you can play on the computer’s credit. The game is slow, but the flashing lights and handle movement that slow it down do add realism.

Skunk is a challenging dice game either solo or with as many as six players. The computer rolls six dice. You must try to score as many points as possible based on the value of the dice and their patterns (three of a kind, three pairs, six ones, and so on). Dice with point value may be saved and their points added to your score; then you continue to roll any remaining dice or another six. Your turn ends either when you choose to end it or when you’re skunked. If you’re skunked, you lose all the points you scored that turn.

Skunk makes the sound of rolling dice and even plays tunes.

One to five people may play Jacks, a card game in which the object is to get the least amount of points. Jacks are valuable because they count no points; all other cards count their face value or ten. This program contains a bug, but it’s triggered only by human error: if you try to take a card from the discard pile when there are no cards there, the program kicks out. Still, Jacks is an exciting game.

These games are excellent in detail and playability. Each game in Galaxy of Games is challenging in its way. The games are especially good for youngsters because the instructions are easy to understand. At four games for the price of one, this is a real bargain. TRS-80 models I and III; 16K; cassette or disk. $14.95 from Hayden Microcomputer Gameware, 50 Essex Street, Rochelle Park, NJ 07662; (800) 631-0856.

Marauder

By Rorke Weigandt and Eric Hammond.

This one’s for the plunder and pillage crowd. The idea is simple: Run around the galaxy trashing planets. Barring any philosophical qualms one may have about extinguishing alien fauna, destroying planets can be fun. Besides, the guards are all robots and mean.

Marauder is a two-part space seek-and-destroy game in which you are the marauder, first controlling a spacecraft blasting away at planetary defenses and then running about in underground tunnels trying to destroy the planet’s main power supply. While there is no real strategy or thought involved in the game, it can be a lot of fun. Players can choose to play the first part or second part exclusively or the whole game.

The first part of the game puts you in control of a ship hovering above the surface of a nameless planet. You must destroy all of the defenses by shooting through a force field and then bombing missile launch sites. The ground installations fire slow-moving heat-seeking mines that follow you, missiles that detonate at the point where they expected you to be, laser beams that rise perpendicular to the planet’s surface, and indestructible fireballs that home in on you but also constantly rise, so that a bit of skillful maneuvering will get you below them and out of harm’s way. Your ship is equipped with a temporary overdrive for those breathless moments when a fireball seems to have your number.

Once you have wiped the surface of the planet clean, you must dive through the wreckage of the fireball launcher to enter the un-
derground maze. The maze is a relatively simple one populated by a particularly bellicose species of robot. You can’t see the vicious droids until they are in line of sight, and vice versa.

Your mission, once in the maze, is to locate the power source and destroy it with your hand weapon. Once you have blown up the power source, you have just time to get back to your ship before the roof begins to cave in.

One peculiarity of the game is that, while your marauder carries his weapon in his right hand, the robots have center-mounted blast-ers; face-to-face confrontations are invariably fatal to you, so you must be extraordinarily careful not to square off with a robot.

All in all, Marauder is as good as any mindless arcade game, and better than most. Its strength and weakness are one and the same: to succeed one needs good reflexes only; there are no strategies to be developed, no intelligent opposition to outfox. If you are looking for entertainment without mental calisthenics, this game should provide hours and hours of fun.

Apple II, Apple II Plus; 48K; disk. $34.95 from Sierra On-Line, 36575 Mudge Ranch Road, Coarsegold, CA 93614; (209) 683-6858.

Hockey

Solitaire computer games, dependent on the interaction between player and machine, require a certain level of sophistication to remain interesting. Those needing at least two human players, on the other hand, provide much of their enjoyment because of the head-to-head combat between competitive rivals. Since the computer acts only as the medium on which the challenge is met, the game itself need not be intricate to be successful.

Such is the case with Gamma Software’s Hockey, a simple game with great potential for enjoyment.

This is standard ice hockey, although the teams are reduced to four players each: three forwards and a goalie. A standard joystick controls one of the forwards, either the man with the puck or his opposing guard. The other two men on each side are monitored by the computer and manage to follow the flow of action fairly well. Should the puck be passed, the passer becomes computer-controlled and the passees then responds to the joystick.

The goalie, who guards the front of the cage, cannot hold the puck but deflects it back onto the playfield. It can then be picked up when a forward touches it with his stick. By the same token, the puck may be stolen by a defensive man who touches it with his stick.

The game begins with a face-off at the center of the rink, a maneuver that’s repeated following each goal (worth one point). Should the game end in a tie, the computer immediately moves into a two-minute sudden death playoff. The game length itself can be optioned at three, five, or eight minutes.

Up to four players can participate. Partners (each with a joystick) divide team responsibilities: one controls the forwards, while the other acts as goalie. This option actually is preferable, since it is difficult to guard one’s goal and try to reclaim the puck when a single joystick controls all four men.

The simple graphics, in which a skater resembles a praying mantis, are enough to differentiate man from stick. The sound effects are more primitive: the crowd’s cheers sound more like pounding surf, and the match ends with only a brief blaaat from the buzzer. Far better is the game’s beginning, when the computer plays the last few bars of “The Star-Spangled Banner” (without, fortunately, an attempt to reproduce Ethel Merman’s vocal).

On the negative side, it is very difficult for the defender to spot his flashing man (which should be guarding the puck) amid the fren-etic activity of a goal rush. The hint in the instruction manual (relax the joystick to the center position and look closely) is of limited use; a goal can be scored in the meantime. It’s also unfortunate that the game does not allow forwards to skate behind the goal, as that maneuver produces intriguing shots. Finally, there is no running game score and no halves or quarters; the game starts from scratch at the end of each match.
In spite of those shortcomings, Hockey has an infectious streak to it, and it’s nice to see a game that can involve so many people. Hockey may look primitive, but that’s appropriate for a game that caters to an instinct as primitive as competitive drive.

Hazard Run
By Dennis Zander.

What hath Burt Reynolds wrought?
Most of the world probably has had it with good ol’ boys and slack-jawed law enforcement officers. Today’s software industry, however, is not apt to leave a generic stone unturned. Those thrilled by converted Mustangs leaping large bodies of water now have the opportunity to duplicate that feat of derring-do with their home computers.

Hazard Run, unfortunately, is not up to its film forerunners. Zander’s program is rather small (144 sectors of Basic) and has a high frustration quotient, which might result in players hurling joysticks through their monitors.

This solitary game employs only three joystick commands: left, right, and fire. After starting the run, the firing button becomes the brake, and the stick commands correspond to the image on the screen. Your car will always accelerate unless the brake is applied. The car remains at the base of the screen while the “course” scrolls downward. Vehicle-crunching hazards include tree trunks, rocks, fences, police cars, canyon walls, bridge struts, and some chickens (fryers, no doubt). Hitting such an object dead on will flip and destroy one of your five cars; sideswiping will tip the car onto two wheels, making it narrower and able to squeeze through tight spots.

Hazard Run’s major problem is the manner in which the joystick steers the car. Rather than simulating the smooth turning action of front-wheel or rear-wheel drive, the car pivots at its center and then back to vertical. This pendular action makes the car a wider target while negotiating a turn, increasing the chances of hitting the object you’re trying to avoid.

Almost worse is the lack of anticipation time. Jumping a river must be done at no less than sixty miles per hour (something you won’t learn from the sparse instructions; just remember, you read it here first) or the car will crash into the far embankment, ending that particular run. Also, the car will not reach sixty from a standing start in the space provided by one screen’s worth of graphics. In other words, there is not sufficient time to negotiate several of the rivers. The game must be partially memorized before it can be completed successfully.

While there are four different runs, ranging from “chicken run” (the shortest and easiest) to “Hell’s canyon” (the longest and hardest), each merely builds on those preceding it. Extra gas will be needed for the longer trips, and fuel may be picked up by running over hazardously placed gas cans.

When jumping a river, point of view switches from bird’s-eye to sidelong. If the car makes the jump, action immediately returns to the aerial view, so be prepared to move quickly. The program plays the same few bars from “Dixie” regardless of the jump’s outcome; it would be nicer (and preferable psychologically) to hear something different as a reward for making the jump.

Clearing the final river results in the anticlimactic delight of watching the pursuing police car land in the drink. It’s a cute touch, but one that rapidly wears thin . . . as does the wait required before each run while the two drivers enter the car.

Hazard Run is just a bit too primitive for today’s market and will be of interest mostly to neophytes and young children.

Our Guarantee: if it doesn’t boot, we will replace it free.

1-800-845-5147

Free Catalog
1340 HPLOT X6,Y6 TO X7,Y7: IF R <> RR THEN HPLOT X2,Y2 TO X3,Y3
1350 GOTO 1400
1360 GOSUB 1800:XX = XE:YY = YE: GOSUB 1600: IF T(D) = 1 THEN 1400
1370 IF T(D) = 0 THEN 1390
1371 IF R <> RR THEN 1380
1372 HPLOT 181,143 TO 181,31 TO 183,31
1373 GOTO 1390
1380 HPLOT X6 + XV,Y7 TO X6 + XV,Y6 + XV TO X2,Y6 + XV
1390 HPLOT X2,Y6 TO X7,Y6 TO X7,Y7 TO X3,Y7
1400 X1 = X5:Y1 = Y5:X2 = X6:Y2 = Y6
1410 X3 = X7:Y3 = Y7:X4 = X8:Y4 = Y8
1420 XX = X:YY = Y: GOSUB 1600:N = T(D)
1430 IF N <> 1 THEN 1500
1440 IF D = 1 THEN Y = Y — 1
1450 IF D = 2 THEN X = X + 1
1460 IF D = 3 THEN Y = Y + 1
1470 IF D = 4 THEN X = X — 1
1480 R = R — 1: IF R = 0 THEN 1540
1490 GOTO 1090
1500 HPLOT X5,Y5 TO X6,Y6 TO X7,Y7 TO X8,Y8 TO X5,Y5
1510 IF N <> 2 THEN 1540
1520 XV = (X2 — X1) / 8:YV = (Y4 — Y1) / 8
1530 HPLOT X8 + XV,Y8 TO X5 + XV,Y5 + YV TO X6 — XV,Y6 + YV TO X7 — XV,Y7
1540 RETURN
1600 T = A%(XX,YY)
1610 IF T < 0 THEN T = T + 65536
1620 S = INT (T / 4096)
1630 T = T — S * 4096
1640 T(1) = INT (T / 512)
1650 T = T — T(1) * 512
1660 T(2) = INT (T / 64)
1670 T = T — T(2) * 64
1680 T(3) = INT (T / 8)
1690 T(4) = T — T(3) * 8
1700 RETURN
1800 XE = X:YE = Y: IF P = 1 THEN YE = YE — 1
1810 IF P = 2 THEN XE = XE + 1
1820 IF P = 3 THEN YE = YE + 1
1830 IF P = 4 THEN XE = XE — 1
1840 RETURN
3000 FOR I = 1 TO 10
3010 FOR J = 1 TO 10
3020 READ A%(J,I)
3030 NEXT J
3040 NEXT I
3100 DATA 72,65,81,73,17,72,9,72,73,9
3110 DATA 520,72,1033,520,1096,513,576,513,512,520
3120 DATA 520,576,513,520,520,8,128,74,9,520
3130 DATA 584,65,513,576,577,1,576,513,520
3140 DATA 520,64,129,74,9,64,137,74,1,520
3150 DATA 520,72,9,592,529,80,513,512,80,513
3160 DATA 648,514,576,1025,1032,1088,137,74,1097,9
3170 DATA 640,66,129,66,577,65,521,576,577,513
3180 DATA 136,74,137,10,72,65,577,65,9,8
3190 DATA 512,576,513,576,577,65,1,576,513

Maze in 3-D

from page 14

Brian Fitzgerald coauthored Taxman and Sheila with Greg Autry. The two founded H.A.L. Labs in 1981, following Brian's graduation from the University of California, Irvine, with a degree in physics and computer science. He has been playing with Apples since 1977.
New Players

Lionel Raff often felt frustrated by the trial and error involved in solving computer games. He objected to luck being a more important factor in successful play than logic or creativity. He took steps to solve this problem for himself and other gamers by founding a software company with the goal of programming games that yield their secrets only to the application of logic—while retaining a sense of adventure and excitement. The company is Raff-Craft.

Lionel Raff is a games player par excellence; a ranked chess player and life master in the American Contract Bridge League. He is also an internationally published Regents Professor in Chemistry and was a National Science Foundation postdoctoral fellow at Columbia University.

Raff's interest in computers began in 1959 when the computers he worked with wouldn't fit in his house, let alone on his game table. He has followed their development with awe and excitement, and by last January he could no longer resist the temptation to get his hand into the computer-gaming action. In the wide-open spaces of Stillwater, Oklahoma, Raff-Craft was born.

They're Off! The company's first venture is Derby Downs, allowing from one to ten players to play an endless series of horse races—one day's program at a time. Each day is unique—with all the excitement and challenge of a day at the races. The players who can evaluate speed, endurance, quality of the jockey, post position, weight, and other factors that influence the outcome of real races are the most likely to win. The tote board allows for quinellas, exactas, perfectas, daily doubles, across the boards—just about all the betting forms available at the races except the new pick six.

Raff-Craft's second project, tentatively titled Zartan's Cube, will allow adventurers to compete with a consummate gamemaster for treasures and glory. Each adventure is different to the extent that problems change each trip, and your intelligence is increased by successful encounters and decreased by unsuccessful ones.

For the future, Raff-Craft is negotiating with the copyright holder of Bracket, the word game of deductive reasoning, for the right to produce the computer version of the game.

Tana Maxwell, vice president of Raff-Craft and a psychologist, is currently involved in the business end of the company but plans to work with Raff on some realistic, fun, and educational psychological interaction games.

For education and entertainment, the Apple seems to have found a prime source of edifying product for its inexhaustible appetite.

APPLE® OWNERS: APOGEE OFFERS

POPULAR SOFTWARE AT BIGGEST DISCOUNTS EVER!

Magic Window by Artasci
Voted most popular word processor by Soft Talk in April issue.
List Price $99.95
APOGEE Price $69.97

Kabul Spy by Sirius Software
A terrific adventure game.
List Price $34.95
APOGEE Price $24.75

Raster Blaster by Budgoco
Plays like real pinball, with excellent graphics.
List Price $29.95
APOGEE Price $19.97

Sensible Speller by Sensible Software
Correct school work and documents. New greatly expanded program.
List Price $125.00
APOGEE Price $87.50

Trickshot by Innovative Designs
You can save your favorite pocket pool trick shots on disk.
List Price $39.95
APOGEE Price $28.97

T. G. Joysticks
Improves your game skills.
List Price $59.95
APOGEE Price $41.97

Add $2.00 for postage and handling. Illinois residents add 6% sales tax.
Send for our complete catalog.
P.O. Box 71, Morton Grove, IL 60053
24 hr. service Order toll free 1-800-621-0660
In Illinois call 1-800-572-0444
We Accept VISA or MASTERCARD
® trademark of Apple Computer Inc.

If you have a favorite program that isn't listed, send us your name, address, and program name, and we will send you our special introductory price. One program per customer.

APOGEE SOFTWARE DISTRIBUTORS
P.O. BOX 71
MORTON GROVE, IL 60053
Adventures in Adventuring

The Thing's the Thing

by KEN ROSE

Shhh!!!!

Be very quiet and you may hear the sound of muffled footprints, the scratching of fingernails against a windowpane, the quiet breathing of a stranger hidden in the next room. Look!

Is that the body of Sir Malcolm Bainbridge slumped over his desk with a Kris plunged into his heart? Can it be that the Forsythe emerald is missing and that the infamous Reginald Forsythe was seen fleeing from his former wife’s home? Think!

Whodunit? Can you walk in the footsteps of Holmes and Father Brown and Marlowe and Templar and Spade and Fell and Chan?

Of course you can. You, of all the people who are interested in the modern magic of the computer, can turn that logical, cool, well-ordered mind to solving mysteries that are far beyond the abilities of mortal men.

Our adventure in adventuring thus far has had a theme of fantasy with a smattering of science fiction thrown in. If you’ve been following these articles, you’re now familiar with the techniques used in resolving conflicts, having the computer recognize commands, moving around, measuring time, and designing simple text adventure games around these conflicts.

This month we’ll deal with the handling of objects in an adventure program—how to pick things up, use things, and drop them. Our adventure has a mystery theme, in that we will be trying to recover a stolen diamond of little value.

You’ll probably notice that as these programs become more sophisticated, they become longer. Most of the length is taken up not by the logic of the program but by the descriptive words needed to flesh out the story. In fact, in most commercial adventure games, the program takes up very little of the disk. The bulk of the disk is filled with the wordy descriptions used to make the game interesting.

This month’s program is written in sections. The rest of the article will deal with what each section does and how it does it. Although this program was written on an Apple II, it’s easily adaptable to any small computer using Basic. The program fits into any machine that has 4K of memory. As we go along, we’ll comment on those things peculiar to Apple Basic and what they mean.

So, without further ado, let us analyze: The Case of the Pig-headed Diamond.

Lines 10 through 150: Line 10 clears the screen and moves the cursor to the upper left-hand corner.

Line 15 goes to the subroutine at line 9000 which describes what the story is all about.

Line 20 dimensions variables. These variables are for the directions used in the fifteen rooms, the room names (RNS), the eleven nouns that the program will use (noun$ found in data line 10160), the descriptions for each of the nouns (des$), and, finally, the room locations for each of the objects (ob) that will be used.

Lines 30 through 140 read in the data for the dimensioned variables. Line 150 does several things. It sets the variable that keeps track of which room we’re in (variable R) to 1, it uses Apple language so that the top line won’t scroll off the screen (poke 34,1), and it goes to line 1490. Line 1490 prints the room description at the top of the screen.

Lines 1000 through 1070 are the parser, the thing that takes the words and splits them into verbs and nouns. A longer explanation can be found in the March issue of Softline (“Please Parse the Zork”), but, briefly, what the parser does is find the blank between the words and set up two separate variables, one for the noun (n1$) and one for the verb (v1$).

Lines 1200 through 1390 are devoted to sending the program to the routines that carry out instructions called for by the verb part of the command. The verbs used in this program are go, the name of a direction in which to go, inventory, quit, look, get, drop, climb, and dig. If the verb is none of these, this routine tells the program to go back to the command line. Note that line 1225 requires two-word commands except in the cases of inventory, quit, go, and the names of directions.

Lines 1400 through 1490 are the lines that control moving around. This is detailed in the May Softline in “From Here to There and Back Again.” Note that in line 1490 special Apple language is again used. Home means clear the screen except for the first line, which was locked in place by poke 34. Vtab 1 sends the cursor to line 1 and call -868 clears the line from the cursor to the end of that line. Then the room name is printed.

Lines 1500 through 1530 are used to print the names of the objects that are found in each room. There are eleven objects used in this game and they are found in line 10160. Each object also has an object variable number. The first object in the list is an ice cube. The variable noun$(1) contains the name ice cubes, the variable des$(1) contains the description of the object ice cubes and the variable ob$(1) contains the location of the object ice cubes. Since there are fifteen rooms, if the value of ob$(1) is 12, then the object is associated with room 12.

The sixteenth room does not exist as a room, but is the player himself. Thus, if we make the value of ob$(1) equal to 16, the ice cubes are associated with the player. You see that line 1510 refers the program to the subroutine located at lines 2200 through 2300. Line 1510 makes X equal to the room number you’re in. The subroutine at line 2200 then checks to see which of the eleven objects has the same value as the room number you’re in and prints them out. And that’s how the objects are shown when you move into a particular room.

Lines 1600 through 1670 interpret the look command. Line 1610 sends us to the subroutine at line 2100, which checks to make sure the noun is valid. If it is, line 1640 then prints the appropriate description for that noun, provided line 1620 has determined either
that the noun is in the room we're in or that we are carrying it around. If the object is in the room and we're not carrying it, line 1630 prints, "That isn't here." Line 1660 gives us the ice cubes if we look in the refrigerator by making the ice cubes (ob-1) equal to our room value. This line also sets a counter, c3, to 1, so that we are prevented by line 1650 from getting more ice cubes.

Lines 1700 through 1760 allow us to pick up or get objects. Once again, we're sent to the subroutine at 2100 to check out the validity of the noun. Since we're room 16, if the value of the noun (ob-number) is equal to the room we're in, the value is changed to 16, which means we're now carrying the object around. If the value of the noun is not equal to the value of the room, then the particular object we're trying to get is not in the room, so we can't change it to 16, so we can't get it. Simple, huh?

Lines 1900 through 1940 list out, or inventory, the objects we're carrying. This is done simply by going to the same subroutine that lists out the objects when we enter a room, subroutine 2200, and having it list out the objects in room 16—which is us.

Lines 2000 through 2080 are the drop routine. The drop routine is the opposite of the get routine. The drop routine checks first to see if the noun is legitimate and then to see if the object value is 16, which means we've got it. If the noun is legitimate and we have it, its value is changed to that of the room we're in, which means we're no longer carrying the object. Note that if we change the values of either object 9 or object 1 to 1 (which means we drop it while we're in room 1), we go to the win game routine.

Lines 2100 through 2180 check the noun variable n1$ against the list of nouns to verify it and then return to the main part of the program. There are a couple of special situations here, though. Certain objects—including the refrigerator, the stove, and the chandelier—can only be looked at, not moved. Setting the variable X in line 2150 will allow an object's room value to be changed. Line 2120 prevents this from happening by bypassing line 2150. Line 2130 prevents you from getting the ladder until you have the shovel.

Finally a word about the pop in line 2180. Normally, you use the word return to return from a gosub command. Note that we are not merely returning from the gosub in line 2180 but are going back to the command line 1010. In order to clear the gosub, Apple Basic uses the word pop.

Lines 2200 through 2330 print out the names of nouns when the player enters a room or does an inventory. This is pretty straightforward, except for the setting of the variable y1 if a noun is printed. If y is not 1, it means that no nouns have been found and so nothing is printed out by line 2320.

Lines 2500 through 2580 are the climbing the ladder and climbing the bucket routines. Line 2520 checks to make sure you have either the ladder or the bucket. If you try to climb anything else, line 2550 gives you a smart-alecky message. Line 2570 is the ladder climbing routine. Note that after you climb the ladder you are given the pendants. Note, too, that this line also changes the description, so that if you look at the chandelier again you will see no more pendants. It also sets counter c1 to 1 so that if you try to climb the ladder again, line 2560 will not allow you to climb the ladder for more pendants.

Lines 2600 through 2640 are the digging routine. If you look at line 2620, you'll see that if you have the shovel (ob-10) and are in the right spot (room 4), you can dig up the ladder. Note that when you do, counter c1 is set to 1. This prevents you from digging up the ladder again via line 2610. Note that if you aren't in room 4 or don't have the shovel, line 2630 gives you another smart-alecky message.

Lines 4000 through 4050 are the winning routine. Figure them out yourself.

As you play the game, note the message you get when you look at the bucket. Although there is no "Kick the Bucket" routine in the program as listed here, it would be a good exercise to add one and see how you do.

Happy adventuring.

29
2560 IF C1 = 1 THEN GOTO 2580
2550 PRINT : PRINT "YOU CAN'T CLIMB A ":PRINT
2540 IF N1$ = NOUN$(2) THEN GOTO 2560
2530 IF N1$ = NOUN$(4) THEN PRINT : PRINT "THE BUCKET IS
2510 IF OB(2) = 16 OR OB(4) = 16 THEN GOTO 2530
2500 REM CLIMBING ROUTINE
2310 IF Y = 0 THEN PRINT "NOTHING" :RETURN
2300 IF X = OB(10) THEN PRINT "SHOVEL":Y = 1
2280 IF X = OB(8) THEN PRINT "PIG":Y = 1
2290 IF X = OB(9) THEN PRINT "PENDANTS":Y = 1
2250 IF X = OB(5) THEN PRINT "STOVE":Y = 1
2260 IF X = OB(6) THEN PRINT "REFRIGERATOR":Y = 1
2270 IF X = OB(7) THEN PRINT "CHANDELIER":Y = 1
2280 IF X = OB(8) THEN PRINT "PIG":Y = 1
2290 IF X = OB(9) THEN PRINT "PENDANTS":Y = 1
2300 IF X = OB(10) THEN PRINT "SHOVEL":Y = 1
2310 IF Y = 0 THEN PRINT "NOTHING" :RETURN
2320 PRINT
2330 Y = 0: RETURN
2340 REM GET STARTS HERE
1700 GOSUB 2100
1710 REM GET STARTS HERE
1720 IF OB(X) = R OR OB(X) = 1 THEN GOTO 1740
1730 PRINT : PRINT "THAT ISN'T HERE.":PRINT : GOTO 1760
1740 IF X > 0 THEN OB(X) = 16
1750 PRINT : PRINT "YOU'VE GOT THE ":,NOUN$(X);":PRINT
1760 X = 0: GOTO 1010
1900 REM INVENTORY ROUTINE
1910 X = 16
1920 PRINT : PRINT "YOU ARE CARRYING: ": PRINT
1930 GOSUB 2200
1940 GOTO 1010
2000 REM DROP ROUTINE
2010 GOSUB 2100
2020 IF OB(X) = 16 THEN GOTO 2040
2030 PRINT : PRINT "I DON'T HAVE THAT.": PRINT : GOTO 2060
2040 OB(X) = R
2050 PRINT : PRINT "I DON'T HAVE THAT.": PRINT : GOTO 2060
2060 IF OB(9) = 1 THEN GOTO 4010
2070 IF OB(1) = 1 THEN GOTO 4020
2080 GOTO 1010
2100 REM ROUTINE FOR CHECKING NOUN
2110 IF V1$ = "LOOK" THEN GOTO 2140
2120 IF N1$ = NOUN$(7) OR N1$ = NOUN$(6) OR N1$ = NOUN$(5) THEN PRINT : PRINT "YOU CAN ONLY LOOK...YOU CAN'T TAKE": PRINT : GOTO 2180
2130 IF N1$ = NOUN$(2) AND C2 = 0 THEN PRINT : PRINT "UGH, IT'S STUCK. YOU'LL HAVE TO DIG IT OUT.": PRINT : GOTO 2180
2140 FOR A = 1 TO 10
2150 IF N1$ = NOUN$(A) THEN X = A:A = 0:RETURN
2160 NEXT
2170 PRINT : PRINT "I DON'T SEE ":NOUN$;":...
2180 POP : PRINT : GOTO 1010
2200 REM ROUTINE FOR LISTING OUT OBJECTS
2210 IF X = OB(1) THEN PRINT "ICECUBES":Y = 1
2220 IF X = OB(2) THEN PRINT "LADDER":Y = 1
2230 IF X = OB(3) THEN PRINT "MATCHES":Y = 1
2240 IF X = OB(4) THEN PRINT "BUCKET":Y = 1
2250 IF X = OB(5) THEN PRINT "STOVE":Y = 1
2260 IF X = OB(6) THEN PRINT "REFRIGERATOR":Y = 1
2270 IF X = OB(7) THEN PRINT "CHANDELIER":Y = 1
2280 IF X = OB(8) THEN PRINT "PIG":Y = 1
2290 IF X = OB(9) THEN PRINT "PENDANTS":Y = 1
2300 IF X = OB(10) THEN PRINT "SHOVEL":Y = 1
2310 IF Y = 0 THEN PRINT "NOTHING":RETURN
2320 PRINT
2330 Y = 0: RETURN
2340 REM CLIMBING ROUTINE
2350 IF OB(2) = 16 OR OB(4) = 16 THEN GOTO 2370
2360 PRINT : PRINT "YOU HAVE TO GET THE PROPER OBJECT
BEFORE YOU CAN CLIMB IT.": PRINT : GOTO 1010
2370 IF N1$ = NOUN$(4) THEN PRINT : PRINT "THE BUCKET IS
VERY UNSTABLE...YOU TRIP, FALL OFF AND BREAK YOUR
GREAT TOE WHICH RENDERS YOU INCAPABLE OF
FINISHING THE GAME.":GOTO 4100
2380 IF N1$ = NOUN$(2) THEN GOTO 2390
2390 PRINT : PRINT "THE CASE OF THE PIG-HEADED
DIAMOND": PRINT
2400 PRINT "YOU HAVE BEEN RETAINED TO RECOVER THE
FAMOUS 'PIG' DIAMOND STOLEN FROM THE SMALL
MANSION OF CARLYSLE SMEDLEY GRUM.": PRINT
2410 PRINT "YOU KNOW THERE IS A DEATH TRAP ON
THE ESTATE SO YOU MUST BE CAREFUL.": PRINT
2420 PRINT "GOOD LUCK!": PRINT "TO PLAY, PRESS ANY KEY": GET A$: RETURN
10000 REM DATA STARTS HERE
10010 DATA 2,0,0,0,0,"DIAMOND BANK...DROP THEM HERE
TO WIN."
10020 DATA 3,1,0,0,0,"LONG SHADY ROAD"
10030 DATA 5,2,0,0,0,"BOTTOM OF HILL"
10040 DATA 0,0,5,0,0,"DUSTY GARAGE"
10050 DATA 8,2,6,4,0,"OPEN FRONT DOOR"
10060 DATA 0,0,5,0,0,"OVERGROWN GARDEN"
10070 DATA 10,0,8,0,0,"LIBRARY"
10080 DATA 11,5,9,7,0,"FRONT HALLWAY"
10090 DATA 12,0,8,0,0,"LIVING ROOM"
10100 DATA 13,7,11,0,0,"BATHROOM"
10110 DATA 14,8,12,10,0,"MIDDLE HALLWAY"
10120 DATA 0,9,0,11,0,"KITCHEN"
10130 DATA 0,0,14,0,0,"BEDROOM"
10140 DATA 0,11,13,0,15,"FAR HALLWAY"
10150 DATA 0,0,0,0,14,"DANK CELLAR"
10160 DATA "ICECUBES","LADDER","MATCHES","BUCKET",
"STOVE","REFRIGERATOR","CHANDELIER","PIG",
"PENDANTS","SHOVEL"
10170 DATA "THEY LOOK LIKE SOME EXTREMELY HARD
FROZEN ICECUBES."","IT IS A RICKETY, SHAKY
LADDER."","THE BOOK OF MATCHES SAYS, 'ENROLL NOW
IN BASIC COMPUTER PROGRAMMING COURSE.' ","THE
BUCKET HAS A SIGN ON IT SAYING 'DON'T KICK ME.'
10180 DATA "IT IS AN OLD FASHIONED GAS STOVE."","THE
REFRIGERATOR CREAKS AND GROANS BUT IT APPEARS TO BE WORKING.
THERE ARE SOME ICECUBES IN IT."","THE CHANDELIER HAS SOME BRIGHT SHINY
DIAMOND-LIKE PENDANTS HANGING FROM IT.""
10190 DATA "IT IS A SHROPSHIRE SHOAT, SOUND ASLEEP,
WHICH WAS HOGGING THE BATHROOM."","THE
PENDANTS APPEAR TO BE DIAMONDS."
10200 DATA "THE SHOVEL IS RUSTY AND LOOKS LIKE YOU
COULD CUT YOURSELF ON IT."
Apple II Graphics

We've talked about how your Apple's memory is laid out, how to poke stuff into memory to effect the display, how to work with binary, hex, and decimal numbers, and how to animate using shapes.

This month, we'll explore the world of byte-move animation. This technique is very different from animating with shape tables and is used in many of today's computer games.

When you create a shape in hi-res, what you are really doing is giving the computer a set of directions to follow when it draws the shape on the screen. (For an explanation of defining shapes, refer to chapter 9 in your Applesoft manual.) Every time the shape is drawn or Xdrawn, your computer follows those directions (for instance, plot the first point, then move up, then move to the left twice without plotting, then plot that point and move down one ...) to recreate the shape on the screen.

That procedure is fine for some applications, but each component instruction must be processed every time the shape is drawn, and that is relatively slow since even a simple shape can easily contain a hundred instructions. Speed is one of the primary requirements for smooth, flicker-free animation in which the figures seem to appear on the screen instantaneously.

Byte-Size Pieces. The idea behind byte-move graphics is to translate any desired figure into the corresponding data values and then poke those values directly onto the screen instead of using shapes to draw the figure. Type in the following program and run it. If you want to spare your fingers, you may omit the rem statements.

```rem
10  REM INITIALIZE Y
20  REM COORDINATES
30  REM
40  Y1% = 1:Y2% = 2:Y3% = 3:Y4% = 4:Y5% = 5:Y6% = 6:Y7% = 7
50  REM
60  REM READ DATA FOR FIGURE
70  REM
80  FOR I = 1 TO 4: REM 4 FRAMES
90  FOR J = 1 TO 7: REM 7 BYTES PER FRAME
100 READ V%(I,J)
110 NEXT J
120 NEXT I
130 REM INITIALIZE ADDRESSES
140 REM OF Y COORDINATES
150 REM
160 Y%(I) = 8192*Y%(I-1) + 256*Y%(I-2) + 32*Y%(I-3) + 8*Y%(I-4) + 2*Y%(I-5) + Y%(I-6) + Y%(I-7)
170 HGR
180 REM
190 REM POKE THE FOUR FRAMES
200 REM
210 FOR I = 1 TO 4
220 POKE Y%(I),V%(I,1):
230 POKE Y%(I),V%(I,2):
240 POKE Y%(I),V%(I,3):
250 POKE Y%(I),V%(I,4):
260 NEXT
270 REM
280 GOTO 210: REM START AGAIN
290 REM
300 DATA 1,2,4,8,16,32,64
310 DATA 8,8,8,8,8,8,8
320 DATA 64,32,16,8,4,2,1
330 DATA 0,0,0,0,127,0,0
```

If you managed to type everything in correctly, you'll see what passes for an airplane propeller spinning in the corner of your monitor screen. You must have noticed the delay in running the program before the animation began. That is a characteristic of byte-move graphics, even in the professional games, and it is caused by the need to initialize several tables before the animation can take place.

In our listing, line 40 sets the seven Y coordinates used in the figure and line 160 assigns the seven corresponding addresses. Lines 80 through 110 set up a table that contains four versions of the prop, each in a different rotation and each using seven screen lines.

To understand this, let's look at a diagram of each of the four frames, where X indicates a screen dot turned on, and — represents an off dot.

Although the figure appears to rotate like a propeller, the program is actually flashing the four frames onto the screen sequentially. It happens quickly enough that your eyes and brain are fooled into thinking the rotation is continuous—the poke is quicker than the eye!

Frame #1 in figure 1 shows the propeller running diagonally, and it also gives the binary bit pattern used to produce each dot pattern (remember, the dots are displayed as the reverse of the bits in each byte) and the equivalent decimal value. If the translation from dot pattern to binary and decimal value overloads your brain, go get something to drink, and then reread the third article in the series where we discuss the (many) peculiarities of hi-res graphics. The decimal values calculated in figure 1 correspond with the data values you see in lines 300 through 310.

The true heart of the program is the loop from lines 210 to 250 where each frame in turn is poked into hi-res screen memory. Using variables in the poke statements obscures the mechanics of what we're doing, but it also enhances the execution, as it takes more time for the computer to generate a number such as 12288 than it does to look that value up in an array.

The other reason for all the variables is that we are going to alter the routine to allow the propeller to be placed at any Y coordinate on the screen. The array Y% will contain the starting addresses for
each line of the screen, and Y1 through Y7 will contain the seven Y coordinates used in the figure.

But let's get back to those poke statements. Line 220 pokes the first and seventh bytes, V%(I,1) and V%(I,7), line 230 pokes the second and sixth, and line 240 pokes the third, fifth, and fourth bytes. The bytes are poked in that peculiar order to improve the image, but you might like the effect obtained by poking the seven bytes in numerical order instead. Try it!

The propeller is an example of stationary animation; that is to say that though the prop moves, it always stays in the same position on the screen as it does so. Most figures you use in a game need to move around the screen, so we'll alter our program shortly to allow that. There are, however, many applications for stationary animation; the Applevision demo on your DOS 3.3 System Master is an example, as is putting a scoreboard on the hi-res screen. In the instance of the scoreboard, the frames would not be pictures of a moving object; instead you would use successive digits.

**Drop the Prop.** With the previous program still in memory, type in the following lines:

```
160 GOSUB 1000: REM CALC ADDRESSES
245 Y1% = Y1% + 1: Y2% = Y2% + 1:
    Y3% = Y3% + 1: Y4% = Y4% + 1:
    Y5% = Y5% + 1: Y6% = Y6% + 1:
    Y7% = Y7% + 1
1000 REM
1010 REM CALCULATE Y
1020 REM COORDINATES
1030 REM
1040 DIM Y%(192)
1050 FOR I = 1 TO 185 STEP 8: READ SA%
1060 FOR J = 0 TO 7: Y%(1 + J) = SA% + J * 1024
1070 NEXT LI
1080 DATA 8192,8320,8448,8576,8704,8832,8960,9088
1090 DATA 8232,8360,8488,8616,8744,8872,9000,9128
1100 DATA 8272,8400,8528,8656,8784,8912,9040,9168
1110 RETURN
```

Lines 1000 through 1110 calculate the starting addresses for each of the 192 lines on the screen in the same way you would find them if you were to use the method described (albeit sketchily) on page 21 of your Apple II Reference Manual. After each frame of the prop is poked onto the screen, line 245 increments each of the seven Y coordinates so that the next frame will appear one line below the last.

Run the modified program, and you'll see the propeller spinning as it drops down the left side of the screen. You will also see a trail of garbage left behind as the figure progresses.

Oh well, you couldn't have a program work right the first time, could you? Most of the figure is erased when the next frame is drawn over it, but since each frame is lowered one line, the top line of each frame remains to haunt you.

The problem is easily remedied by inserting:

```
244 Y0% = Y1%
215 POKE Y0%(Y0%),0
```

Line 244 sets Y0% to the coordinates of the top line, and line 215 pokes a zero into that address in order to erase the old top line. Now, when you run the program, the picture moves down the screen without leaving a trail.

**So Much for the Easy Stuff.** So far you have done a stationary animation and a vertical animation using byte-move, but we have left horizontal animation for last. Type in and run the following routine:

```
10 HGR
20 FOR L = 8192 TO 8231
30 POKE L — 1,0: REM ERASE PREVIOUS BYTE
40 POKE L,127
50 FOR I = 1 TO 50: NEXT I
60 NEXT I
```

The program is short and simple, and it moves a line across the screen quickly, even with the delay loop. But it has one drawback fatal to any game: the animation is jerky instead of being nice and smooth. Poking the value 127 turns on all seven dots of a byte, and if you increase the delay, you'll see that the line moves in one-byte increments, which explains the uneven movement. The answer is elegant, though not without problems: move the figure along one dot at a time.

Imagine that you are looking out a window that is seven dots wide, and that the line crawls across your field of vision. At first you see only the leading dot, then the first two dots, then three, four, and so on until all seven dots are visible through the window. Then, as the line continues, the leading dot moves out of range, then the second dot follows; that continues until the window is empty.

In computer terms, the window is one byte of memory, and when just the lead dot of the line is showing in a byte, the other six dots are showing in the previous byte. In that case, you need one byte with just the left-most dot on, and another with the six right-hand dots on; the values 1 (0000 0001) and 126 (0111 1110) will do the trick. Again, remember that the bit pattern is the reverse of the desired dot pattern. (Curses!) From Basic type:

```
HGR
POKE 8192,126: POKE 8193,1
POKE 8192,124: POKE 8193,3
POKE 8192,120: POKE 8193,7
POKE 8192,112: POKE 8193,15
POKE 8192,96: POKE 8193,31
POKE 8192,64: POKE 8193,63
POKE 8192,0: POKE 8193,127
```

It is another characteristic of byte-move graphics that each figure requires seven shifted copies, or separations. That means that a
figure one byte wide actually requires two bytes, a two byte figure requires three bytes, and so on.

Entering each pair of pokes shifts the line one dot to the right, so the cumulative effect is to move the line slowly across the screen. You could continue the process by poking the same sequence of values into locations 8193 and 8194, but at that rate it would take you several hours to go all the way across. The following program does essentially that, but faster.

```
10 DIM A%(280) : REM 280 X COORDINATES
20 REM
30 REM READ THE VALUES FOR
40 REM THE 7 PAIRS OF FRAMES
50 REM
60 FOR I = 0 TO 6
70 READ 11/4(1),F/0(1)
80 NEXT I
90 REM
100 HGR
110 REM
120 REM INITIALIZE THE TABLE
130 REM OF ADDRESSES
140 REM
150 J = 0
160 FOR I = 8192 TO 8231
170 A%(J) = I: J = J + 1
180 NEXT I
190 REM
200 REM PLOT THE LINE AT
210 REM EACH X COORDINATE
220 REM
230 FOR X = 1 TO 280
240 Q% = INT (X / 7)
250 R% = X — (7 * Q%)
260 C% = Q% + 1
270 POKE A%(Q%),T%(R%): POKE A%(C%),H%(R%)
280 NEXT X
290 END
```

Again, we use variables to speed up the program and confuse the reader. The values for the line are read into arrays T% (for tail) and H% (head) in lines 30 to 80, and array A% contains the addresses for each of the forty bytes across the top of the screen and is initialized in lines 120 through 180. The loop from 230 to 280 plots the line at every X coordinate across the screen, but lines 240, 250, and 260 merit more study.

The purpose of these lines is to determine which pair of bytes is being used and which of the seven pairs of values need to be poked. They do that by dividing the X coordinate by 7, and calculating the quotient (Q%) and the remainder (R%). For example, when the X coordinate is 73, seven g'zinta 73 (do you remember your g'zintas?) ten times, with three left over. So you need to poke the tenth and eleventh bytes with the third pair of values.

As you can see, even as simple a figure as the line requires seven different versions and significant preparation to animate horizontally, but the animation that results is as smooth as you could wish for, even if it is a bit slow. Most games are written in machine language to take advantage of the better speed of execution, but these examples in Basic serve to give you the idea.

Next time, we'll talk about ways you can streamline animation by doing partial modifications, preshifting, and precomputing. After that, we'll talk about some of the methods used to detect collisions between objects on the screen.

But for now, you have enough stuff to make your head hurt until the next issue arrives.

---

**Rana Systems**

Your Apple computer can grow from ordinary to awesome in a matter of minutes.

The Elite disk drive Series by Rana Systems gives you that kind of magic. Quickly, easily and cost-effectively.

This superb family of sophisticated floppy drives was designed and perfected by a team of industry-respected engineers expressly for Apple® computer owners who have long been demanding more disk drive for their money.

Now they have a drive that makes an Apple perform the way it should. For example:

- **Elite One**, the most economical Rana drive, delivers 15% more storage capacity than Apple's drive. The top-of-the-line **Elite Three** will provide an astonishing four-times more storage, approaching hard disk performance. It's done through exclusive high-density single and double-sided disks and heads.

**Smartmodem**

The PKASO Interface

The master PKASO printer interface at a very low cost

- **Gray scale printing**
- **Snapshot screen dump**
- **Support for the Apple Z-80 CP/M Apple III compatibility**

**Gold Disk**

- **Applewriter**
- **Interface Cards**
- **BASIC disk with every $90 worth of purchase**

**Attach System Saver**

- **Free Data Disk With Apple II**
- **EPROM chip makes high-resolution graphics easy.**
- **Version to accommodate the Amiga, Epson Series™, iBm Paper Tigers, Gencerics 7300, NEC 8023, C. Joth Prowriter, Okidata Series™ and more.**

**The Grappler**

Apple Graphics Interface

- **Interface Cards**
- **Free Data Disk With Every $90 Worth of Purchase**
- **EPROM chip makes high-resolution graphics easy.**
- **Version to accommodate the Amiga, Epson Series™, iBm Paper Tigers, Gencerics 7300, NEC 8023, C. Joth Prowriter, Okidata Series™ and more.**

**Apple II**

- **Applewriter**
- **Interface Cards**
- **BASIC disk with every $90 worth of purchase**

**TG Products**

- **JOYSTICK**
- **COMPUTER**
- **EXECUTIVE SECRETARY**
- **FINESSE"**
- **No Credit Card Special Discount**
- **END OF SUMMER SPECIAL DISCOUNT FOR ALL STUDENTS AND PARENTS OF STUDENTS!**

**GOLD DISK™**

- **Software**
- **Box 102**
- **Glen Arm, Md. 21057**
- **TOLL FREE 1-800-368-2260**

**APPLE**

- **A Trademark of Apple Computer Inc.**

**SEPTEMBER 1982**
PINBALL HAS GONE INTO ORBIT!
Now that you’ve conquered pool hall pinball, it’s time to take your Apple* into outer space

ZERO GRAVITY PINBALL

A machine language arcade game with full-color hi-res graphics, and sounds like you’ve never heard before.

ZERO GRAVITY PINBALL
$29.95 plus postage
Requires game paddle 0, Apple II*, with Applesoft* ROM, 48K, DOS 3.2 or 3.3

AVANT-GARDE CREATIONS
P.O. Box 30160
Eugene, OR 97403
(503) 345-3043

*Apple II, Applesoft are registered trademarks of Apple Computer, Inc.
Crossfire (Apple), Sierra On-Line
Guardian, Continental
Goblins, Highlands
Galactic Chase, Prism
Epoch, Sirius
Dodge Racer, Synapse
County Fair, DataMost
Chicken (Atari), Synapse
David's Midnight Magic, Epyx
Cyclod (Atari), Sirius
Avalanche, Atari
Lemmings, Lazer Silk, Geppi
Lazer Ray, George Spivak
Missions, Sirius
Mar Tedoro, Syncro

**Score**

ABM, Muse
Action Quest, JY Software
Alien Ambush, Peter Fokos
Alien Descent (TRS-80)
Alien Rain, Broderbund
Alien Swarm, In-Home
Alien Typhoon, Broderbund
Andromeda, Geppi
Apple Panic (Apple), Broderbund
Apple Panic (Atari), Broderbund
Asteroid Fields
Astrolords, Atari
Autobahn, Sirius
Avalanche, Atari
Bandini, Sirius
Beer Run, Sirius
Beezian, Bez
Beez-Off, Bez
Blitz Blaster, Creative Computing
Blitz Blaster, Creative Computing
Borg, Sirius
Bug Attack (Apple), Cavalier
Bug Attack (Atari), Cavalier
Cannonball Blitz, Sierra On-Line
Canyon Climber (Atari), Datasoft
Caverns of Mars, Atari
Ceiling Zero, Turkeny
Chicken (Atari), Synapse
Chipton, California Pacific
Clowns and Balloons (Atari), Datasoft
Congo, Sentinel
Cosmic Mission, Atari Int'l
County Fair, DataMost
Cricctek,
Crossfire (Apple), Sierra On-Line
Crossfire (Atari), Sierra On-Line
Crush, Crumble, and Chomp, Epyx
Cybog (Apple), Sirius
Cybog (Atari), Sirius
Dark Forest, Synapse
Datestones of Rym, Epyx
David's Midnight Magic, Broderbund
Dodger Racer, Synapse
Dogfight, Micro Lab
Dragon's Eye, Epyx
Eliminator, Adventure Int'l
Epoch, Sirius
Escape, Muse
Escape from Arcturus, Synergistic
Falcons, Piccadilly
Firebird, Geppi
Fireball, Geppi
Fishing, Muse
Galactic Chase, Prism
Galactic Empire, Broderbund
Galactic Quest, Crystalware
Galaxy Travelers, Atari
Galaxy Wars, Broderbund
Gamma Goblins, Sirius
Genetic Drift, Broderbund
Ghouls and Ghosts Plus
Goblians, Highlinds
Golden Mountain, Broderbund
Gold Rush, Sentinel
Gorgon, Sirius
Guardian, Continental
Hadrion, Sirius
Head-On, California Pacific
High Orbit, Geppi
Horizon Y, Geppi
Human Fly, CPU
Hungry Boy, California Pacific
Int'l Gran Prix, Riverbank
Intruder Alert, Dynacom
Invasion Force, Computhings
Jawbreaker (Apple), Sierra On-Line
Jawbreaker (Atari), Sierra On-Line
Jellyfish, Sirius
Juggler, IDS
Kamikaze, Hayden
Kickbox, Shoot-Out, K-Byte
Kayos, Computer Magic
Labyrinth, Broderbund
Laf Paq, Apple Zap, Sierra On-Line
Laf Paq, Creepy Corners, Sierra On-Line
Laf Paq, Mine Sweep, Sierra On-Line
Laser Maze, Avant-Garde
Lazer Silk, Geppi
Leemnings, Sirius
Maureaen, Sierra On-Line
Mar Tesoro, Syncro

**Player**

Peter Sivo, Saratoga, CA
Steve Halberstadt, Newark, DE
Derin Basden, Fresno, CA
Chris Athane, Topsham, MA
Ron Bunch, Collegedale, TN
Shane Rolin, Monroeville, PA
Denise Achram, Canton, MI
Joey Grisafi, Houston, TX
James Baker, Alexandria, VA
Shane Rolin, Monroeville, PA
Jeff Feldman, Miami Beach, FL
Ken Williams, Niles, OH
Mark Hall, Lake Oswego, OR
Brian Hall, Millard, MI
Norman Fong, San Francisco, CA
Norman Fong, San Francisco, CA
Harry Ilg, Chesterton, MO
Norman Humbert, Fountain Valley, CA
Kerry Shetline, Merrick, Monroisnt, NJ
Klaus Liebold, Poway, CA
Matt Newell, Lincoln, NE
David Porter, Millindale, IL
Chris Conway, Hattiesburg, MS
Ron Bunch, Collegedale, TN
Shane Rolin, Monroeville, PA
Tak Sesho, Boston, MA
Denise Achram, Canton, MI
Shane Rolin, Monroeville, PA
Jason Meggs, Bethesda, MD
Joey Grisafi, Houston, TX
Mark Mckenna, San Jose, CA
Matt Skinner, Springfield, VA
Foster City, CA
Shane Rolin, Monroeville, PA
Tak Sesho, Boston, MA

**Verified score**

*Verified*
John Hickey would like to know how long the present and previous title holders for Falcons had the game before posting their highs. He can't seem to make any progress.

Our sympathies to Peter Gordon, who played Snake Byte all night before going to bed (leaving the computer on, at an hour past one, from his father), and was about to wake up his sisters the next morning to show them his high score when a power failure hit Teaneck, New Jersey, and all slept.

Fred Nieswanger, thirteen, has not been able to figure out how to get past the first chasm in the Princess game, at risk of great peril from his father, and was about to wake up his sisters the next morning to show them his high score when a power failure hit Teaneck, New Jersey, and all slept...

Eric Snider emerged the victor from an eleven-hour intercontinental struggle with Eric Papel for supremacy in Serpentine. (There is certainly no significance in the similarities between Mr. Snider's name and hometown and that of author David Snider--he didn't get his copy before anyone else or anything like that, heck no.) Snider wants to know if we will return disks used as verification of high scores. Yes, if a disk mailer with return address and adequate postage is included.

A.J. Benway's Caverns of Mars challenge to Shane Rolin is by the way, do you live in Pittsburgh or Monroeville? Do you commute?) must now include Jeff Parish a liar; maybe he has a different version than whose verification photo, we admit, did not include Eric Vesper, Saint Louis, MO.

A.J. Benway's Caverns of Mars challenge to Shane Rolin is by the way, do you live in Pittsburgh or Monroeville? Do you commute?) must now include Jeff Parish a liar; maybe he has a different version than whose verification photo, we admit, did not include Eric Vesper, Saint Louis, MO.

Highlines

John Hickey would like to know how long the present and previous title holders for Falcons had the game before posting their highs. He can't seem to make any progress.

Our sympathies to Peter Gordon, who played Snake Byte all night before going to bed (leaving the computer on, at an hour past one, from his father), and was about to wake up his sisters the next morning to show them his high score when a power failure hit Teaneck, New Jersey, and all slept...

Eric Snider emerged the victor from an eleven-hour intercontinental struggle with Eric Papel for supremacy in Serpentine. (There is certainly no significance in the similarities between Mr. Snider's name and hometown and that of author David Snider--he didn't get his copy before anyone else or anything like that, heck no.) Snider wants to know if we will return disks used as verification of high scores. Yes, if a disk mailer with return address and adequate postage is included.

A.J. Benway's Caverns of Mars challenge to Shane Rolin is by the way, do you live in Pittsburgh or Monroeville? Do you commute?) must now include Jeff Parish a liar; maybe he has a different version than whose verification photo, we admit, did not include Eric Vesper, Saint Louis, MO.

A.J. Benway's Caverns of Mars challenge to Shane Rolin is by the way, do you live in Pittsburgh or Monroeville? Do you commute?) must now include Jeff Parish a liar; maybe he has a different version than whose verification photo, we admit, did not include Eric Vesper, Saint Louis, MO.
The Official FROGGER™

by SEGA®

The Popular Arcade Game
Now Available For Home Computers

FROGGER runs on any 48K APPLE II with
DOS 3.3 or ATARI 400/800 with 32K
and a Diskdrive, 16K and cassette.
Frogger is available for $34.95 at your
local computer store or order
directly from...

APPLE VERSION BY
OLAF LUBECK

ATAIR VERSION BY
JOHN HARRIS

All orders by VISA, Mastercard, check, or COD. Add one dollar for shipping. California residents please include 6% sales tax.

ON-LINE systems
36375 Mudge Ranch Rd., Coarsegold, CA 93614
(209) 683-6858

® and ™ designate trademarks of Sega Enterprises, Inc.
© 1981 Sega Enterprises, Inc.
Welcome to your breaking point.

Try to remain calm. We understand. You are worried that your computer might be attempting to coerce you? Well, what's a little coercion among friends?

As a new arrival on The Island, you will be expected to divulge certain information. You will not leave until we obtain it... and we will. Of course, when we gain, you lose.

Please endeavor to maintain the proper perspective. A pleasant adventure can so easily lead to obsession... Applesoft, 48K, DOS 3.3... $32.95 Available at computer stores on finer islands everywhere. Exclusively from It™ Interactive fantasies, a division of Edu-Ware Services.

P.O. Box 22222 • Agoura, CA 91301

Prisoner 2

A science fiction nightmare by David Mullich.